Learning the Latent Structure of Translation

Markos Mylonakis

Learning the Latent Structure of Translation

Markos Mylonakis

Learning the Latent Structure of Translation

ILLC Dissertation Series DS-2012-02

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Learning the Latent Structure of Translation

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op donderdag 19 januari 2012, te 12.00 uur

door

Markos Mylonakis

geboren te Cholargos, Griekenland.

Promotiecommissie:

Promotor: Prof. dr. R. Scha
Co-promotor: Dr. K. Sima'an

Overige leden:
Prof. dr. P. Adriaans
Prof. dr. R. Bod
Prof. dr. K. Knight
Prof. dr.-ing. H. Ney
Dr. A. Way

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The investigations were supported by a VIDI grant (nr. 639.022.604) from the Netherlands Organization for Scientific Research (NWO).

Copyright © 2012 by Markos Mylonakis

Cover design by Georgios Grigoriadis and Maria Mylona.
Printed and bound by Wöhrmann Print Service, Zutphen.

ISBN: 978-90-5776-235-2

to my beloved mother

and to Maria, my only rose

the pen that was writing this thesis

we were always holding it together

Contents

Acknowledgments

A thesis is perhaps the most personal and solitary piece of work that a scientist will ever produce. We young researchers must find the courage to venture for the first time alone to the dark corners of knowledge and make a bold step forward where nobody has ever been before, which is outright scary! Luckily, even though in this age-old intellectual rite-of-passage we are the ones who ultimately have to call the shots, we are never truly alone. There is a university, a promotor and a supervisor, there are also committee members and colleagues, friends, relatives and loved ones. There are the people who helped us make it to the starting line and those that are waiting for us after the end of the journey. I feel deeply indebted to all those who, in every possible way, accompanied me in these scientific and personal explorations. In these few lines, I would like to express my utmost gratitude for their help and encouragement.

In the beginning there was hope, aspirations and the great unknown. And God said: 'Let there be a supervisor', and there was Khalil Sima'an. I would like to thank Khalil for his continuous support for more than six years now, all the way from supervising my Master thesis to my PhD defence. He would pose the great questions and encourage me to go out and find my own answers. He taught me how to move past the surface and dare to face the deeper facets of scientific problems, however frightening this may be. He made this project possible with his constant encouragement and sincere belief that I could do it, even in my moments of doubt. For all this I am truly indebted to him, but most importantly I thank him for providing the living example of a true scientist.

I was also privileged to have Remko Scha as my promotor. Remko provided exceptional feedback through all the process of writing the thesis. His insightful comments decisively aided in increasing the quality of this work, as well as my own intuitive understanding of the material, and helped to bring everything into context. Throughout my PhD, I enjoyed the purest moments of scientific bliss during the discussions that I held with Remko and Khalil on the chapters of this thesis.

Apart from my promotor and co-promotor, I would like to wholeheartedly thank the other members of my doctoral committee: Pieter Adriaans, Rens Bod, Kevin Knight, Hermann Ney and Andy Way. I thank them for their kind interest in this work and for the valuable time that they spent reading this thesis. I am particularly indebted to Kevin Knight for the enthusiasm he showed for this work and for providing me with his detailed comments. Hermann Ney, unknowingly, played an instrumental role in my choice to join the ranks of Statistical Machine Translation researchers, by delivering a most interesting speech on SMT at the CLIN conference in Amsterdam, back in 2005. His work together with the work of Kevin Knight and his colleagues at the Information Sciences Institute has provided a guiding light for my efforts all this time. The work of Rens Bod and Andy Way on modelling natural language data with fragments has been highly influential for this thesis, while Pieter Adriaans serves as a shining example of the broad-minded scientist that I aspire one day to become.

This thesis and the work behind it decisively benefited by the intellectual air of the Institute of Logic, Language and Computation, which I breathed throughout my PhD studies. Apart from Khalil, Remko and Rens, I was lucky enough to interact with Jelle Zuidema, Stefan Frank, Tejaswini Deoskar, Maxim Khalilov, Gideon Borensztajn, Federico Sangati, Reut Tsarfaty, Gideon Maillette de Buy Wenniger, Sanne Korzec, Paul van Ulsen, Anne Maren Krüse and Lena Kurzen. I would like to particularly thank Tejaswini for our highly interesting collaboration and for giving me the chance to have some hands-on experience with parsing, as well as fellow SMT researchers Maxim and Gideon Maillette de Buy Wenniger for being extremely helpful whenever needed and for our interesting discussions. I would like to further thank Andreas Zollmann, ILLC alumnus and excellent fellow SMT researcher, for his intuitive feedback on work included in this thesis and his kind interest and encouragement. I thank Federico and Gideon Borensztajn for being great friends and for being patient and encouraging during our ping-pong games. I am deeply grateful to Paul van Ulsen for helping me learn how to speak decent Dutch by being patient with me during our long discussions in the language. Our walks in the Dutch countryside uncovered for me a whole new (flat) world of beauty and allowed me to learn so much about the country and its history.

Our ex-director, Leen Torenvliet was kind enough to bare with my continuous complaints due to my insatiable demand for computing power and was instrumental in making sure that our group was provided with the infrastructure we needed; the results in this thesis would have never been reached without it. Our manager during most of my time at the ILLC, Ingrid van Loon, was always there when needed and made sure the PhD project stayed on track during the less easy times. I would like to also thank our administrative staff, Tanja Kassenaar, Karin Gigengack, Marjan Veldhuisen and Peter van Ormondt, who always generously provided their support and advice.

Before the ILLC, there were the Netherlands and the University of Amsterdam and the people who helped me to arrive in this amazing country. First of all, I would like to thank the anonymous employee of the European Space Agency (or perhaps the computer program he was using) for unknowingly initiating the process that brought me in Amsterdam. How this happened exactly will always remain a story worth telling. Then, it was Theo Gevers who gave me a place in the Master of Artificial Intelligence of the University of Amsterdam and Niels Molenaar, Hideko Gieske and the other people at the International Office who made sure, in a most surrealistic way, that I would start a great new life here.

Back then, I took the decision to get on a one-way flight and arrive in Amsterdam, without any reassurances that there was even a place for me at the MSc programme, as it was weeks before I finally got formally accepted and courses had already began. I was living out of a suitcase and sleeping on a hostel bed. After checking in my hostel, I took the tram and went, without an appointment, to see Niels. Right there in the middle of my daring leap of faith, I was not sure if I would land on a new adventure or if my trip would end as a big disaster. Niels sat to talk to me, smiled and said 'well, now that you're here, let's just fix everything'. After a couple of days, I had a new apartment and I was following courses. I will always remember these days as a time when I was taught how pure kindness and human interest can transform the most perilous moments into bright new beginnings.

Apart from all these wonderful people that I was lucky enough to encounter during my time at the UvA, I would also like to acknowledge the enduring support and encouragement that I received during this period from my friends and beloved family members. Angelos and Valentini Nastoulis have embraced me with their true friendship for almost two decades now. My discussions with Angelos allow me to feel for a few moments like a philosopher of the old days, when we take time to pause and question everything anew. Valentini's warmth of heart and encouragement has provided the fuel to move further many a time. Manolis Foundoulakis has been a close friend for a whole decade and he was decisive in convincing me to move toward a researcher's education and career. His belief in me all this time and our precious time together whenever possible helped to make the life project he inspired a reality. Persa Karanika and Iphigeneia Vrettou remained good friends despite the distance and their kind words and the inspiration they provided whenever we met was really important to me. Georgios Grigoriadis has laid his mark firmly on this thesis by kindly providing his expert skills for the cover design.

One thing that kept me going through all this time was my love of the mountains and nature, which always provided solace when I needed it most. Although my decision to come to the Netherlands meant that I would have to travel a bit further away to come closer to the windy peaks, somehow magically there was always a chance to do so during these years. I would like to thank all the people that I met in these trips, particularly those that accompanied me while walking the Santiago de Compostela path in Spain: Asuncion Revert Garcia, Sara Shellenbarger, Magnus Casara and Barbara Blaukopf are only some of them. Their words of wisdom have made myself as a person, and this thesis as a direct result, so much better. Still, I would not have met all these beautiful people if it had not been for Vangelis Vroutsis, the mountaineering instructor who inspired the love of the mountains in me. The news that he passed away during my PhD studies, while venturing out in the mountains with his beloved students, deeply saddened me. I will always continue to admire the prime example of a fine mountaineer that he was and, in his memory, I shall keep the mountaineering flame alive and strong within me.

Apart from my friends in Greece, I was also lucky to meet highly interesting people in the Netherlands as well and they have all supported me while performing this work in many different ways. Isaac Esteban was the best study mate during my MSc years and a true inspiration for the time afterwards. Gideon Borensztajn, apart from a colleague and a long-time room-mate, was also a good friend who provided enormous help during the last three years. Tejaswini Deoskar was a further colleague I was fortunate to get to know more and was extremely encouraging when inevitably things got tough, apart from being a great travel mate. As they were located in Groningen which is 200 km away from Amsterdam, I did not manage to see Barbara Plank and Martijn Wieling very often. Still I was always happy to meet them at the most distant places during conferences. Aspasia Beneti, Simon Butterworth, Antoinette Christou, Vanessa Diehr, Nina Godeke, Dimitra Kassari, Sybren van der Kolk, Elisavet and Iliana Kyritsi, Maria Petrovas, Fotis Stringos and Na Yang are just some of the people I was fortunate to have around me during the time I was compiling this work.

Then there is family, and what an amazing one I was blessed to have in this life! Words cannot describe the extent of the support and encouragement I received from my family all this time. First and foremost, I would like to deeply thank my mother, Maria Xydianoy, for her enduring love and her persevering belief in me. Day and night, through the brightest successes and the darkest moments, we were always walking this path together. Everything I am and will be, I owe it to her. My father, Nikos Mylonakis, was instrumental in getting me obsessed with computers and artificial intelligence. Thanks to him, I was lucky to have grown up with a computer at home since the day I was born, a rare coincidence for my generation. Together, we explored and dreamt what these machines could do for us humans and the influence of these days will always endure within me. My sister, Zoi Mylonaki, steadfastly stood beside me throughout all these years and her unceasing encouragement meant the world to me. She is someone I deeply admire and her existence makes this world so much better in many different ways. My brother, Damianos Mylonakis, has provided his love and advice without a break. The person he will be in 12 years is the man I aspire to be today, even though I know I could never reach that far. Antonis Katelouzos and Stamatis Salamouras were also extremely supportive and our interesting discussions supplied ample food for thought. Alexandros, Martha, Anastasia and Kostas Mylonas embraced me in their family like one of their own and their warm words of encouragement were precious for me. I am also indebted to all my relatives in Crete, Athens and Larisa (and those venturing all over the world) whose love and positive energy also drove this work forward. Above all, I would like to express my gratitude to my grandparents Zoi and Damianos Mylonakis, who passed away recently, for their love and support. Grandma Zoi, the thread of love and appreciation that connects us is stronger than death itself.

I would like to extend my most sincere words of gratitude to my beloved partner in life, Maria Mylona. Every step of this journey, we travelled together. All this time, there were moments I felt disappointed, weary or scared. Still, there was not a single second I felt alone and this I owe it to her. Her constant support, her unconstrained belief in me, her brave heart often extended as a covering shield, but above all her enduring love, was what made all this not only possible, but also worthwhile living. From deep within, I thank her for everything she is for me.

The final word of appreciation goes to all those who I did not explicitly mention and for which I ask their forgiveness. It goes to all those who chose to extend a kind word when it was most needed. It goes to the teachers and professors of the past who shaped my intellect and to the friends and acquaintances whose advice brought me where I stand now. It goes to Nikos Kazantzakis, Odysseas Elytis and the other writers and poets, who tried to teach me how to live life, although I am afraid I will most probably always remain a bad student despite honestly doing my best. It goes to all the unsung heroes who inspire me by doing their best to keep their spirit lucid and free in the darkest of times. Last but not least, I express my gratitude to this magnificent country and its people, the place that wholeheartedly embraced myself and my dreams for such a long time: the Netherlands.

Markos Mylonakis
Amsterdam, December 2011

Chapter 1

Introduction

Human languages are highly structured both from a syntactic and from a semantic point of view. This fundamental property makes it possible to efficiently convey an unlimited number of semantic concepts through natural language sentences. Crucially, multilingual data created by translation involve an additional layer of structure, which pivots between the syntactic and semantic patterns that appear in the different manifestations of human language.

In this thesis, we present methods to automatically learn phrase-based Statistical Machine Translation (SMT) models that assume a latent bilingual structure as their central modelling variable. Acknowledging that each language is strongly characterised by its individual structural properties, we aim to learn a bilingual structure that augments and supersedes its monolingual counterparts to bridge the gap between them. This structure is a latent one, because the translation data that we use to discover it do not explicitly identify it. The parallel corpora we use, consist of *source* sentences in the language we wish to translate from, paired with an existing human translation in the *target* language, but without any information on why the particular translation was chosen.

The goal of uncovering the hidden structure of translation is not new. On the contrary, it has formed the spearhead of Machine Translation (MT) research, right from the first steps of this field and up to this day. Already in the early days of MT, researchers strove to manually identify the latent patterns of translation, and encode them as a set of rules that governs the translation process. However, it was gradually recognised that the level of complexity of cross-language communication rendered this effort extremely difficult. Statistical Machine Translation aims to overcome the limitations of rule-based systems, through automatically learning bilingual correspondences between the source and target sentences from parallel corpora. All SMT models also assume an explicit or implicit latent structure in translation data, and one of the central problems in SMT is learning this bilingual structure using a parallel corpus.

Crucially, SMT research, in its large majority, has been fairly modest in the kinds of structure assumed in its models, not daring to explore the complexity and richness of bilingual data. SMT models mostly stay close to the lexical surface to model translation from strings to strings, greatly trivialising the syntactic aspects of language. These oversimplifying assumptions allowed to avoid the learning challenges posed by more complex models of translation.

In this work, we move further than this and contribute methods to model and learn the latent structure of translation. We choose to face the problems that plagued previous efforts in this direction and propose solutions. We find that, to a large extent, these problems can be attributed to the sparse nature of translation data. As the models become more complex, naïve learning algorithms are increasingly exposed to the danger of fixating on the particularities and the inherent noise of training data, crucially missing the opportunity to identify the underlying patterns.

We contribute a learning framework that addresses these issues, based on a long-established Machine Learning method: Cross-Validation. We show how this can be fused with the well-understood Maximum Likelihood Estimation (MLE) approach, to formulate a Cross-Validated MLE (CV-MLE) learning objective that directly aims to discover latent patterns that generalise well. We further provide the Cross-Validated EM algorithm, an instance of the equally well-understood Expectation-Maximization algorithm, to optimise parameters of models employing latent variables according to the CV-MLE criterion.

We subsequently apply our learning framework to induce, for the first time using a clear learning objective, translation models which capture the hierarchical, recursive structure of translation. Our method learns how to exploit monolingual syntactic structure to discover linguistically motivated translation patterns. We empirically show that our learnt models compare favourably to the state-of-the-art across multiple language pairs. In this way, we showcase how learning the structural aspects of translation can aid in delivering tangible improvement in translation performance.

In the rest of this chapter, we briefly introduce the three concepts which underlie this thesis.

1. We highlight the *latent* character of bilingual correspondence and consider what this implies for methods aspiring to automatically learn it.

2. We discuss different approaches to modelling *translation structure* and introduce the translation paradigms that we will employ in later chapters.

3. We consider the task of *learning* models assuming latent translation structure variables and discuss some of the challenges that we address in the rest of the thesis.

We close the introduction to this work with an overview of each of the chapters that follow.

1.1 The Latent Nature of Translation

We regard as translation structure the bilingual patterns that describe the correspondences between pairs of sentences in two languages, with each considered as the translation of the other. This structure identifies how the components of each sentence map to those of its translation counterpart. As such sentence components we might for example consider words, contiguous or discontiguous phrases, linguistic constituents or semantic units. Translation structures describe how these components correspond to each other, explaining the transformations taking place during the translation process.

In an SMT model trained from parallel corpora, the model variables corresponding to the translation structure are *latent*. The training corpora consist of whole source sentences each paired with their target language translations, without further annotation regarding how their sub-strings relate to each other. Even though sometimes, as is the case in this thesis, these training sentence-pairs might also be word-aligned[1], we still cannot directly identify in the data other bilingual patterns, such as the clustering of words into phrase-pairs or the hierarchical correspondences between bilingual spans. A model assuming latent translation structure considers the values of the latent structure variables as missing from the training data; the problem of modelling them involves learning from incomplete data.

The training data provide no explicit clues on the form or properties of the hidden translation structure. It is the task of the modeller to define these by setting up the parts of the translation model space relating to the latent structural variables. Modelling options include choosing a word or phrase-based approach, assuming a flat structure directly over the lexical surface or a multilevel hierarchical structure, establishing a link between syntactic and translation analyses etc.

We believe that good choices related to the assumptions on the form of the latent translation structure, are those that lead to learning translation models which generalise well and translate adequately. This entails that the appropriateness of a model assuming a certain flavour of hidden translation structure must be evaluated in relation to the data that it will first train upon and those that it will later process, as well as the algorithmic context within which it will be employed. The learning algorithms which are used to train it, as well as the translation (decoding) apparatus that will be used to select translations for source sentences given the trained model, can also have a significant impact on the actual translation performance of the model. Furthermore, some translation models perform better in practice for certain language pairs, even for certain translation directions between them.

[1] Word-aligned sentence-pairs include the word to word correspondences between the source and target sentences. These correspondences can be automatically identified by trained word-alignment models.

Language Sparsity Irrespective of the particular choices involved, translation structure modelling seeks to take advantage of the inherent structural properties of monolingual source and target data to better model the correspondences between them. One might argue that in the face of the increasing availability of parallel training data, aiming to understand these correspondences is not necessary. As the size of the training data grows, there is a higher chance of retrieving from them the translations for large segments of test source sentences, eliminating the need to analyse how smaller fragments combine. However, such a view disregards the sparse nature of language. While extracting the translations of multi-word fragments from the training data has been shown to significantly raise translation quality (Och and Ney, 2004; Koehn et al., 2003; Chiang, 2005a) and the empirical part of this thesis uses solely such models, there is a limit on the extent that this can be applied to avoid modelling how these fragments combine. Even if we had access to a parallel corpus consisting of all the sentences on the world wide web and their translations, we would find it hard to match longer segments of yet unseen source sentences. This hardly relates solely to rare uses of language, but also for seemingly 'normal' segments of sentences such as the first four words of this sentence[2]. Irrespective of the size of our training data, the sparsity of natural language makes modelling the latent structural aspects of translation necessary, in order to produce fluent translations that convey meaning accurately.

1.2 Modelling Translation Structure

The development of translation models in the literature has proceeded in a stepwise fashion. Right from the beginning of SMT, the seminal work on the IBM Statistical Machine Translation models (Brown et al., 1993) was presented as a succession of translation models of increasing complexity. Formulating translation models is challenging and involves weighing together the perceived expressiveness of the models on the one hand, with the complexity of the computations involved and the machine learning challenges on the other.

From a probabilistic point of view, any translation model assumes a certain amount of structure between sentence-pairs, by preferring translations with certain properties (e.g. monotone translations that largely keep the word-order intact) over alternative ones. In this thesis, we focus on models assuming transla-

[2] Searching the web for the phrase 'this hardly relates solely' returns zero matches on Google, while 'this hardly' and 'relates solely' returns hundreds of thousands of matches. To be able to translate the original four-word phrase adequately, a system having access to a hypothetical parallel corpus with the size of the web must still know how to combine together the translations of its two-word sub-phrases. The same applies for other phrases from this paragraph such as 'web and their translations', 'translations of multi-word fragments', 'while extracting the translations', 'view disregards the sparse' and many others. A lot of relatively short phrases from this thesis, like in any other natural language text, have never been formulated before.

tion structures based on contiguous and discontiguous multi-word units (phrases). The value of such multi-word units had been already recognised in the original IBM SMT models (Brown et al., 1993). While these early models are widely referred to as 'word-based', a subset of them[3] considers how single words produce multiple words as their translation, and models their tendency to cluster together as a phrase in the translated sentence. However, it was Phrase-Based SMT (Och et al., 1999; Koehn et al., 2003) that introduced modelling the phrase to phrase (i.e. many-to-many words), mapping between the source and target sentences.

A Phrase-Based Approach In this thesis we also follow a phrase-based approach to translation, considering the correspondences between both contiguous and discontiguous multi-word segments of sentences. Under this view, the translation structure for a sentence-pair, consisting of a source sentence and its target language translation, involves the following aspects:

Phrase Segmentation The structure must describe how the sentence-pair is segmented in phrase-pairs, where each target phrase in a phrase-pair is the translation of its source counterpart. These phrase-pairs can be contiguous (e.g. in lowercased English-French ⟨i am / je suis⟩), or discontiguous (e.g. ⟨not / ne ... pas⟩). Each phrase-pair is considered atomic, i.e. it cannot be further analysed in terms of combining together smaller phrase or word-pairs.

Reordering The target parts of phrase-pairs are frequently reordered in relation to the order of the source phrases they are paired with. The translation structure must specify how the source and target parts of phrase-pairs are positioned in the source and target sentences respectively.

Abstract Hierarchical Structure Some of the models explored in this thesis explain the correspondence between the phrase-pairs of a sentence-pair in terms of an abstract hierarchical structure. This makes use of abstract (unlexicalised) categories, possibly linguistically motivated, which are combined together to form a hierarchical, recursive structure spanning across the sentence-pair. This structure might also describe the reordering patterns between the phrase-pairs.

Hierarchical Modelling As the thesis progresses, we focus on models assuming an abstract hierarchical translation structure that progressively gets more involved. These models will be based on the probabilistic Synchronous Context-Free Grammar formalism and its Inversion-Transduction Grammar subset (Wu,

[3] IBM Model 3 introduces a 'word fertility' variable tracking the number of words produced as the translation of a single word. Models 4 and 5 further model the tendency of these multiple words originating as translations of the same word to cluster together.

1997). This formalism extends the familiar concept of probabilistic Context-Free Grammars from the monolingual to the bilingual domain, to model pairs of strings instead of single sentences. It allows to both model discontiguous phrase-pairs as well as a hierarchical bilingual structure making use of abstract categories that are recursively expanded to derive a sentence-pair.

The introduction of synchronous grammars for SMT (Wu, 1997; Chiang, 2005a) cleared the way to take advantage of the inherent hierarchical structure of language in Machine Translation. This created the conditions to bring together the hierarchical, phrase-based modelling of MT with the existing thread of research exploring linguistic syntax-based SMT (Yamada and Knight, 2001; Galley et al., 2004). The result is work which explores hierarchical translation models driven by linguistic syntax for monolingual data.

This thesis concerns itself with all of the above models of translation structure. A comprehensive presentation of these models can be found in Chapter 2, while Chapters 4 to 6 examine our empirical work on learning progressively more complex models assuming a latent translation structure.

1.3 Learning Phrase-Based Translation Structure

Expectation-Maximization After establishing a certain translation model space, the next step involves estimating its parameters from the parallel training data. The introduction of SMT methods in terms of the IBM SMT models was based on employing the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to estimate translation model parameters according to a Maximum Likelihood Estimation (MLE) learning objective. Translation models are typically too complex to compute an MLE estimate analytically. Instead, the EM algorithm iteratively climbs the training data likelihood function producing a series of estimates, each further raising the data likelihood until convergence to a local optimum. The same methodology has been applied to other word-based SMT models such as the HMM alignment model (Vogel et al., 1996). However, as the translation models became more complex and especially after the introduction of Phrase-Based SMT, the weaknesses of naïve applications of EM became evident, and researchers turned to heuristic, ad hoc estimators.

Learning Fragment Models The transition to phrase-based models brought with it new learning challenges. As we discuss in detail in Chapter 3, phrase-based SMT models belong to the wider family of Fragment Models (FMs), which was first introduced in the context of Data Oriented Processing (Scha, 1990; Bod, 1992; Bod et al., 2003). FMs model complex data by considering how these are composed from data fragments of arbitrary sizes, up to regarding a complete data point as a single fragment. This modelling approach of FMs is extremely powerful, allowing to learn models with an arbitrary level of abstraction from the training data instances, leaving it to the learning algorithm to select the abstraction level which better generalises.

However, arriving at an abstraction level which will perform well when applying the model on yet unseen data is, almost by definition, extremely difficult for any learning objective or training algorithm based on the notion of 'fitting' the training data. The model space of FMs includes estimates which fit the training data so well that they essentially memorise them, while at the same time failing to anticipate novel data instances. Learning algorithms fitting the training set will return such degenerate estimates, leading to poor generalisation performance. Maximum Likelihood estimation and the EM algorithm fall in this category and for this reason, when applied straightforwardly, are of little use to estimate phrase-based SMT models and Fragment Models on the whole.

These issues led to the current trend of training phrase-based SMT models heuristically. Interestingly, these learning challenges are hardly new in the field of Natural Language Processing. They have been encountered before in the literature on the estimation of a natural language parsing model: Data Oriented Parsing (Prescher et al., 2004; Zollmann and Sima'an, 2006). In this thesis, we show how phrase-based SMT modelling is related to this prior literature and Fragment Modelling in general, and how all these models can benefit from the application of more appropriate learning approaches.

Learning to Generalise with CV-EM In order to formulate a solution, we will consider how these learning challenges touch upon a foundational problem in Machine Learning: addressing overfitting and estimating the potential of models to generalise, frequently understood in terms of the well-known bias-variance trade-off. Increasing the complexity of a model typically increases its ability to fit the training data, but also entails the danger of adapting to their particularities too closely, missing the underlying patterns. There has been a host of solutions proposed to alleviate this problem and find a good balance between fit and generalisation capacity. These include data-driven methods such as validation and cross-validation, methods employing Bayesian priors to counter overfitting or information theoretic approaches such as the Bayesian (Schwarz, 1978) and Akaike (Akaike, 1974) Information Criteria.

In this work, we revisit the problem of employing the EM algorithm for phrase-based and hierarchical translation models. We show that a heuristic solution to the estimation problems is not necessary and aim to unlock the potential of EM as an estimator for modern SMT models, by directly addressing the learning challenges involved. To do this, we opt for the data-driven Cross-Validation method. We integrate Cross-Validation within the Expectation-Maximization algorithmic framework to arrive at Cross-Validated EM (CV-EM): an instance of the EM algorithm which optimises model parameters according to a Cross-Validated Maximum Likelihood Estimation (CV-MLE) objective. The application of the CV-EM algorithm, which we present in Chapter 3, will have a crucial role in our empirical work. It will contribute in leading estimation away from the overfitting hypotheses over the value of the latent translation structure variables, and towards regions of the parameter space which appear to generalise well according to a cross-validation criterion.

Other chapters consider the development of learning methodologies centred around the CV-EM algorithm for a series of latent translation structure models of increasing complexity. Learning contiguous Phrase-Based SMT models in Chapter 4 addresses the challenges of disambiguating the segmentation of sentence-pairs in phrase-pairs. Chapter 5 builds upon this to proceed to learn phrase-based models assuming a relatively simple hierarchical translation structure. Finally, Chapter 6 introduces a methodology to induce models using a linguistically motivated abstract translation structure, taking advantage of cues related to the syntactic structure of language to explain the correspondences between the two sides of bilingual data.

1.4 Thesis Overview

We close this introductory chapter with an overview of the rest of the thesis. For each chapter, we describe the relevant research context and delineate our contributions, our key empirical findings and conclusions.

Chapter 2: The Crossroads Between Machine Translation and Machine Learning

In this chapter, we follow the crossing paths of Statistical Machine Translation and Machine Learning. We start by examining some of the modelling paradigms that have been influential on SMT research, such as the noisy-channel approach of Shannon (Shannon, 1948), and examine the contrast between generative and discriminative modelling of translation. We also consider a categorisation of translation modelling frameworks, according to their approach on abstracting away from the lexical surface, the nature of the assumed latent variables and the learning methodology applied.

We continue with a presentation of the SMT modelling frameworks that are relevant to this work, such as the IBM word-based SMT models (Brown et al., 1993), Phrase-Based SMT models (Och et al., 1999; Koehn et al., 2003; Marcu and Wong, 2002) and Hierarchical SMT (Wu, 1997; Chiang, 2005a). We pay particular attention to the modelling concepts behind the latent translation variables that each of these models assumes, such as word and phrase alignments, segmentation, reordering and translation hierarchical structure. For every modelling framework, we highlight its impact in the SMT literature and describe the learning challenges it introduced and how these were treated. In this way, we trace the progression of Machine Learning use in SMT research, which starts from the employment of well-founded estimation methods such as the EM algorithm, only to gradually resort to heuristic ad hoc solutions as the learning challenges mounted.

This thesis aspires to reconnect the learning methodology for modern phrase-based and hierarchical SMT models with the principled learning approaches employed in early work on SMT, in order to overcome the limits of the heuristic training methods. In the second part of the chapter, we build the theoretical background that will allow us to gain insights in the problems involved and that will provide the foundations for the learning methodology we propose to address them. We present the Expectation-Maximization algorithm, together with its crucial algorithmic and estimation properties. We proceed to examine the Bias-Variance decomposition of the Generalisation Error produced by a model and discuss the application of the Cross-Validation method to estimate it. The EM algorithm and Cross-Validation will form the two theoretical pillars under the novel learning algorithm we introduce in the next chapter: Cross-Validated EM.

Chapter 3: Fragment Models Estimation with the CV-EM Algorithm

The assumptions behind many modelling paradigms for complex, structured data, such as Markovian modelling or Bayesian Networks, can be understood to model data by examining how these are derived by combining together fixed-size data fragments. The Data Oriented Processing (DOP) paradigm (Scha, 1990; Bod and Scha, 1996) introduced the concept of Fragment Modelling: the derivation of data points from data fragments of *arbitrary* sizes, up to considering full data points themselves as single fragments. This is a modelling approach which is highly interesting for phrase-based SMT models assuming a latent translation structure. These too belong to the family of Fragment Models, with the data fragments for these models being contiguous or non-contiguous phrase-pairs, and the rest of the latent structure describing how these combine together.

In Chapter 3 we begin by examining the fragment-based DOP paradigm and its well-known implementation for natural language parsing, Data Oriented Parsing (Bod et al., 2003). We then abstract away from particular applications of DOP, to examine the implications of training Fragment Models using estimators which maximise model fit, such as Maximum Likelihood Estimation, extending earlier findings (Prescher et al., 2004; Zollmann and Sima'an, 2006). We consider why such training methods fail to produce estimates that generalise and discuss some of the alternatives proposed in prior literature.

In the second part of the chapter, we contribute a novel learning algorithm for Fragment Models: Cross-Validated Expectation-Maximization (CV-EM). Firstly, we examine the pitfalls related to the step of formulating a Fragment Model from the training corpus. During this step, copies of large segments of the training corpus are essentially integrated in the model space. This makes trivial and useless the crucial learning step of disambiguating between our hypotheses over the values of latent model variables by fitting the training data, as the hypotheses that will be preferred do nothing more than memorise the training corpus.

To address this, we introduce Cross-Validated Maximum Likelihood Estimation (CV-MLE), an estimation objective which cross-validates the hypotheses over the missing part of incomplete data to safeguard against hypotheses which do not generalise. We show how CV-MLE crucially retains many of the desirable estimation properties of plain MLE. We then contribute a practical implementation of the CV-MLE optimisation in terms of the CV-EM algorithm. We show that CV-EM is a true instance of the Expectation-Maximization algorithm and discuss its algorithmic and estimation properties and guarantees. We close with a comparison of CV-EM to prior research in model estimation and with an overview of what CV-EM has to offer for Fragment Model estimation.

Chapter 4: Learning Phrase-Pair Segmentation

Chapter 4 is the first of a series of three chapters which make up the second part of this thesis. They present our contributions on the learning of three distinct SMT model families of increasing complexity, each considering different assumptions on the form of the hidden latent translation structure.

In this chapter, we begin by contributing a method to learn the conditional translation probabilities of Phrase-Based SMT (PBSMT) models employing contiguous phrase-pairs, as a replacement for the heuristic estimators that are typically used. These probabilities are the central probabilistic component of PBSMT models and estimating their values essentially boils down to disambiguating how sentence-pairs segment into contiguous phrase-pairs.

Prior research had shown that a Maximum-Likelihood estimation objective as optimised by the EM algorithm performs considerably worse than the heuristic estimators (DeNero et al., 2006). We argue that this is not surprising, by showing that PBSMT models are instances of Fragments Models, and for this reason inherit the estimation problems that plague this model family. Even though the heuristic estimators already provide reasonable translation performance, we motivate the need for a better founded estimation methodology and describe how CV-EM can be applied instead to estimate the PBSMT model parameters.

Our approach is based on using the CV-EM algorithm to disambiguate sentence-pair segmentation by maximising the Cross-Validated conditional likelihood of each sentence in a sentence-pair given its counterpart, across both translation directions. Our algorithm explores a binary segmentation space where each phrase-pair either combines monotonically or swaps in relation to the neighbouring ones. Cross-validating this hypothesis space over segmentations using CV-MLE and

CV-EM contributes in overcoming the overfitting tendency of MLE and arrive at estimates which generalise well.

We evaluate our approach against a baseline employing a heuristic estimator for translation from French and German on one side, to English on the other. We find that our estimator performs at least on a par with the heuristic one, with some configurations even performing slightly better. These experiments showcase how the theoretically appealing properties of CV-MLE and CV-EM translate in competitive empirical results. This finding essentially invalidates the need for a heuristic estimator, as previously justified in the face of no access to alternatives which perform at least equally well.

Chapter 5: Learning Stochastic Synchronous Grammars

The previous chapter already introduced the use of a binary segmentation space. However, as our aim was to estimate the parameters of a PBSMT model, the models we examined did not venture further than the lexical surface. In this thesis, Chapter 5 marks the transition from such models assuming a flat latent translation structure, towards models which consider translation as a recursive process. The models we will examine are centred around a hierarchical latent translation structure variable. Their formulation is based on the binary subset of the stochastic Synchronous Context-Free Grammars (SCFGs), an extension of the Context-Free Grammars for parallel strings, where every production's right-hand side employs up to two bilingual non-terminals. These grammars combine the availability of algorithms to process them with a reasonable polynomial complexity, together with a high coverage of translation phenomena (Wu, 1997; Huang et al., 2009), with both features underlining their potential as foundations for formulating translation models.

Modelling with a synchronous grammar aims to take advantage of the recursive nature of language, as described by monolingual grammars, to capture the bilingual translation patterns. Still, the expressiveness of these models introduces new modelling and learning challenges.

Firstly, while the SCFG formalism seems superficially highly similar to its monolingual predecessor, by linking together the recursive structures of the source and target sentences, the result is more than the 'sum' of the two. It also specifies the syntactic element correspondences and the reordering patterns between the syntactic structures of the two languages. In this chapter, we discuss this, argue that an SCFG grammar must be designed with these issues in mind and contribute a design which addresses this, the 'switch' SCFG.

Secondly, the latent variable of stochastic SCFG models encapsulates several aspects of translation which previous models considered separately. While PBSMT models separate phrase segmentation from reordering, the latent hierarchical structure assumed by SCFG-based models must not only capture both of these core parts of the translation processes, but also their interdependence. As an example, such a latent variable must regulate the trade-off between a detailed multi-level hierarchical structure and the memorisation of longer phrase-pairs: a sentence-pair segmented in few, long phrase-pairs necessitates a relatively shallow hierarchical structure to explain how these combine together.

Chapter 5 considers if the CV-EM algorithm is able to effectively learn such latent variables despite these issues. First, we describe how models employing a phrase-based SCFG as their backbone fall into the Fragment Models family, motivating the use of the CV-EM algorithm for their estimation, and then describe an implementation of CV-EM for SCFGs. We then consider learning stochastic grammars based on two SCFG designs, a simple one reminiscent of the abstract structure employed by standard hierarchical SMT implementations (Chiang, 2005a), as well as one employing our 'switch' SCFG design. We test the CV-EM induced grammars for both designs against a standard hierarchical translation baseline on a translation task from French to English. We find that both designs offer translation performance on a par with the heuristically estimated baseline, with the 'switch' SCFG scoring better than the simpler variation.

Chapter 5 is crucial in examining the potential of the CV-EM algorithm to learn translation models that assume a latent translation structure which, in contrast to Chapter 4 is not directly attached to the observed lexical surface. By confirming that our learning methodology is able to also learn such latent variables, we prepare the grounds for the next chapter. There, we transition from the simple hierarchical structures examined in Chapter 5 towards a significantly more complex, linguistically motivated latent translation structure.

Chapter 6: Learning Linguistically Motivated Latent Translation Structure

While structure can be found in natural language when examining it at different syntactic and semantic levels, linguistic syntax is widely considered as one of its most salient properties. For this reason, the linguistic structure of sentences has been targeted for more than a decade as an informative data source that can lead to better translations. This has further turned the spotlight towards methods which take a syntactic but not necessarily linguistic approach to translation, such as SCFG-based approaches, as promising devices to model the dependence of the translation process on linguistic notions of natural language structure.

Nevertheless, even though a host of translation phenomena can be described in linguistic terms, it must be recognised that, overall, linguistic structure correlates with a mere subset of the transformations that take place between the two sides of a language pair (Dorr, 1994; Fox, 2002; Koehn et al., 2003). As a result, methods which assume that translation can be fully explained in terms of correspondences and transformations that are solely driven by the linguistic structure of sentences, frequently fail to deliver competitive translation performance, due to imposing unnecessary constraints on the translation process. The challenge is to find ways to take advantage of linguistic analyses when they are relevant for translation, while avoiding to be overly constrained by them when they are not.

In Chapter 6, we contribute a method to learn a linguistically *motivated* hierarchical translation model, by identifying the linguistic patterns which are informative for translation. We begin by constructing, for each training data sentence-pair, a chart covering with multiple linguistically motivated labels each aligned bilingual span. These labels are extracted from linguistic parses of the source sentence, where each of the multiple labels covering every span describes it from different linguistic perspectives and at varying levels of granularity. We then consider *all* binary structures which employ these labels to analyse the parallel training data, and use a translation-centric learning objective to disambiguate between them, according to their ability to explain the translation correspondences. This allows us to learn a model which is able to recursively analyse in linguistic terms the translation process across the whole sentence.

Our methodology builds on the foundations laid in the previous chapters. The synchronous recursive structure we consider, the Hierarchical-Reordering SCFG (HR-SCFG), is based on the principles behind the 'switch' SCFG of Chapter 5. The learning algorithm is an implementation of the Cross-Validated EM algorithm, as introduced in Chapter 3. In Chapters 4 and 5 we applied CV-EM for simpler translation models with most of their parameters directly relating to the lexical surface. Here, we separate the estimation of the lexical part of the model from the part related to the higher-level abstract hierarchical structure, and apply CV-EM to learn the latter: a linguistically motivated recursive structure which explains the correspondences and transformations between source and target sentences.

Crucially, contrary to other syntax-driven approaches (Way, 1999; Poutsma, 2000; Yamada and Knight, 2001; Galley et al., 2006; Huang et al., 2006; Liu et al., 2006), our method is linguistically motivated but not constrained. A translation-centric CV-MLE learning objective makes sure that only linguistically informed structures that help to explain translation are preferred, while the use of Cross-Validation aids in discovering those structures which are likely to generalise.

Other work (Marton and Resnik, 2008; Venugopal et al., 2009; Chiang et al., 2009) takes a more flexible approach, which is more similar to our own efforts. They opt to influence translation output using linguistically motivated features, or features based on source-side linguistically-guided latent syntactic categories (Huang et al., 2010). However, the features employed by these methods are *local* in nature, considering the linguistic plausibility of applying individual synchronous rules. As a result, these efforts totally lack the concept of a linguistically motivated hierarchical abstract *structure* reaching across the whole sentence-pair, which is exactly the focus of our own methodology.

The work of (Hassan et al., 2009) stands somewhat in the middle in comparison with fully syntax-driven SMT on the one hand and approaches using local syntax-based features on the other. Their system extends the Direct Translation Model of Ittycheriah and Roukos (2007) with dependency-grammar based syntactic features, and takes under account an incrementally built target language dependency structure. However, their system solely considers minimal phrase-pairs which translate single source words, and, while they reach further than other feature-based systems, their target-side syntactic analyses are eagerly constructed without reference to a globally optimal structure. In contrast, we specifically focus on the challenges involved with training phrase-based systems with many-to-many phrase correspondences, and search for the bilingual structures that best explain sentence-pairs in their entirety.

We complete the picture, by contributing a set of decoding techniques to efficiently and effectively translate using the latent translation structure model learnt by CV-EM. We find that the learnt models and our translation system provides statistically significant translation improvements, up to +1.92 BLEU score points, for four different empirical tasks, translating from English to French, German, Dutch and Chinese.

The results of Chapter 6 complete those of Chapters 4 and 5, to provide considerable evidence to back the key hypothesis of this thesis: models assuming a latent translation structure *can* be learnt under a clear learning objective, as implemented in terms of a well-understood optimisation framework and learning algorithm. The learnt models are able to provide real-world, competitive translation performance in comparison to heuristic training regimes, rendering the use of the latter unnecessary. Still, we believe that the true potential of our methodology is not in providing a reliable and effective substitute for these heuristic estimators. On the contrary, it lies in carving a path to the future, by making possible the estimation of powerful translation models that uncover the latent side of translation, and whose estimation under ad hoc algorithms would have been hardly possible.

Sources of the Chapters

Some chapters of this dissertation are partially based on the following publications or present experimental results that were first reported in them. The Cross-Validated Expectation-Maximization algorithm presented in Chapter 3 has first appeared in (Mylonakis and Sima'an, 2008) and is further discussed in (Mylonakis and Sima'an, 2010). Chapter 4 is partially based on material and results first included in (Mylonakis and Sima'an, 2008). Chapter 5 is similarly related to (Mylonakis and Sima'an, 2010), while Chapter 6 extends material first published in (Mylonakis and Sima'an, 2011).

Markos Mylonakis and Khalil Sima'an. 2008. Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 630–639, Honolulu, USA, October. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2010. Learning Probabilistic Synchronous CFGs for Phrase-Based Translation. In *Fourteenth Conference on Computational Natural Language Learning*, pages 117–125, Uppsala, Sweden, July. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2011. Learning Hierarchical Translation Structure with Linguistic Annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–652, Portland, Oregon, USA, June. Association for Computational Linguistics.

Chapter 2

Machine Translation and Machine Learning Concepts

Machine Translation (MT) on the one hand and Artificial Intelligence (AI) and Machine Learning (ML) on the other, have always followed crossing paths. MT has constantly figured as one of the most prominent fields of AI research, with the progress and disillusions on MT strongly affecting the appeal of AI, as highlighted during the period before and after the ALPAC report (Pierce et al., 1966).

The last two decades, MT has flourished as a result of the availability of more and cheaper computing power and the introduction of statistical models for Natural Language Processing (NLP). Most of the MT systems before this were based on translation lexica and fixed predefined rules which would 'fire' for a host of translation phenomena. In contrast, the statistical approach is centred around the formulation of stochastic translation models and training these models on corpora. The transition towards explaining the translation process through a statistical model crucially allowed tapping into the wealth of Machine Learning research in statistical estimation. Furthermore, it also contributed to the introduction of novel Machine Learning approaches motivated by the challenges posed by the MT models, such as the large number of parameters, the interplay between memorising and generalising, dealing with yet unseen events and others. This thesis follows this trend by contributing MT solutions through exploring and proposing novel approaches on fundamental learning problems.

The pioneering work on Statistical Machine Translation (SMT) in the IBM labs (Brown et al., 1990) introduced *word-based* statistical translation models. From there on, the major steps in the SMT literature involve models translating contiguous phrases together (Och et al., 1999; Koehn et al., 2003) and later recursive translation employing phrasal patterns with gaps (Chiang, 2005a). Recent developments focus on employing hierarchical structure for MT, often taking advantage of monolingual syntactic analyses, e.g. (Galley et al., 2004; Zollmann and Venugopal, 2006; Mylonakis and Sima'an, 2011).

As the MT models become more complex however, the stress on the associated

ML methods used to train them and translate with them is increased. Training the IBM models already relied on approximations for the more complicated models. The subsequent step towards models employing phrases, hierarchical or not, was also marked by a transition to heuristic estimation, making less clear how the estimates relate to the training corpus. This is not without reason: it is notoriously difficult to estimate such models. We believe however that moving away from relying on hand-crafted arbitrary heuristics and towards well-founded estimation is fundamental when progressing to even more complex models involving rich latent MT structure, in the same way that leaving behind hand-crafted translation rules was important in the early years of statistical MT. This has recently proved a very vibrant research direction and part of the work in this thesis is occupied with this topic.

In this chapter we will first present the basic concepts of the key frameworks and models for Machine Translation that play a role in this thesis. We begin by presenting the IBM word-based SMT models, and continue with a discussion of phrase-based and hierarchical SMT. In the second part, we will focus on the two Machine Learning methods which form the backbone of the learning approach contributed by this work. We will first examine the basics of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), a key and powerful statistical estimation algorithm that has already been widely and successfully employed for MT (Brown et al., 1993). Nevertheless, the application of EM on phrase-based models exposed their strong tendency to overfit the training data and generalise poorly. Keeping this in mind, we subsequently close the chapter by introducing Cross-Validation (CV), a well-understood method to estimate the generalisation error of model estimates. In the following chapter we will show how EM and CV can be combined towards MT model estimation specifically aiming towards strong generalisation over yet unseen data.

2.1 Modelling Machine Translation

The branch of Machine Translation where a high proportion of current MT research is directed and on which this work focuses is Statistical MT. Given a source language sentence $\mathbf{f}$, the fundamental problem in MT is to produce its target language translation $\mathbf{e}$ by means of a computer program. Output $\mathbf{e}$ must both sufficiently convey the *meaning* of the original sentence $\mathbf{f}$, as well as enjoy target language *fluency*. SMT aims to achieve this through the application of statistical models. By introducing a probability distribution $p(\mathbf{e}|\mathbf{f})$, assigning to every target sentence $\mathbf{e}$ a probability of being the translation of source input $\mathbf{f}$, an SMT system outputs the target sentence $\hat{\mathbf{e}}$ with the highest conditional probability:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) \tag{2.1}$$

Building an SMT system can be mostly divided in three parts. Firstly, it involves *designing the model* $p(\mathbf{e}|\mathbf{f})$. Some of the questions here might be what kind of translation phenomena does it capture and how does it capture them, what are the parameters and which latent variables are assumed. Model design plays a crucial role in SMT, as it defines the rules of the game: what needs to be learnt from the training corpora and later applied to actually translate, according to the modellers view of translation. After the model is set, we need to *train* it, select the model instance which is best according to some learning objective, by employing training data possibly coupled with prior knowledge. This entails the usage of a statistical *estimator*. The final step, *decoding*, employs the trained model estimate to actually translate by selecting for every input $\mathbf{f}$ the translation $\hat{\mathbf{e}}$ according to equation (2.1).

2.1.1 The Noisy Channel Approach

Shannon's noisy channel (Shannon, 1948) has been an influential paradigm for SMT (Brown et al., 1990). Instead of directly modelling target sentences given source input, we consider the target sentence as a message which got corrupted while being transmitted through a translation communication channel, resulting in the source sentence. Our objective is to retrieve this original message. We use Bayes law to rewrite the search objective of (2.1) in the equivalent formulation below, separately modelling the language of the target sentences and the corruption of these sentences when translated from target to source.

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_{\mathbf{e}} \quad \overbrace{p(\mathbf{f}|\mathbf{e})}^{\text{translation model}} \qquad \overbrace{p(\mathbf{e})}^{\text{language model}} \end{aligned} \tag{2.2}$$

This crucially splits the modelling effort in a stochastic component focused on translation correspondence, the Translation Model (TM), and a component exclusively occupied with output well-formedness, the Language Model (LM). Each of these models is then occupied with one of the two key objectives of the translation system's output outlined above: meaning correspondence and fluency. Considering these two notions apart avoids modelling all aspects of translation at once, letting the TM focus on the transformations that take place during translation while the LM attends to output fluency. In addition, it also allows employing different resources for training. While the translation model usually requires more expensive bilingual data to train, language model training only demands monolingual data which are cheaper to assemble in large quantities.

Early SMT work, such as the IBM models later discussed in this chapter, applied the Noisy Channel paradigm in a relatively literal fashion. However, translation adequacy and fluency can in practice hardly be considered separate. Malformed target output cannot appropriately convey the meaning of the source sentence; an adequate translation would probably be expected to also be relatively well-formed. Subsequent SMT research deviated from a strict reading of the Noisy Channel approach, regarding the language model probability of the target sentence as just one of the elements considered to assess the overall translation probability, together with other, more bilingual in nature, translation features.

2.1.2 Generative and Discriminative Models

Generative translation models capture the stochastic joint generation of source and target sentence pairs. They can also straightforwardly be employed to select the translation $\mathbf{e}$ with the highest probability given $\mathbf{f}$, as with $\mathbf{f}$ fixed we have:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} p(\mathbf{e}, \mathbf{f}) \tag{2.3}$$

These models are usually based on a generative process tracking the steps to emit the tuple $\langle \mathbf{e}, \mathbf{f} \rangle$. For example, we might begin by considering the generation of corresponding source and target word-pairs following the word order of the source language, subsequently reordering the target language words to form the target sentence. Each of the generative steps is modelled by a separate distribution conditioned on the previous steps, often under independence assumptions which simplify the modelling effort. Some conditional translation models $p(\mathbf{e}|\mathbf{f})$ are formulated in a similar fashion, emitting $\mathbf{e}$ from $\mathbf{f}$ under a generative process (Brown et al., 1993).

Generative models require extensive effort to consider all the steps and transformations that take place during translation, as well as to introduce independence assumptions taking into account the available training data (e.g. to avoid overfitting) or computational limitations etc. In contrast, *discriminative* modelling directly models the conditional distribution $p(\mathbf{e}|\mathbf{f})$, instead of putting effort towards formulating a full generative process emitting samples $\langle \mathbf{e}, \mathbf{f} \rangle$. For MT, this typically happens through employing *feature functions* $\phi_i(\mathbf{e}, \mathbf{f})$, each assigning a non-negative score examining the two sentences from a different perspective, e.g. word or phrase correspondence, output fluency (frequently the LM score), target word reordering and others. The modeller does not need to consider a coherent generative story but only what kind of features could be useful in discriminating between strong and weak translations. These scores are weighted together log-linearly with weights λ_i and normalised to obtain the conditional translation model (Och and Ney, 2004).

$$p(\mathbf{e}|\mathbf{f}) = \frac{1}{Z_\lambda(\mathbf{f})} \exp\left(\sum_i \lambda_i \phi_i(\mathbf{e}, \mathbf{f})\right) \tag{2.4}$$

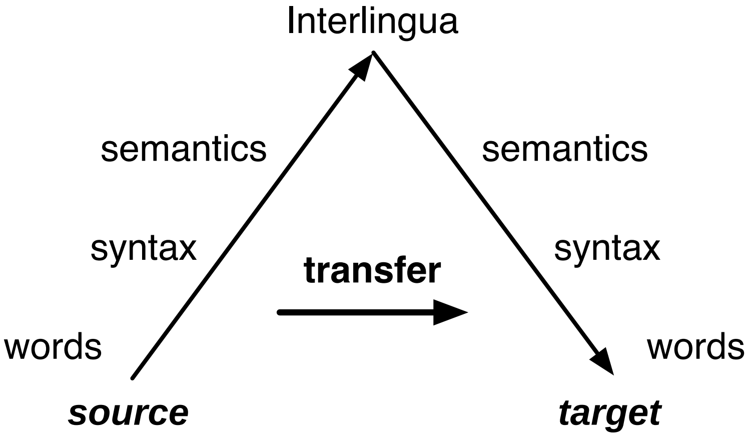


Figure 2.1: The Machine Translation pyramid

The step of estimating the feature weights is crucial. Possible training objectives are constrained entropy maximisation (Berger et al., 1996) and error rate minimisation according to a translation quality metric (Och, 2003).

Crucially, if we consider selecting the translation **e** with the highest probability as a classification task, while for many machine learning tasks the class space is relatively constrained, in MT (and NLP in general) the class space is very large or even countably infinite. For this, frequently a generative process is still needed as part of a translation system based on a discriminative model, to supply the set of target sentence translations that will be scored by the model to select the most probable one, sometimes producing also a score embedded as a feature function of (2.4). The latter is the case for all phrase-based and hierarchical translation approaches later discussed in this chapter.

One disadvantage of discriminative MT models is that it is more difficult to introduce and train the parameters for *latent* variables in the model, such as latent structure which is not part of the observed training data. In this thesis we take a hybrid approach. We first train a generative model employing latent translation variables, which is afterwards included as both a feature function and a generative process backbone for a discriminative translation model.

2.1.3 Model Categorisation

Apart from the probabilistic formalisation approach that they follow as discussed above (e.g. generative, discriminative, hybrid), MT models can be also categorised according to the, often latent, *abstraction* from the lexical level that they employ. The familiar MT pyramid in Figure 2.1 (Vauquois, 1968) presents a view on the different levels of abstraction. At the most basic level, MT models operate directly on the lexical surface, translating and reordering based on lexical

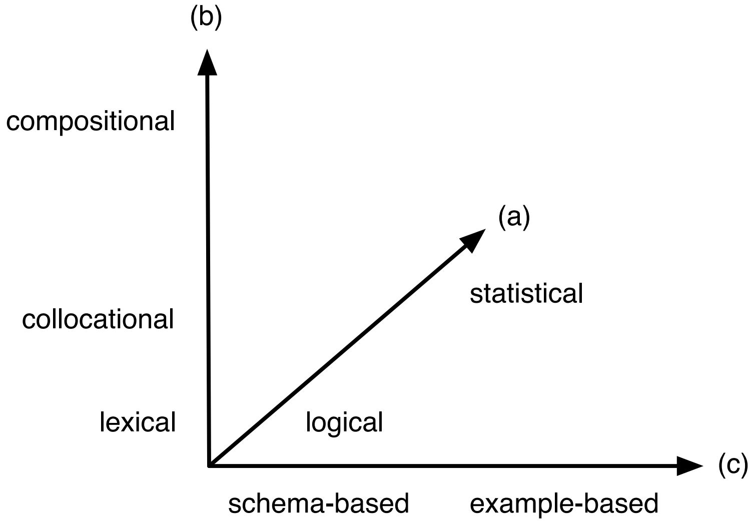


Figure 2.2: 3-axes categorisation of Machine Translation models

cues. Moving up the pyramid we find models which utilise syntactic and semantic categorisations and representations, with a transfer step modelling the transformation of this representation from source to target form. Finally, at the top of the pyramid stands the Interlingua approach, which is based on constructing an internal, natural language independent abstraction of the full meaning of the source sentence and subsequently building the target language from it.

Historically, MT research has followed an interesting pattern exploring the MT pyramid. The early approaches on MT, starting already from the IBM-Georgetown demonstration, emphasised the employment of grammatical abstractions and rule-based transfer steps between source and target language, while at the same period the Interlingua approach was quite influential. Since the advent of Statistical MT from the late 80's and onwards, most state-of-the-art MT systems (e.g. (Brown et al., 1993; Och et al., 1999; Koehn et al., 2003)) directly modelled translation on the lexical level, with this trend lasting for almost two decades. Recently, there have been considerable research efforts of increasing sophistication on syntactical approaches on SMT (e.g. (Chiang, 2005a; Zollmann and Venugopal, 2006)), finally delivering state-of-the-art performance, particularly for language pairs with heavy reordering such as English-Chinese.

Wu (2005) introduces a 3-axes system to categorise MT models, presented in Figure 2.2. Axis (a) examines if the model is mostly based on mathematical logic, or if it makes substantial use of statistics and probabilities. All the models we examine in this thesis are statistical MT models. The second axis (b) relates to the degree of recursion in the model. At the bottom stand lexical-based models like the IBM models, while moving up the axis we find collocational models such as those employed in phrase-based SMT and finally fully compositional models such as those backed by Synchronous Context Free Grammars. The models contributed in later chapters of this work fall in these last two categories along axis (b).

Axis (c) considers if the model is based on abstracting versus memorising the training data. In the first case, an abstraction such as a generative translation model is built during training, which is later employed during test time to translate. In contrast, example-based MT (Nagao, 1984) relies on memorising the training data (examples) and reusing these to translate by breaking them down, adapting and recombining them according to the input source sentence. Recent SMT systems blur the line between statistical model estimation (schema-based MT) and memorisation (example-based MT). For example, while Data Oriented Translation (Poutsma, 2000) memorises aligned fragments of syntactic trees of source and target sentences, it also learns a probabilistic model that describes how to combine them together to derive sentence-pairs. Instead of training a hierarchical translation model prior to test time, (Lopez, 2007) memorises the training parallel corpus and extracts and scores recursive translation rules separately for every input sentence during test time. Furthermore, phrase-based models, whether they are recursive in nature or not, memorise parts of the training data. In this thesis, inspired by Data Oriented Processing (Bod and Scha, 1996), we take this further by opting for an *all phrase-pairs* approach, extracting and memorising all corresponding phrases of the training parallel corpus, up to the whole sentence-pair.

2.2 Word-Based Translation

Statistical MT was introduced by researchers from the IBM T.J. Watson research centre in the late 80's (Brown et al., 1990). This first attempt at SMT was further refined in the formulation of a succession of word-based translation models of increasing complexity, the IBM translation models (Brown et al., 1993). The models are founded on a Noisy-Channel approach to translation, as discussed in section 2.1.1 above. The translation process that is being modelled is inverted, so that we introduce a target-to-source translation model $p(\mathbf{f}|\mathbf{e})$, as well as a language model component $p(\mathbf{e})$ over the target language. This relieves the translation model from the task of concentrating probability mass on well-formed output sentences, as would be required by a direct translation model $p(\mathbf{e}|\mathbf{f})$. This task is assigned to the language model instead.

The authors recognise the three foundational problems in SMT: (a) estimating the *language model* probabilities $p(\mathbf{e})$ over target language sentences $\mathbf{e}$, (b) estimating the *translation model* probabilities $p(\mathbf{f}|\mathbf{e})$ from target to source and (c) *decoding*, i.e. searching for $\hat{\mathbf{e}}$ which maximises the product of the two as in equation (2.2). SMT took advantage of the existing research on language mod-

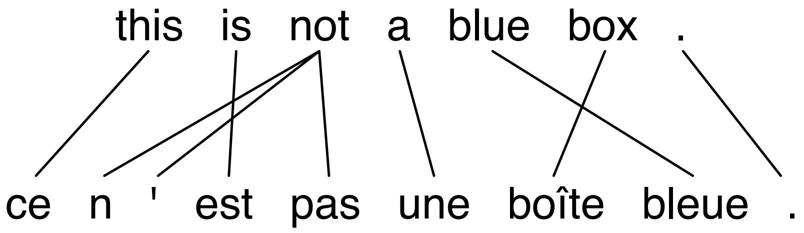


Figure 2.3: An alignment between English and French

elling by the speech processing community[1]. Hence, attention was drawn to the translation model, while the highly non-trivial task of decoding efficiently forms a research direction of its own which we do not treat extensively in this thesis.

2.2.1 Alignments

The key concept introduced by the IBM models are word *alignments*, links between words which are considered translations of each other. For the parallel sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$, $\mathcal{A}(\mathbf{e}, \mathbf{f})$ is the set of word-position pairs of aligned words between the two sentences. Using French to English examples, these links can in general connect a single source and target word (*the* - *le*), a single source to multiple target words (*pick up* - *ramasser*), a single target to multiple source words (*implemented* - *mis en application*); or, considering every source word aligned to every target word, multiple source to multiple target words (e.g. idioms such as *take the trouble* - *prendre la peine*). The commonly aligned words need not be contiguous (e.g. *not* - *ne* ___ *pas*). An example of an aligned sentence pair can be seen in Figure 2.3.

To simplify matters, the IBM models considered only alignments where every source word f is at most aligned to a single target word e. As in Figure 2.3, this can nevertheless result in multiple target words being aligned to the same source word. For a source sentence $\mathbf{f} = f_1^m$ of length m and a target sentence $\mathbf{e} = e_1^l$ of length l, the alignment variable is then defined as the vector $\mathbf{a} := a_1^m$, with a_j the position of the target word that f_j is aligned to. A source word is also allowed to remain unaligned, in which case we consider it aligned with an additional empty token at target word-position zero.

The alignment $\mathbf{a}$ between $\langle \mathbf{e}, \mathbf{f} \rangle$ is a *latent* variable, given that such information is not normally part of a parallel training corpus. The translation probability must thus sum over all values of $\mathbf{a}$.

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{f}, \mathbf{a}|\mathbf{e}) \tag{2.5}$$

[1] See (Chen and Goodman, 1998) for an overview of most models employed up to this day.

2.2.2 Translation Models

With the help of the alignment variable and without loss of generality, we can write employing the chain rule:

$$p(\mathbf{f},\mathbf{a}|\mathbf{e}) = \overbrace{p(m|\mathbf{e})}^{\text{source length}} \prod_{j=1}^{m} \overbrace{p(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})}^{\text{current alignment point}} \; \overbrace{p(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e})}^{\text{current lexical choice}} \tag{2.6}$$

In the generative process of the equation above, first the length of the $\mathbf{f}$ sentence is sampled from $p(m|\mathbf{e})$. Subsequently, in series from left to right in the order of $\mathbf{f}$, the alignment point for the current source position is established given the previous alignments and source words, as well as m and the $\mathbf{e}$ sentence. Finally, the current source word is generated given a similar conditioning variable set with the addition of the current alignment point established in the previous step.

Working with such detailed conditioning contexts as in (2.6) leads to computational difficulties in estimation and can be prone to overfitting. The succession of models in (Brown et al., 1993) are the result of applying different sets of assumptions simplifying equation (2.6).

IBM Model 1 In the simplest of the translation models, $p(m|\mathbf{e})$ is assumed to be independent of both m and $\mathbf{e}$ and thus equal to a constant ϵ. For every f_j word, its alignment is sampled uniformly from the l word positions of $\mathbf{e}$ plus the option of aligning to empty and is therefore $(l+1)^{-1}$. Finally, lexical selection takes place conditioning only on the e_{a_j} target word that f_j is aligned to. Given these assumptions, (2.6) can be rewritten as:

$$p(\mathbf{f},\mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} p(f_j|e_{a_j}) \tag{2.7}$$

Equation (2.7) establishes a foundational parameter set in MT models, the conditional translation probabilities $p(f|e)$, as well as a key data structure of MT systems: the *translation table* holding these probabilities for every $\langle e, f\rangle$ pair.

IBM Model 2 Model 1 does not occupy itself with the word order in the two strings as dictated by the alignment points and any reordering of the f words is assigned an equal translation probability. Model 2 introduces non-uniform alignment probabilities $p(a_j|j,m,l)$, by conditioning each alignment point on the word position of the word being aligned. This allows preferring when translating from French to English similar word positions between the two languages, as is often the case in human translations. Under the assumptions of Model 2, (2.6) becomes:

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} p(a_j|j, m, l) p(f_j|e_{a_j}) \tag{2.8}$$

Model 2 introduces a further prominent component of MT models, namely a non-trivial *reordering model* aiming at distinguishing between good and weak reorderings of translated, in this case lexical, sentence components.

IBM Models 3 to 5 Even though the IBM models are founded on a word-based view of translation, many translation phenomena involve more than single words in each of the two languages, with some examples already discussed in section 2.2.1. Models 3 to 5 venture to capture the translation of single target words e into multiple source words. The tendency of certain e words to align to more than a single source word is modelled through target word *fertility* distributions, providing the probability of a given number of alignments leading to e. The reordering models then allow capturing a preference for these commonly aligned f words to be clustered together in the word order of sentence $\mathbf{f}$.

HMM Alignment Model A different approach to the same problem is found in (Vogel et al., 1996). The Hidden Markov Model (HMM) for word alignment treats the clustering of translated words by modelling alignment as a Hidden Markov process.

$$p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \prod_{j=1}^{m} p(a_j|a_{j-1}, l) p(f_j|e_{a_j}) \tag{2.9}$$

The latent alignment variable is modelled in a Markovian fashion, with every alignment point conditioned on that of the previous source word under distribution $p(a_j|a_{j-1}, l)$. Target words are then emitted conditioned on the source word that they are aligned to, as in the IBM models. This allows modelling in a simpler fashion the movement of clusters of translated target words in the source sentence than the IBM models.

While a detailed examination of these last models is beyond the scope of this thesis, it is important to note that the need to model phrasal translations was already recognised in word-based approaches, attempting to address this problem through word-based modelling steps. Phrase-based translation, discussed later in this chapter, capitalises on the advancements in word-based MT to directly model phrasal translation phenomena.

2.2.3 Estimation

The parameters θ for these word-based translation models are trained with Maximum-Likelihood Estimation (MLE) on a training parallel corpus $\mathcal{X}$ of sentence pairs

$\langle \mathbf{e}, \mathbf{f} \rangle$. Maximising the likelihood $\mathcal{L}$ of the corpus to retrieve MLE estimate $\hat{\theta}$, boils down to maximising the conditional translation probability of independently emitting source sentences given target sentences, as the language model probabilities are constants given the sentence pairs.

$$\mathcal{L}(\mathcal{X};\theta) = \prod_{\langle \mathbf{e},\mathbf{f} \rangle} p(\mathbf{e},\mathbf{f};\theta) = \prod_{\langle \mathbf{e},\mathbf{f} \rangle} p(\mathbf{e})p(\mathbf{f}|\mathbf{e};\theta) \tag{2.10}$$

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\mathcal{X};\theta) = \arg\max_{\theta} \prod_{\langle \mathbf{e},\mathbf{f} \rangle} p(\mathbf{e})p(\mathbf{f}|\mathbf{e};\theta) = \arg\max_{\theta} \prod_{\langle \mathbf{e},\mathbf{f} \rangle} p(\mathbf{f}|\mathbf{e};\theta) \tag{2.11}$$

All the word-based translation models of this section interpret translation as the latent alignment variable $\mathbf{a}$. Using equation (2.5), we rewrite the search for the MLE estimate as:

$$\hat{\theta} = \arg\max_{\theta} \prod_{\langle \mathbf{e},\mathbf{f} \rangle} \sum_{\mathbf{a}} p(\mathbf{f},\mathbf{a}|\mathbf{e};\theta) \tag{2.12}$$

For all the models discussed, solving this optimisation problem cannot be addressed analytically. Instead, after initialising the parameter set, an instance of the Expectation Maximisation algorithm iteratively climbs the likelihood function returning a series of estimates, each increasing the training data likelihood until a local maximum is reached. The models presented here largely succeed each other by refining the parameter set. For this, they are usually trained in a pipeline with the more complicated models initialising some of their parameters using the estimates of simpler models.

Even though EM does not usually manage to find the global maximum of the likelihood function with respect to the model parameters, the word-based translation models were noted for employing a clear learning objective during training. This is a property not shared by the latter, phrase-based models discussed next.

2.2.4 The Word-Alignment Task

Overall, despite the fact that the word-based models are complete translation models, they were little used for translation proper. Instead, they were repurposed to *word-align* parallel corpora, i.e. retrieving for every sentence pair the alignment $\hat{\mathbf{a}}$ which maximises $p(\mathbf{f},\mathbf{a}|\mathbf{e})$. Since all the models discussed are based on the assumption that every f aligns to a single e word, more complicated alignment patterns can be attained by aligning across both translation directions, computing $\hat{\mathbf{a}}_1$ for target to source and $\hat{\mathbf{a}}_2$ for source to target. A function of $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ can then be employed to arrive at a hopefully more comprehensive alignment set, which needs not obey the constraints imposed on the alignment space by the IBM models (one alignment point per source word). Choices for this are the intersection of the two alignment sets, their union, or heuristic functions to retrieve an alignment set between intersection and union (Och et al., 1999; Och and Ney, 2004).

The usage of word-based models for word-alignment was and still remains crucial for the development of improved translation models . All but one (the Joint Translation Model (Marcu and Wong, 2002)) of the MT models presented in the rest of this chapter utilise training methods which assume the word-alignments given, which in practice relates to alignments induced from the parallel corpus using word-based translation models.

2.3 Phrase-Based Translation

Many translation phenomena span across multiple words. Examples include idiomatic expressions, agreement between words, meaning differences according to the surrounding context and local reordering. Modelling such occurrences through word-based means often leads to awkward models that are difficult to train and decode with.

This motivated modelling phrasal translations directly, by means of memorising phrasal translation fragments and learning a model employing them to translate. In this section we will restrict ourselves to approaches based on *contiguous* phrases, while the next section treats *hierarchical* phrasal translation through a recursive process. We first discuss conditional probability phrase-based models, whose estimation is largely based on heuristics. We then subsequently present a method that has been proposed to estimate phrase translation probabilities with a clearer objective function: a joint probability phrase-based model.

2.3.1 Conditional Log-Linear Models

While phrase-based models were already introduced in earlier work such as (Wang and Waibel, 1998; Och and Weber, 1998), modern Phrase-Based SMT (PBSMT) traces its origins in the alignment template approach (Och et al., 1999; Och and Ney, 2004). This original formulation was also based on bilingual word classes (Och, 1999). As this feature is not widely adopted today as part of phrase-based models, we drop it from the presentation below for clarity reasons.

Alignment Templates Assuming an already word-aligned corpus, an Alignment Template (AT) is a triple $z = \langle \tilde{e}, \tilde{f}, \tilde{a} \rangle$, corresponding to a bilingual phrase-pair together with the alignment points between the phrases' words. In other words, an alignment template is memorising a contiguous bilingual phrase-pair together with the internal word-alignment between the two phrases. Some examples of ATs, covering a few of the phrasal translation phenomena mentioned earlier such as local reorderings and multi-word expressions, can be seen in the

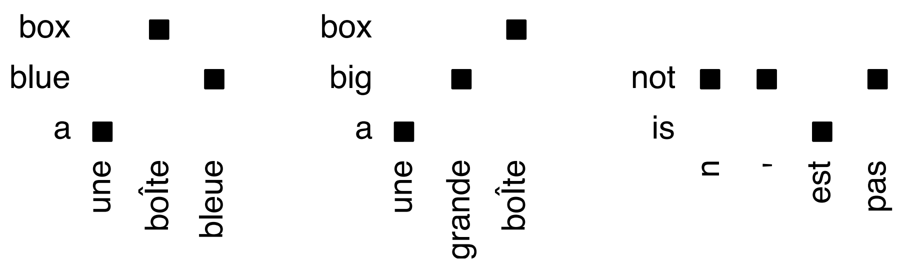


Figure 2.4: Examples of alignment templates

matrices of Figure 2.4, where blocks indicate an alignment point between the words on the two axes.

To arrive at translation **e** given source **f** employing ATs as building blocks, we need three latent variables, each governing source sentence segmentation, AT application and target sentence reordering, as can be seen in Figure 2.5. Firstly, the source sentence is split in K contiguous phrases according to the value of the *segmentation* variable $\boldsymbol{\sigma}$, choosing from the set of all possible segmentations $\boldsymbol{\Sigma}(\mathbf{f})$. Then, for each phrase $\tilde{f}$ of the segmented input, a sequence $\mathbf{z} = z_1^K$ of ATs is applied. Each AT $z_i = \langle \tilde{e}, \tilde{f}, \tilde{a} \rangle$ has $\tilde{f}_i$ as its source phrase and leads to the phrase's translation $\tilde{e}$ as well as establishes the alignment points $\tilde{a}$ between them. Finally, a reordering $\boldsymbol{\pi} = \pi_1^K$ of the ATs' target sides positions with respect to the source determines the word order of the output. The conditional translation probability of an aligned sentence-pair within the AT model could then be written as a conditional generative process.

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = \sum_{\boldsymbol{\sigma} \in \boldsymbol{\Sigma}(\mathbf{f})} \overbrace{p(\boldsymbol{\sigma}|\mathbf{f})}^{\text{segmentation}} \; \overbrace{p(\mathbf{z}|\boldsymbol{\sigma}, \mathbf{f})}^{\text{AT application}} \; \overbrace{p(\boldsymbol{\pi}|\mathbf{z}, \boldsymbol{\sigma}, \mathbf{f})}^{\text{reordering}} \tag{2.13}$$

If we assume that the AT corresponding to each of the phrases of the segmented **f** sentence is applied independently, we have:

$$p(\mathbf{z} = z_1^K|\boldsymbol{\sigma}, \mathbf{f}) = \prod_{k=1}^{K} p(z_k = \langle \tilde{e}_k, \tilde{f}_k, \tilde{a}_k \rangle | \tilde{f}_k) \tag{2.14}$$

Heuristic Estimation An important point is that, in order to estimate the parameters of $p(z|\tilde{f})$ above and those of the distributions in equation (2.13) in general, the training corpus must either be already segmented in phrase-pairs or we must disambiguate between the possible segmentations of each sentence-pair. However, parallel corpora are normally not segmented and, as discussed in (DeNero et al., 2006; Mylonakis and Sima'an, 2010), estimation of the AT model's distributions is prone to overfitting. For this, parameter estimation for the AT

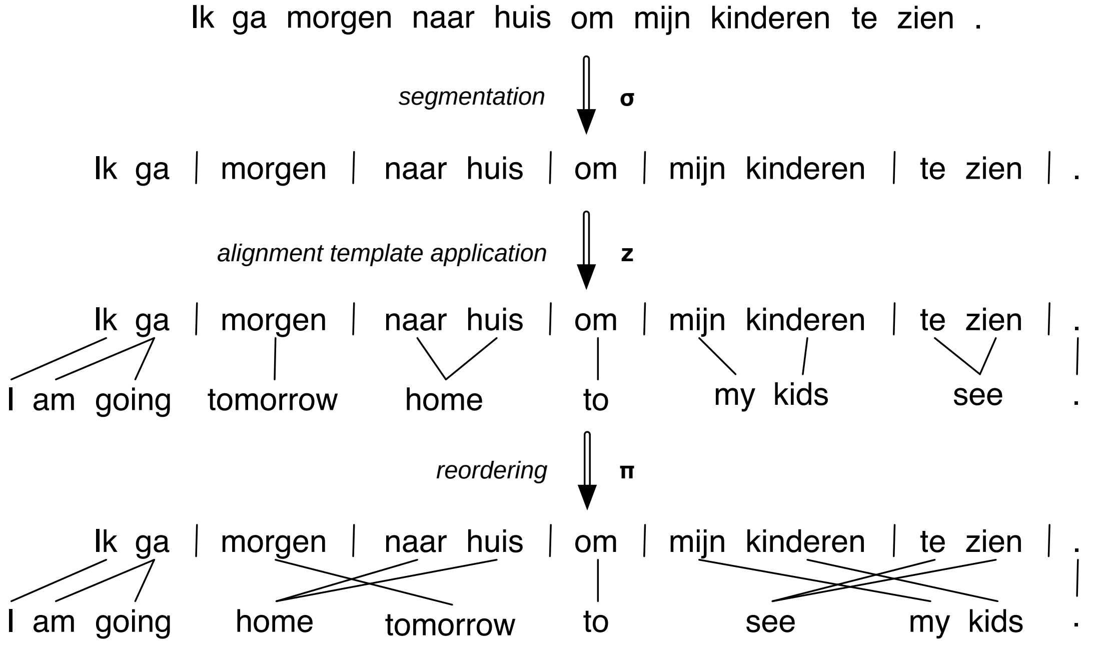

Ik ga morgen naar huis om mijn kinderen te zien .
segmentation
σ
Ik ga | morgen | naar huis | om | mijn kinderen | te zien | .
alignment template application
z
Ik ga | morgen | naar huis | om | mijn kinderen | te zien | .
I am going tomorrow home to my kids see .
reordering
π
Ik ga | morgen | naar huis | om | mijn kinderen | te zien | .
I am going home tomorrow to see my kids .

Figure 2.5: The alignment template approach latent variables

model takes place heuristically, disregarding the latent segmentation variable $\boldsymbol{\sigma}$.

A multiset of ATs is constructed from the word-aligned parallel corpus, by *extracting* ATs under the following heuristic rule: an alignment template is extracted once for every pair of $\langle \tilde{e}, \tilde{f} \rangle$ phrases with (a) at least one alignment point between words of the phrases and (b) all alignment points are contained within the phrase-pair, i.e. there are no alignment points from words of the phrase-pair leading to words outside the phrase pair. With $C(z)$ counting the number of times a particular AT was extracted, the conditional probability of the template given its source phrase is defined as:

$$p(z = \langle \tilde{e}, \tilde{f}, \tilde{a} \rangle | \tilde{f}) = \frac{C(z)}{\sum_{\tilde{e}', \tilde{a}'} C(\langle \tilde{e}', \tilde{f}, \tilde{a}' \rangle)} \tag{2.15}$$

Crucially, the above is *not* the MLE estimate when training on the parallel corpus, but one that uses the counts in the heuristically extracted ATs multiset. As a consequence of its heuristic nature, this estimate is not known to optimise any meaningful function of the training parallel corpus itself. As these estimates have little to do with the formulation of equation (2.13), they are instead employed as features in a log-linear conditional translation model

Log-Linear Model Formulating the AT model as a log-linear, feature-based model allows for the easier integration of additional translation features examining translation quality from different perspectives. Each feature $\phi(\mathbf{e}, \mathbf{f}, \mathbf{z}, \boldsymbol{\pi})$ assigns a non-negative score to the construction of $\mathbf{e}$ from $\mathbf{f}$, using ATs $\mathbf{z}$ under reordering $\boldsymbol{\pi}$. The features are weighted together log-linearly under weights λ, arriving at the following conditional translation model, with $Z(\mathbf{f})$ as the normalisation constant:

$$p(\mathbf{e}, \mathbf{z}, \boldsymbol{\pi}|\mathbf{f}) = \frac{1}{Z(\mathbf{f})} \exp \sum_i \lambda_i \; \phi_i(\mathbf{e}, \mathbf{f}, \mathbf{z}, \boldsymbol{\pi}) \tag{2.16}$$

While in principle marginalising out the latent variables $\mathbf{z}$, $\boldsymbol{\pi}$ is needed to arrive at the conditional translation probability $p(\mathbf{e}|\mathbf{f})$, this is in practice computationally prohibitive. Decoding is instead recast as a Viterbi search for the AT application and reordering which maximises the probability of (2.16):

$$\langle \hat{\mathbf{e}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\pi}} \rangle = \underset{\mathbf{e},\mathbf{z},\boldsymbol{\pi}}{\arg\max} \sum_i \lambda_i \; \phi_i(\mathbf{e}, \mathbf{f}, \mathbf{z}, \boldsymbol{\pi}) \tag{2.17}$$

A key feature employed in the AT model (Och and Ney, 2004) is the logarithm of the conditional probability of independently selecting each alignment template as in equation (2.14), according to the heuristic scores of (2.15). This is complemented by features examining word correspondence within every template through the template's alignment pattern, as well as a word penalty feature counting the number of target words produced, regulating target output length. A simple target phrase-reordering model based on word-position movement of target phrases provides the means to prefer mostly monotonic phrase alignments, which largely preserve the order of the source sentence as is the case for many European language pairs. Finally, the language model score of the target output $\mathbf{e}$ is added as an additional feature focusing on target sentence well-formedness.

Phrase-based SMT The original formulation of the alignment template models in (Och et al., 1999; Och and Ney, 2004) put an emphasis on the alignment between phrase-pairs being an integral part of the memorised fragments extracted from the word-aligned training corpus. Zens et al. (2002) and (Koehn et al., 2003) depart from this view, to formulate a phrase-based SMT model. Founded on a simplified version of the assumptions of the alignment template approach, it extracts only *phrase-pairs* $\langle \tilde{e}, \tilde{f} \rangle$ where the AT approach would extract full phrasal alignment templates. The conditional phrase translation probabilities $p(\tilde{e}|\tilde{f})$ as well as $p(\tilde{f}|\tilde{e})$ are trained under a similar extraction heuristic as in (2.15).

$$p(\tilde{e}|\tilde{f}) = \frac{C(\langle \tilde{e}, \tilde{f} \rangle)}{\sum_{\tilde{e}'} C(\langle \tilde{e}', \tilde{f} \rangle)} \qquad p(\tilde{f}|\tilde{e}) = \frac{C(\langle \tilde{e}, \tilde{f} \rangle)}{\sum_{\tilde{f}'} C(\langle \tilde{e}, \tilde{f}' \rangle)} \tag{2.18}$$

This leads again to a log-linear translation model as in (2.16), this time employing features $\phi(\mathbf{e}, \mathbf{f}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi})$. Translating $\mathbf{f}$ under the model is performed through a Viterbi search on the space of all constructions of target sentences by applying and reordering phrase-pairs to a segmentation of $\mathbf{f}$, similarly to (2.17).

$$\langle \hat{\mathbf{e}}, \widehat{\tilde{e}_1^K}, \widehat{\tilde{f}_1^K}, \hat{\boldsymbol{\pi}} \rangle = \operatorname*{arg\,max}_{\mathbf{e}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi}} \sum_i \lambda_i \, \phi_i(\mathbf{e}, \mathbf{f}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi}) \tag{2.19}$$

Model Features While equations (2.16), (2.17) and (2.19) allow integrating an arbitrary set of translation features in the model, the following is a list of basic features often included in PBSMT systems.

- Conditional phrase translation probabilities: A score based on the logarithm of the conditional probability of independently translating each phrase, according to the heuristic scores of (2.18). Interestingly, two scores are employed, $\phi_{\text{PHR}}^{\mathbf{e}|\mathbf{f}}$ examining conditional translation of phrases from target to source and $\phi_{\text{PHR}}^{\mathbf{f}|\mathbf{e}}$ treating the opposite translation direction.

$$\phi_{\text{PHR}}^{\mathbf{e}|\mathbf{f}} = \log \prod_{k=1}^{K} p(\tilde{e}_k|\tilde{f}_k) \qquad \phi_{\text{PHR}}^{\mathbf{f}|\mathbf{e}} = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \tag{2.20}$$

- Lexical smoothing features: These features consider the quality of phrasal translations on the lexical level. They serve as a smoothing conditional phrase translation probability value, which is particularly helpful for phrase-pairs extracted only a few times from the training corpus and for which the phrase translation probability values above are set using sparse evidence. They are based on a model similar to IBM Model 1 of equation (2.7), employing lexical translation probabilities $w(e|f)$, $w(f|e)$ estimated under relative frequency from the word-aligned training parallel corpus, with the unaligned words treated as being aligned to an additional EMPTY token. The contribution of multiply aligned words is averaged and if a phrase-pair appears with multiple alignment patterns, the maximum alignment score is used. One feature per translation direction ($\phi_{\text{LEX}}^{\mathbf{e}|\mathbf{f}}$, $\phi_{\text{LEX}}^{\mathbf{f}|\mathbf{e}}$) is also employed for this feature category.

  The equations arriving at $\phi_{\text{LEX}}^{\mathbf{e}|\mathbf{f}}$ are shown below, while the values of $\phi_{\text{LEX}}^{\mathbf{f}|\mathbf{e}}$ are similarly computed based on $w(f|e)$ instead. For the phrase-pair of target phrase $\tilde{e}$ of length $\tilde{l}$ and source phrase $\tilde{f}$ of length $\tilde{m}$, with alignment points $\langle i, j \rangle \in \tilde{a}$ between them, we have:

$$w(e|f) = \frac{C(e,f)}{\sum_{e'} C(e',f)} \tag{2.21}$$

$$p_w(\tilde{e}|\tilde{f},\tilde{a}) = \prod_{i=1}^{\tilde{l}} \frac{1}{|\{j|\langle i,j\rangle \in \tilde{a}\}|} \sum_{\langle i,j\rangle\in\tilde{a}} w(e_i|f_j) \tag{2.22}$$

$$\widehat{p_w}(\tilde{e}|\tilde{f}) = \max_{\tilde{a}} p_w(\tilde{e}|\tilde{f},\tilde{a}) \tag{2.23}$$

$$\phi^{\mathbf{e}|\mathbf{f}}_{\text{LEX}} = \log \prod_i \widehat{p_w}(\tilde{e}|\tilde{f}) \tag{2.24}$$

- Phrase reordering feature: This feature ϕ_{RE} examines the reordering pattern of the phrasal translations. A choice employed in earlier PBSMT systems was based on distance-based scores. For example (Koehn et al., 2003) captured the movement of phrasal translations in the target sentence by providing a score proportional to the sum of the distances of each phrase's first word to the previous phrase's last word.

  A different approach was followed by (Tillman, 2004; Koehn et al., 2005). For each phrase-pair in the training corpus, a *monotone*, *swapping* or *discontinuous* reordering event was recorded based on the target phrase's relative position in regard to the previous source phrase's translation. A simple model built around relative frequency estimates from these heuristic counts is used to compute the reordering feature score. This lexicalised model allows learning for example that between French and English, *grande* usually translates monotonically, while *bleue* frequently swaps in relation to the noun before it.

- Word and phrase penalties: A feature counting how many target words are produced helps tune output length. An additional feature counting how many phrase-pairs were used in the derivation of a translation can be employed to prefer less (i.e. larger) or more (i.e. smaller) phrase-pairs.

- Language model: Finally, a language model feature ϕ_{LM} is one of the most crucial features of PBSMT models, examining output well-formedness. This is typically the logarithm of the probability $p(\mathbf{e})$ assigned to the target sentence by an already trained language model.

  The state-of-the-art Markovian language models employed for MT consider the target output across phrase-pair boundaries, often providing essential input to the model towards preferring overall well-formed target sentences. However, for the same reason the LM feature is also one of the most difficult to integrate in decoder implementations. A comprehensive presentation of LMs frequently used for MT can be found in (Chen and Goodman, 1998).

Feature Weights Training Log-linear models combine a multitude of feature functions together. Each of these features focuses on a different aspect of translation, while their typical value ranges often differ. A crucial part of training such a model is setting the feature weights λ, to optimally combine all the feature outputs together so that strong translations score higher. There is a growing literature of approaches and related learning objectives to perform this, including constrained entropy maximisation (Berger et al., 1996) and the Margin Infused Relaxed Algorithm (MIRA) (Crammer et al., 2006), that have been proven appealing also for SMT. Nevertheless, for the majority of phrase-based model applications including the relevant parts of this thesis, the feature weights are trained employing Minimum Error Rate Training (MERT) (Och, 2003).

Relying on optimising some information theoretical value such as test data likelihood or perplexity under the model often assumes a zero-one loss function. This ignores that apart from exact matches, there exists an ordering of alternative translations according to their appeal for humans. MT evaluation metrics such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and METEOR (Banerjee and Lavie, 2005) try to address this deficiency by providing a numeric assessment of MT output quality which aims to correlate with human judgement. MERT focuses on optimising model parameters with respect to the quality of the model's output as measured by an MT evaluation metric, most frequently BLEU.

Let us assume a *development* set of source sentences $\mathbf{f}_1^S$ and target reference translations $\mathbf{r}_1^S$. We further assume a set $\mathbf{C}_s = \{\mathbf{e}_{s,1}, \ldots, \mathbf{e}_{s,K}\}$ of K candidate translations for each $\mathbf{f}_s$ input sentence. Let the number of errors of a translation $\mathbf{e}$ in relation to reference $\mathbf{r}$ according to the metric in use be $E(\mathbf{r}, \mathbf{e})$ and assume the total errors over the development corpus is the sum of the errors of the individual sentences, i.e. $E(\mathbf{r}_1^S, \mathbf{e}_1^S) = \sum_{s=1}^S E(\mathbf{r}_s, \mathbf{e}_s)$. MERT is centred around optimising the feature weights λ_1^M by minimising the error of the best candidate translations $\hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M)$ between every set $\mathbf{C}_s$ according to the model.

$$\hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M) = \arg\max_{\mathbf{e} \in \mathbf{C}_s} \sum_{m=1}^{M} \lambda_m \phi_m(\mathbf{e}|\mathbf{f}_s) \tag{2.25}$$

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \sum_{s=1}^{S} E(\mathbf{r}_s, \hat{\mathbf{e}}(\mathbf{f}_s; \lambda_1^M)) \tag{2.26}$$

As the criterion of (2.26) is difficult to solve analytically, (Och, 2003) proposes an approximation of it and an optimisation algorithm operating on this approximation. This is used in practice as follows. The log-linear model is used to produce an N-best list of candidate translations for each development source sentence. The optimisation algorithm is run to arrive at the feature weights which promote (i.e. assign greater probability to) the best candidate translations. However, as the new model parameters might change not only the ordering but also the sentences present in the N-best list, a new N-best list is generated with the current best weights. These new candidate translations are added to the existing set produced during the previous steps and the optimisation algorithm is run on this expanded set of translation candidates. The process iterates until there are no novel candidate translations produced with the latest feature weights setting.

Impact of Phrase-Based models The introduction and adoption by the SMT community of the phrase-based and log-linear modelling approach had a profound impact on modern SMT, even though many of the implications did not become initially apparent. One of the most visible novelties was the generalisation of the word translation table to the new central data-structure of PBSMT systems: the phrase-table. While it seems this might have been initially conceived as merely increasing the minimal translation units set to include phrase-pairs in addition to word-pairs, it gradually became evident that this memorisation of training corpus fragments significantly changes the nature of these models and brings with it fundamental challenges in training and applying them. This crucial issue will be further discussed in Chapter 3.

PBSMT modelling also takes a distance from the noisy-channel, generative process modelling approach of the IBM models, opting instead for discriminative, feature based models. Probabilistic conditional models that examine candidate translations across *both* translation directions are combined together, as part of an array of different translation features. However, while these conditional translation probability models form the backbone of the log-linear PBSMT models, they are still estimated heuristically, disregarding the latent segmentation of sentence-pairs into phrase-pairs. This issue is touched upon by the Joint Translation Model discussed below.

Finally, training the PBSMT log-linear model parameters mostly abandons directly fitting the training or development data. Instead, most PBSMT implementations opt for translation metric (e.g. BLEU) score optimisation, as evidenced by the prevalence of the MERT method for tuning feature weights.

2.3.2 Joint Probability Model

While both the Alignment Template approach and its later development into Phrase-Based SMT directly model phrase-pairs, their training is largely based on an already word-aligned parallel corpus. Moreover, to estimate a phrasal translation model they rely on various heuristics. These range from establishing what constitutes a phrase-pair given a word-aligned sentence-pair to estimation based on normalising heuristic phrase-pair extraction counts.

Marcu and Wong (2002) propose a joint, purely phrase-based SMT model, which directly models the generation of sentence-pairs from phrase-pairs, without assuming a word-alignment variable. Instead, they generalise word-alignments in phrasal alignments, which also include phrases of length one. They then draw from the estimation literature related to word-alignment models (such as the IBM models) to estimate the model parameters by maximising the likelihood of the training set.

In more detail, the Joint Probability Translation Model (JPTM) generative process is based on drawing a bag of phrase-pairs $\langle \tilde{e}, \tilde{f} \rangle$ from a joint distribution with probability $p(\tilde{e}, \tilde{f})$. Subsequently, the phrases of both source and target are ordered according to a position-based distortion distribution $d(pos(\tilde{e}_1^K), pos(\tilde{f}_1^K))$, employing the word positions *pos* of the phrases for a particular phrase ordering. With C the bag of phrase-pairs $c_i = \langle \tilde{e}, \tilde{f} \rangle$, $L(\mathbf{e}, \mathbf{f})$ the set of all such bags of phrase-pairs from whose reordering and concatenation the sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$ can be formed and assuming that each phrase-pair is independently sampled from the joint distribution, we have:

$$p(\mathbf{e}, \mathbf{f}) = \sum_{C \in L(\mathbf{e},\mathbf{f})} \left\{ \prod_{c_i \in C} p(\tilde{e}, \tilde{f}) \right\} \times d(pos(\tilde{e}_1^K), pos(\tilde{f}_1^K)) \tag{2.27}$$

Distortion models Two models were derived from equation (2.27), employing different distortion distributions. *Model 1* assumes a uniform distortion distribution, effectively modelling jointly phrases under similar assumptions as those used by IBM Model 1 to model words conditionally. While Model 1 is shown to be able to induce reasonable phrasal alignments, it can hardly be used to translate with, given that it imposes no constraints on phrase positioning. For this reason, *Model 2* introduces a distortion distribution based on absolute word positions in a manner reminiscent of IBM model 2.

Parameter Estimation The JPTM follows the estimation principles established with the IBM models and estimates model parameters using Maximum Likelihood Estimation under the EM algorithm. However, this poses significant computational challenges. A corpus of length N with sentence pairs of length n each contains $O(Nn^4)$ phrase-pairs. For typical values of $N = 1\text{M}$, $n = 40$ this amounts to a number of phrase-pairs in the order of trillions. Moreover, each such sentence pair has $\sum_{k=1}^{n} k! S^2(n, k)$ different phrase-alignment patterns where $S(n, k)$ is the Stirling number of the second kind, amounting to a number in the order of 10^{83} for $n = 40$. Even though this number represents an overestimation as it includes pairs of non-contiguous phrases not covered by the JPTM, it serves as a gross indication of the computational challenges involved.

These computational challenges are addressed as follows. Firstly, only the probability for phrase-pairs that appear at least 5 times in the training corpus and are of at most length 6 is tracked, greatly reducing the size of the phrase-table. A formula employing the Stirling number estimate of phrase-pair segmentations discussed above is employed to arrive at an estimate of the expected counts of phrase-pairs given the corpus and an initial uniform joint distribution $p(\tilde{e}, \tilde{f})$.

The joint distribution is initialised using these expected counts, approximating a single step of the EM algorithm initialised with uniform $p(\tilde{e}, \tilde{f})$. These choices both restrict the number of phrase-pairs considered and provide a reasonable initial estimate for their probability.

In addition, as it is not feasible during the expectation step of the EM algorithm[2] to consider all phrasal segmentations and alignments, the most probable Viterbi alignment is found and fractional expected counts are computed by exploring neighbouring phrasal alignments. This approximation of the expectation step had already been used successfully in the training of the IBM Models 3 to 5. Later work such as (Cherry and Lin, 2007) tries to estimate the parameters of the JPTM by limiting the phrase-alignment search space, using the Inversion Transduction Grammar and the assumptions behind it as a modelling vehicle.

Under these constraints and approximations, the parameters for the joint phrase-pair distribution $p(\tilde{e}, \tilde{f})$ and the distortion distribution $d(pos(\tilde{e}_1^K), pos(\tilde{f}_1^K))$ are estimated. Both are employed to derive conditional distributions $p(\tilde{f}|\tilde{e})$ and $d(pos(\tilde{f}_1^K)|pos(\tilde{e}_1^K))$, which are used in a Noisy Channel decoder, which also uses a language model over the target sentences $\mathbf{e}$. Overall, in (Marcu and Wong, 2002) it is shown that for a corpus of 100K sentence-pairs the JPTM performs significantly better than translating with the word-based IBM Model 4.

Impact of the JPTM The Joint Probability Translation Model still provides inspiration for phrase-based translation research as a successful attempt to estimate model parameters under a clear learning objective such as Maximum Likelihood, in contrast to the heuristics employed by the PBSMT models discussed earlier in this chapter. It builds on the prior work on estimation for word-based models to propose solutions for the computational challenges involved and highlights the feasibility of better understood learning objectives for phrase-based SMT. This thesis proceeds along similar lines to also contribute in later chapters well-understood estimators for phrase-based models.

As with PBSMT models, an MLE estimator such as that approximated in (Marcu and Wong, 2002) can be shown to heavily overfit the training data, assigning non-zero probability only to full sentence-pairs. While the authors manage to arrive at usable estimates due to the particular constraints posed during estimation, this issue remains a crucial weakness of MLE estimators for phrase-based models, including the joint-probability model. It makes little sense to aim at replacing heuristic estimation with a better understood estimator when we still have to rely on dubious constraints to arrive at estimates which generalise well. In this work, we address this overfitting behaviour of MLE estimators for phrase-based models to formulate a robust training framework with no need of artificial constraints.

Furthermore, the work on the JPTM showed how a generative, joint-probability

[2] The steps of the EM algorithm are discussed later in this chapter, in section 2.6.

model over sentence-pairs can provide translation performance related to that attained using conditional models. However, the JPTM as implemented above did not prove in the long run to be competitive in terms of translation performance in comparison to the discriminative PBSMT models, as the log-linear formulation of the latter makes integrating diverse translation features easier. This established the understanding that conditional-probability models are superior in terms of performance to joint-probability models, a perception strengthened by the fact that Marcu and Wong (2002) as well converted their joint-probability estimates to conditional distributions prior to decoding. In later chapters of this thesis, we provide evidence that a joint-probability model can provide strong performance as the backbone of a log-linear model employing additional features.

While the JPTM modelled contiguous phrase-pairs, the authors note the possible extension to models using non-contiguous phrases. The following section explores how *synchronous* grammars, modelling the generation of strings across two languages, can be employed to translate with non-contiguous phrase-pairs.

2.4 Hierarchical SMT

At the same time as contiguous phrase-based SMT models dominated the state-of-the-art in the first half of the past decade, three influential desiderata on SMT research were established. The first and most straightforward, although by no means trivial, was translating with non-contiguous phrase-pairs. As we mention above, this was already proposed at a desired extension of the JPTM by the time of its publication. However, the theoretical and practical challenges involved in training and decoding with such models delayed their introduction. The second issue was employing a syntactic approach for MT. Inspired by the advances in monolingual syntactic parsing, this line of research aimed at applying grammatical formalisms on the bilingual string-pairs involved in MT. The final desideratum concerned taking advantage of linguistic syntactic annotations in MT modelling. These can be used for example to constrain existing models that are not otherwise linguistically motivated or as integral parts of syntactic MT approaches. The application of Synchronous Context Free Grammars on MT has interestingly provided the foundations to pursue all three goals above.

2.4.1 Synchronous CFG Grammars

Synchronous grammars in the general sense are formal grammars whose language is a set of string-pairs. Monolingual syntactic approaches have long been extended to generate, recognise and process bilingual strings. These include the syntax-directed translation (Aho and Ullman, 1969) and syntax-directed transduction (Lewis and Stearns, 1968) approaches , as well as the more recent Multiple CFGs (Seki et al., 1991) and Multitext Grammars (Melamed, 2003), all stemming from monolingual Context Free Grammars (CFGs). CFGs are not the only formalism to be extended to parallel strings, as we also find Synchronous Tree-Adjoining Grammars (TAGs) (Shieber and Schabes, 1990) extending monolingual TAGs for bilingual parsing, as well as Synchronous Tree-Substitution Grammars (Poutsma, 2000; Eisner, 2003) generating string-pairs by combining pairs of linked syntactic subtrees.

In this work we will confine ourselves to what is currently covered by the term Synchronous Context Free Grammars (SCFGs). While these grammars trace their foundations back to (Lewis and Stearns, 1968; Aho and Ullman, 1969), they have more recently been established as syntactic formalisms for MT after the introduction of a computationally and linguistically appealing subset of SCFGs, the Inversion Transduction Grammars (Wu, 1997) and its phrase-based extension (Chiang, 2005a).

SCFGs provide an appealing formalism to describe the translation process, which explains the generation of parallel strings recursively and allows capturing long-range reordering phenomena. Formally, an SCFG $\mathbf{G}$ is defined as the tuple $\langle N, E, F, R, S \rangle$, where N is the finite set of non-terminals with $S \in N$ the start symbol, F and E are finite sets of words for the source and target language and R is a finite set of rewrite rules. Every rule expands a left-hand side non-terminal to a right-hand side pair of strings, a source language string over the vocabulary $F \cup N$ and a target language string over $E \cup N$. The number of non-terminals in the two strings is equal and the rule is complemented with a mapping between them.

String pairs in the language of the SCFG are those with a valid derivation, consisting of a sequence of rule applications, starting from S and recursively expanding the linked non-terminals at the right-hand side of rules. *Probabilistic* SCFGs augment every rule in R with a probability, under the constraint that probabilities of rules with the same left-hand side sum up to one. The probability of each derived string pair is then the product of the probabilities of rules used in the derivation. Unless otherwise stated, for the rest of this work when we refer to SCFGs we will be pointing to their stochastic extension.

The recursive nature of languages can be extended to the relation between them that a translation process establishes. SCFGs can crucially express both the recursive nature of translation and the reordering patterns that emerge. An example of a small grammar capturing Subject-Verb-Object (SVO) to Subject-Object-Verb (SOV) reordering and recursive construction of questions between English and Japanese can be seen in Figure 2.6.

Binary SCFGs The *rank* of an SCFG is defined as the maximum number of non-terminals in a grammar rule's right-hand side. The grammar in Figure 2.6 would be of rank 3. Contrary to monolingual Context Free Grammars, there does not always exist a conversion of an SCFG of a higher rank to one of a lower rank

$$\begin{aligned}
S &\rightarrow X_{\boxed{1}} \,/\, X_{\boxed{1}} \\
S &\rightarrow \text{Do } X_{\boxed{1}} \text{ ? } / \; X_{\boxed{1}} \text{ ka ?} \\
X &\rightarrow NP_{\boxed{1}}\; VB_{\boxed{2}}\; NP_{\boxed{3}} \,/\, NP_{\boxed{1}}\; NP_{\boxed{3}}\; VB_{\boxed{2}} \\
NP &\rightarrow \text{I / watashi ga} \\
VB &\rightarrow \text{open / akemasu} \\
NP &\rightarrow \text{the book / hako o}
\end{aligned}$$

Figure 2.6: An SCFG rule set for SVO to SOV reordering and question construction from English to (romanised) Japanese

with the same language of string pairs. For example, even though all SCFGs of rank 3 can be converted to an equivalent one (i.e. defining the same language of string-pairs) of rank 2, the same does not apply for some SCFGs with rank 4 and above. We can convert the grammar of Figure 2.6 to one of rank 2, by replacing the third rule with the following 2 rules:

$$\begin{aligned}
X &\rightarrow NP_{\boxed{1}}\; Z_{\boxed{2}} \,/\, NP_{\boxed{1}}\; Z_{\boxed{2}} \\
Z &\rightarrow VB_{\boxed{1}}\; NP_{\boxed{2}} \,/\, NP_{\boxed{2}}\; VB_{\boxed{1}}
\end{aligned}$$

However, the following rule involving 4 non-terminals on its right-hand side cannot be binarised.

$$X \rightarrow X_{\boxed{1}}\; X_{\boxed{2}}\; X_{\boxed{3}}\; X_{\boxed{4}} / \; X_{\boxed{3}}\; X_{\boxed{1}}\; X_{\boxed{4}}\; X_{\boxed{2}}$$

The computational complexity and memory demands of algorithms parsing or decoding with SCFGs increases with the rank of the grammar. For this, most machine translation applications focus on SCFGs of rank two, binary SCFGs (bSCFGs) (Wu, 1997), as well as SCFGs which are *binarisable*. These are Synchronous CFGs of any rank for which a conversion to an equivalent binary SCFG exists. Fortunately, binarisable SCFGs seem to be able to cover most of the reordering patterns encountered in natural language pairs (Wu, 1997; Huang et al., 2009). This feature, coupled with the relative computational efficiency of algorithms employing bSCFGs makes the latter an appealing formalism to describe translation phenomena.

Inversion Transduction Grammars Binary SCFGs were brought to prominence by the introduction of the Inversion Transduction Grammars (ITGs) of (Wu, 1997). ITGs are a subset of SCFGs as we defined them above, where the right-hand side of rules of arbitrary length either keeps its order between the two strings or this order is inverted. Wu (1997) shows that all of these grammars can be converted to a normal form, involving either two non-terminals B, C or a word-pair $\langle e, f\rangle$. Rules leading to two non-terminals on the right-hand side can either map the two to translate monotonically across the two languages keeping the order intact, or swap the two non-terminals inverting the strings covered by them. For these grammars, we can switch to a simpler notation than that of Figure 2.6, denoting with [ ] monotone and with ⟨ ⟩ swapping reordering. Using this notation, grammars in the ITG normal form contain only rules of the forms[3]:

$$\mathrm{A} \rightarrow [\mathrm{B}\ \ \mathrm{C}] \qquad \mathrm{A} \rightarrow \langle \mathrm{B}\ \ \mathrm{C}\rangle \qquad \mathrm{A} \rightarrow \mathrm{e} \,/\, \mathrm{f} \tag{2.28}$$

SCFG Algorithms While SCFGs are closely related to the monolingual Context Free Grammar formalism, performing tasks such as parsing with an arbitrary SCFG can be notoriously hard, with (Satta and Peserico, 2005) showing that both parsing and decoding are NP-hard. Nevertheless, while the results above apply in the general case, binary SCFGs can still be processed in polynomial time, making them an ideal candidate for practical applications. The algorithms involved then for the most usual tasks are an extension of algorithms employed for monolingual CFGs.

Parsing Parsing string-pairs using bSCFGs can be performed in polynomial time employing a modified version of the CYK algorithm (Cocke, 1969; Younger, 1967; Kasami, 1965). Running time is then $O(n^6|\mathbf{G}|)$, polynomial in both the length of each string n and the size of the grammar $|\mathbf{G}|$. However, the higher exponent makes parsing with SCFGs with the computational resources available nowadays significantly more challenging than monolingual parsing, with applications frequently having to resort to constraints or approximations.

Decoding Finding all target strings $\mathbf{e}$ for a given source $\mathbf{f}$ that belong in the language $\{\langle \mathbf{e}, \mathbf{f}\rangle\}$ of $\mathbf{G}$, or $\mathbf{e}$ belonging to the most probable $\langle \mathbf{e}, \mathbf{f}\rangle$ according to a stochastic SCFG has interestingly a lower complexity $O(n^3)$ in respect to the sentence size. Decoding can be performed by modified versions of Earley-style parsers (Earley, 1970), such as synchronous adaptations of the CYK+ algorithm (Chappelier and Rajman, 1998). However, as we will discuss later in this chapter, state-of-the-art applications of bSCFGs for translation include the usage of a language model over the target language

[3] We skip productions involving an empty token in one of the two strings.

output during decoding, which complicates computations and demands non-trivial hypothesis search and pruning strategies.

Expectation-Maximization Estimating the parameters of a synchronous CFG given a corpus of parallel sentences can be performed using the EM algorithm. The Expectation step of the algorithm demands the computation of expected usage counts for all rules of the bSCFG given the current estimate. This can be performed with a modified version of the Inside-Outside algorithm (Baker, 1979; Lari and Young, 1990), running in the same complexity as SCFG parsing, namely $O(n^6|\mathbf{G}|)$.

2.4.2 The Hiero Translation System

SCFGs were initially introduced for machine translation as a stochastic *word-based* translation process in the form of the Inversion-Transduction Grammar. Simultaneously, progress in phrase-based translation showcased how translating with phrases significantly improves translation quality in comparison with word-based models. The advances in syntactic modelling of translation on the one hand and those in phrase-based translation and the related methods such as log-linear modelling for MT and MERT estimation on the other, converge with the introduction of Hierarchical Phrase-based SMT (HPBSMT) and the Hiero translation system (Chiang, 2005a; Chiang, 2007).

A key practical consideration in extending word-based ITG to the SCFG employed by Chiang is that SCFGs including phrases in the right-hand side of rules can make use of similar efficient decoding algorithms as ITGs, as long as they are binary SCFGs employing up to two non-terminals on rule expansions. Chiang takes advantage of this feature to propose an SMT system capable of employing *non-contiguous* phrase-pairs.

Synchronous Grammar Hiero is based on a synchronous grammar with a single, general-use non-terminal X covering all learnt phrase-pairs. Beginning from the start symbol S, an initial phrase-span structure is constructed monotonically using a simple 'glue grammar', which in practice constitutes the only rules allowed to be applied to spans larger than a predefined cut-off length[4].

$$\begin{aligned} S \rightarrow & S_{\boxed{1}}\, X_{\boxed{2}} \,/\, S_{\boxed{1}}\, X_{\boxed{2}} \\ S \rightarrow & X_{\boxed{1}} \,/\, X_{\boxed{1}} \end{aligned}$$

The true power of the system lies in expanding these initial phrase-spans with a set of recursive translation rules expanding towards non-contiguous phrase-pairs, such as those of Figure 2.7. Similarly to ITG grammars, the gaps in

[4] A usual setting of this is 10.

$$\begin{array}{ll}
X \to & \text{do not } X_{\boxed{1}} \text{ / ne } X_{\boxed{1}} \text{ pas} \\
X \to & \text{financial } X_{\boxed{1}} \text{ / } X_{\boxed{1}} \text{ économiques} \\
X \to & \text{this } X_{\boxed{1}} X_{\boxed{2}} \text{ / cette } X_{\boxed{1}} \text{ de } X_{\boxed{2}} \\
X \to & X_{\boxed{1}} \text{ ' s common } X_{\boxed{2}} \text{ policy /} \\
 & \text{politique } X_{\boxed{2}} \text{ commune de } X_{\boxed{1}}
\end{array}$$

Figure 2.7: Hiero SCFG rules for English and French

the two sides of these phrase-pairs, as expressed by the X non-terminals in the synchronous productions, are mapped to each other so that the linked spans translate either monotonically or swap. However, long-range reordering is handled by the glue rules, limiting these reorderings to translating monotonically. This, together with the use of a single non-terminal X apart from the start symbol highlights that employing an SCFG grammar for the Hiero system is more of a vehicle to model and decode with non-contiguous phrase-pairs, than an attempt to learn and exploit the hierarchical structure of parallel data.

Nevertheless, non-contiguous phrase-pairs with binary reordering greatly increase the descriptive power of the phrase-table. In the first place, they allow memorising phrase patterns whose words might lie far apart in the training corpus, without the need to couple them to the particular in-between context that they appear with. Secondly, they provide the means to learn context-driven reordering patterns, some examples of which can be seen in Figure 2.7. This reduces the need for a separate reordering model such as those employed for PBSMT, with the original Hiero system relying solely on the non-contiguous phrase-pairs to reorder during decoding.

Even more crucially, non-contiguous phrase-pairs offer a generalisation of the phrase-table that reduces the effect of data sparsity. Taking the famous 'ne . . . pas' negation construction in French as an example, PBSMT models can memorise and reuse during decoding only translations of instances of it appearing in the training data with particular verbs, such as 'ne veux pas' or 'ne peux pas'. Phrase-tables that employ non-contiguous phrase-pairs such as Hiero's are able to successfully generalise these instances to 'ne X pas', greatly expanding the generalisation power of the phrase-table.

This power does not come without its challenges. As the space of possible contiguous phrase-pairs that the rules of the synchronous grammar can lead to increases, so does the need to avoid generalising towards erroneous phrase translations. For example we would like to somehow consider as more probable expanding the non-terminal X of 'ne X pas' towards a verb than a noun. The grammar of the HPBSMT does not consider this and instead relies on additional features and most prominently on the language model feature to disambiguate between stronger and weaker expansions.

Finally, restricting the right hand side of the SCFG rules to two non-terminals also limits the descriptive power of bSCFG grammars as they can only cover binary reordering patterns. However, given the evidence that most of the actual reordering taking place in natural language pairs does follow these constraints as discussed in (Huang et al., 2009), it might well be that this shortcoming is actually a strength of bSCFGs. Namely, that they greatly decrease the size of the search space in a manner that not only improves computational efficiency, but also correlates with the transformations found in natural language translation, avoiding search errors and preventing distributing probability to translations that have little to do with natural language correspondences.

Translation Model and Training Establishing the form of the translation model and training follows the same pattern as PBSMT models. Firstly, the grammar is built by complementing the fixed glue-grammar rules with non-contiguous phrase-pair rules like those of Figure 2.7, which are extracted from the training corpus. Non-contiguous phrase-pair extraction from a word-aligned parallel corpus follows the same heuristics in regard to what constitutes a phrase-pair as contiguous phrase-pair extraction in alignment template and PBSMT systems. In addition however to pairs of contiguous phrases, Hiero extracts also phrase-pairs with 'gaps', from those which include internal spans that are themselves considered phrase-pairs. This process continues recursively, extracting rules that include up to two 'gaps' on the right hand side. The phrasal alignment pattern between the internal phrase-pairs is preserved by the mapping of the X non-terminals in the rule's right-hand side. All grammar rules created in this process have again the single non-terminal X as their left-hand side.

The grammar extracted is further augmented to become a *weighted* bSCFG by assigning weights to the extracted rules using similar heuristics based on extraction counts as these used in PBSMT, for example by normalising extraction counts per source or target right-hand side. The weighted bSCFG provides a score for a derivation as the product of the weights of the rules taking part in it. This score is used as a feature to form part of a log-linear, feature-based translation model. For every derivation D with $\mathbf{f}$ as the source string, $p(D)$ is proportional to a log-linear function of a language model feature and a number of additional features examining each rule application r independently from the rest of the derivation.

$$p(D) \propto \phi_{\text{LM}}^{\lambda_{\text{LM}}} \times \prod_{r \in D} \prod_{i \neq \text{LM}} \phi_i^{\lambda_i}(r) \tag{2.29}$$

The feature weights λ are trained by Minimum Error Rate Training, in the same way as the similar log-linear models employed for phrase-based translation.

Equation (2.29) can be misleading, given that the bSCFG could be considered as merely providing the scores for a handful of features among a multitude of these. On the contrary, the bSCFG functions as the *backbone* of the log-linear model, as the space of translations $\mathbf{e}$ considered for the input sentence $\mathbf{f}$ are exactly those which take part in string pairs $\langle \mathbf{e}, \mathbf{f} \rangle$ in the language of the bSCFG. In addition, the importance of the weighted bSCFG scores is central, given that the rest of the features are either monolingual (as the LM feature) or are smoothing features similar to those used in PBSMT models.

Impact of the Hiero system At first sight, Hiero introduces a hierarchical phrase based translation system capable of employing non-contiguous phrase-pairs. This allows to generalise the PBSMT phrase-table and address training data sparsity by enabling the memorisation and reuse of non-contiguous phrase patterns, disentangled from their particular application in the parallel corpus.

More importantly though, Hiero showcased a *syntactic* approach for SMT that was both computationally scalable and offered state-of-the-art performance. The fact that Hiero offered a simple instantiation of such an approach stimulated further research in a number of open questions.

- As is the case with PBSMT's phrase translation probabilities, the weights assigned to bSCFG rules play a central role in discerning strong from weak translations. How can we estimate these crucial model parameters with a more meaningful learning objective than heuristic estimation ?

- The hierarchical and compositional nature of syntactic SMT can lead to weak generalisations if the structural part of the synchronous grammars does not focus on productions which make linguistic sense. How can we learn a richer syntactic MT structure, possibly taking advantage of linguistic syntax, which better models the hierarchical nature of natural language translation ?

These questions will be the central themes of chapters 5 and 6 of this thesis.

2.4.3 Linguistically Augmented Hierarchical Translation

While the Hiero system exhibited a syntactic approach to MT it stopped short of employing *linguistic* syntax. The latter provides linguistic annotations to sentences which are generally understood to correlate with translation transformations up to a certain extent. Even more, these annotations are recursive, a feature which promotes them as prominent candidates to be integrated in a hierarchical translation system.

We may categorise MT systems employing linguistic syntax in two categories. The first brings together systems where linguistic syntax is the primary vehicle to describe the translation process. An example could be models which explain translation through transformations on the source parse tree or which build the target sentence through combining target parse fragments, such as (Yamada and Knight, 2001; Galley et al., 2006). The second category includes systems which, recognising their informative value for translation, take advantage of linguistic annotations while not strictly adhering to them to explain the translation process. For example, in (Venugopal et al., 2009; Chiang et al., 2009) the linguistic plausibility of translations is assessed through additional features in an otherwise non-contiguous phrase-based system. Below we focus on SAMT, a linguistically enriched extension of the Hiero system.

Syntax Augmented MT The Syntax Augmented MT (SAMT) system (Zollmann and Venugopal, 2006) can be classified in the latter category. It extends HPBSMT by extracting linguistically augmented hierarchical translation rules. For this it utilises constituency parse trees of the *target* sentences. Whenever a rule like those in Figure 2.7 is extracted, if the spans substituted by the generic non-terminal X are also covered by a constituent non-terminal NT in the target sentence parse, additional rules substituting X with NT are extracted. This leads to SCFG rules enriched with the use of a linguistically augmented non-terminal set, such as those in Figure 2.8. The rest of the SAMT model details mostly follow those of the Hiero system. A log-linear model employs features based on heuristic extraction counts and feature weights are optimised by MERT.

The rules of Figure 2.8 delineate how linguistic annotations are used towards an SCFG grammar which is, depending on the assigned rule scores and feature weights, possibly more selective in the recursive expansion of non-contiguous phrase-pairs. This is performed without completely committing to the linguistic structure itself. This is further exemplified by an additional set of rules employed by SAMT, which are augmented by SCFG non-terminals crossing linguistic constituent brackets. These may substitute X in the extracted rules when the underlying phrase-pair is a concatenation $NT1 + NT2$ of two or more constituents. They can also constitute a partial syntactic category $NT1$ missing a non-terminal $NT2$ on its right $NT1/NT2$ or its left $NT1\backslash NT2$, as in Categorial Grammar (Bar-Hillel, 1953). Examples of rules utilising these non-terminal categories appear in Figure 2.9

Overall, SAMT highlights a flexible approach on how to utilise linguistic syntax of the target language for MT. Decoding with a SAMT model results in the construction of target syntactic analyses which are able to take advantage of SCFG rules that are more linguistically aware than Hiero, hopefully producing translations which are more grammatically sound.

$$\begin{aligned}
X &\rightarrow && \text{do not } X_{\boxed{1}} \,/\, \text{ne } X_{\boxed{1}} \text{ pas} \\
X &\rightarrow && \text{do not } VB_{\boxed{1}} \,/\, \text{ne } VB_{\boxed{1}} \text{ pas} \\
VP &\rightarrow && \text{do not } VB_{\boxed{1}} \,/\, \text{ne } VB_{\boxed{1}} \text{ pas} \\
X &\rightarrow && \text{financial } X_{\boxed{1}} \,/\, X_{\boxed{1}} \text{ économiques} \\
NP &\rightarrow && \text{financial } NN_{\boxed{1}} \,/\, NN_{\boxed{1}} \text{ économiques} \\
X &\rightarrow && X_{\boxed{1}} \text{ ' s common } X_{\boxed{2}} \text{ policy /} \\
& && \text{politique } X_{\boxed{2}} \text{ commune de } X_{\boxed{1}} \\
X &\rightarrow && X_{\boxed{1}} \text{ ' s common } JJ_{\boxed{2}} \text{ policy /} \\
& && \text{politique } JJ_{\boxed{2}} \text{ commune de } X_{\boxed{1}} \\
NP &\rightarrow && X_{\boxed{1}} \text{ ' s common } JJ_{\boxed{2}} \text{ policy /} \\
& && \text{politique } JJ_{\boxed{2}} \text{ commune de } X_{\boxed{1}}
\end{aligned}$$

Figure 2.8: SAMT SCFG rules for English and French, extending Hiero's X-rules

$$\begin{aligned}
DT + NN &\rightarrow && DT_{\boxed{1}} \; NN_{\boxed{1}} / \; DT_{\boxed{1}} \; NN_{\boxed{1}} \\
NP\backslash DT &\rightarrow && \text{financial } NN_{\boxed{1}} \,/\, NN_{\boxed{1}} \text{ économiques} \\
S/VP &\rightarrow && \text{financial } NN_{\boxed{1}} \,/\, NN_{\boxed{1}} \text{ économiques} \\
VP\backslash VB &\rightarrow && DT + NN_{\boxed{1}} \text{ ' s common } X_{\boxed{2}} \text{ policy /} \\
& && \text{politique } X_{\boxed{2}} \text{ commune de } DT + NN_{\boxed{1}}
\end{aligned}$$

Figure 2.9: SAMT rules with compound non-terminals

2.5 Limits of Estimation Heuristics

SAMT also serves to exemplify the limits of heuristic rule scores based on rule extraction heuristics. Their usage was already debatable when counting contiguous or non-contiguous phrase-pairs in PBSMT and its hierarchical extension, given that this ignores the latent segmentation variable splitting sentence-pairs down to phrase-pairs. We might argue that this issue is somehow mitigated by the fact that at least we are counting events (phrase-pairs) in the observed part of the data (sentence-pairs), even though we skip explaining how we arrived there (segmentation).

In approaches like SAMT, we, somehow silently but still crucially, move forward to heuristically extract counts from the latent structure itself: the Synchronous CFG parse of the sentence-pair. This parse, for the linguistically augmented rules of SAMT, goes a long way past a simple segmentation in non-contiguous phrase pairs. While extracting event surface counts without completely disambiguating the latent structure underlying them might already feel uncomfortable, relying on artificial counts over the unobserved variables can completely undermine our confidence in the scores derived from them.

This issue is shared by all MT approaches employing a richer syntactic structure to explain the translation process (generative models) or discriminate between strong and weak translations (discriminative models). As the latent structures involved in these models become more complex, the risks from heuristically estimating the parameters of these structures increase. Crucially, this also increases the possible gains from learning these latent structures from parallel data.

2.6 Expectation-Maximization Algorithm

A popular and widely successful method to estimate the parameters of generative models using training data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is Maximum Likelihood Estimation. Under MLE, we seek to find the parameter set $\hat{\theta}$ for a stochastic model $p(X = \mathbf{x}; \theta)$ which maximises the likelihood of observing the training set $\mathcal{X}$. Assuming all samples $\mathbf{x}$ are independent and identically distributed (i.i.d), we have:

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\mathcal{X}; \theta) = \arg\max_{\theta} \prod_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}; \theta) \tag{2.30}$$

For a number of modelling problems, we might be fortuitous enough to possess training data which include the outcomes of all the random variables the model assumes for every data point, i.e. we can infer the model from *complete* data. For example, for a straightforward stochastic model over a loaded dice, complete data can refer to a number of rolling outcomes for it. If we interpret the model as a generative process, we might say that we then have, for every training data point, access to all the generative steps involved in their emission. Training a PCFG from a parsed sentence corpus (a treebank) falls under this case.

In these cases, MLE boils down to Relative Frequency Estimation (RFE), i.e. assigning to the model parameters the relative frequencies of the model's random variable outcomes in the observed data. While a straightforward RFE estimate is generally speaking readily computable, arriving at estimates which generalise well, i.e. predict unseen data accurately, is highly non-trivial. The additional estimation efforts are then mostly directed to *smoothing*: accounting for data sparsity, the fact that our training data provides but a limited glimpse in the distribution of outcomes of random variable X.

In other cases however, we might assume a model with *latent variables*, i.e. involving random variables whose outcome is not observed in the training data. One motivation towards formulating such a model could be aiming to uncover hidden patterns in the data, such as the word-alignments between sentence pairs of Figure 2.3. An additional objective might plainly be to better model the unknown data distribution, hoping that the assumptions behind our model can aid to better describe the data. An example might be using a mixture model to fit data, when a single standard distribution such as the Gaussian is not considered to be enough to accurately describe the underlying distribution. In these cases we have to estimate model parameters using what we then consider as *incomplete* data with *missing* values. The missing information in two aforementioned examples would be respectively the word-alignments for the parallel sentences and the indication of each data point's origin among the mixture's distributions.

Machine Translation is a prominent ML field where estimating model parameters from incomplete data is a central issue. In most experimental settings training data are merely sentence-aligned, with no further information on how, starting from the source sentence, we arrived at the target language output. The only model for which these are considered complete data is one which tracks the translation of whole sentences as a single unit, which has very limited applicability given the sparsity of the training data. Any meaningful generative SMT model which aspires to generalise well is thus bound to employ latent variables. We have already considered such cases in this chapter, such as the word-alignments for the IBM models, phrase segmentation for PBSMT models and the hierarchical translation structure for SCFG-based models.

In this section we will examine the challenges of estimating model parameters with MLE using incomplete data and how the Expectation-Maximization algorithm can address these, focusing on discrete distributions.

MLE with Incomplete Data Let us consider a model, parametrised by vector θ, over random variable $Z = \langle X, Y \rangle$, whose values $\mathbf{z}$ are tuples consisting of values $\mathbf{x}$ of the *observed* data variable X and values $\mathbf{y}$ of *missing* data variable Y. We use these names for X and Y, because the incomplete training data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

that we possess present only the values for X, while information over the value of Y for each data point is missing.

Given this setting, we have to rewrite the MLE criterion of (2.30) so that the unobserved variable is marginalised. For this we will need a function $Z(\mathbf{x})$ which maps every observed data point $\mathbf{x}$ to the set of all possible complete data from which it could descend from.

$$Z(\mathbf{x}) = \{\mathbf{z}_1 = \langle \mathbf{x}, \mathbf{y}_1 \rangle, \mathbf{z}_2 = \langle \mathbf{x}, \mathbf{y}_2 \rangle, \dots\} \tag{2.31}$$

The Maximum-Likelihood Estimate of a model over complete data can then be computed from incomplete data as follows.

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} \prod_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}; \theta) \\ &= \arg\max_{\theta} \prod_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle \in Z(\mathbf{x})} p(\mathbf{x}, \mathbf{y}; \theta) \end{aligned} \tag{2.32}$$

The problem is that, in most non-trivial cases, the optimisation above involving a product of sums cannot be solved analytically. We may however still iteratively arrive at increasingly better fitting parameter settings, by means of the EM algorithm.

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977)[5] provides a method to iteratively arrive at a *local* optimum of the likelihood function of (2.32), many times in a computationally appealing way. It stands out from superficially similar looking iterative algorithms, as it provides theoretical guarantees over its operation and its output.

Initialisation We begin by initialising the model's parameter set by an initial setting $\hat{\theta}_0$. Parameter initialisation can be crucial as EM can only climb towards a local optimum of the data likelihood function. For complex likelihood functions with more than one local maximum, the initialisation point determines towards which of these we will converge. As a result, a weak initialisation can result in a globally suboptimal estimate.

However, as in most cases we are not aware of the likelihood function's exact form and there is usually no clear way to judge the quality of an initialisation point, popular initialisation choices are uniform distributions or a randomly set parameter set. If practically possible, it makes also sense to run the EM algorithm multiple times, each one starting from a different initialisation point (random restarts). Hopefully, each will climb towards a different local optimum and we can select the estimate which corresponds to the best local optimum reached.

[5] (Prescher, 2004) and (Bilmes, 1997) provide interesting tutorials of the EM algorithm.

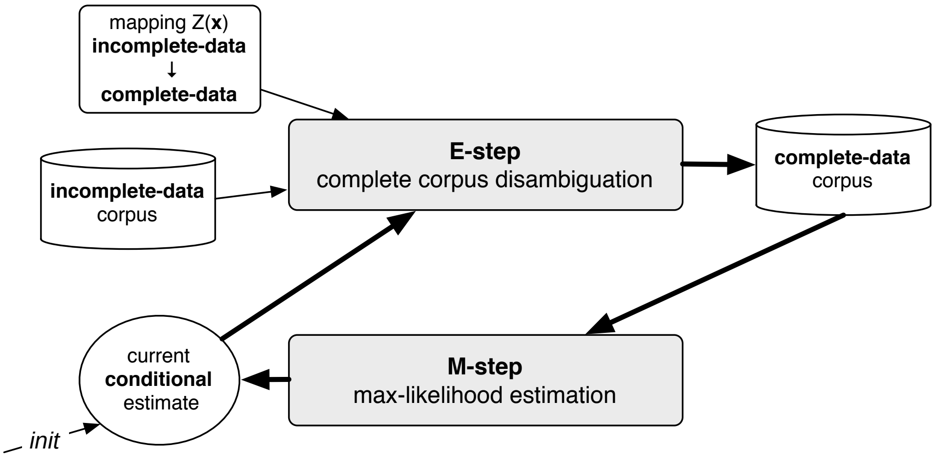


Figure 2.10: The two steps of the EM algorithm, adapted from (Prescher, 2004).

Iterative Procedure The initialisation point $\hat{\theta}_0$ together with the incomplete data training corpus $\mathcal{X}$ and function $Z(\mathbf{x})$ form the input of the EM algorithm. The basic operating principle behind EM can be described along the following lines.

If we apply function $Z(\mathbf{x})$ to every $\mathbf{x} \in \mathcal{X}$, we can build from the incomplete-data corpus $\mathcal{X}$ a *complete*-data corpus $\mathcal{Z}(\mathcal{X})$, extending each $\mathbf{x}$ to all its possible complete data expansions. For example, if the incomplete corpus consists of aligned sentence-pairs, we can expand each unaligned sentence-pair to multiple aligned ones, considering all possible alignments between the words of the two strings. The problem is that we do not know how to distribute the unit count of each observed data point among all its complete-data expansions; e.g. we do not know what the count of a particular alignment pattern would be given that it's sentence pair has been observed once. If we had that information and could thus disambiguate corpus $Z(\mathcal{X})$, MLE would boil down to relative frequency estimation.

The EM algorithm breaks out of this impasse following a two steps procedure. First, it uses in each iteration r the previous parameter estimate $\hat{\theta}_{r-1}$ (starting with $\hat{\theta}_0$) to disambiguate the complete-data corpus. This is performed by computing the expected counts of each complete-data expansion $\mathbf{z} \in Z(\mathbf{x})$ with respect to $\hat{\theta}_{r-1}$ (*E-step*). In the word-alignment example, we would compute for every sentence the expected counts of each possible word-alignment, as if it was produced from a model parametrised by $\hat{\theta}_{r-1}$. Subsequently, putting then $\hat{\theta}_{r-1}$ aside, EM moves on to compute a new estimate $\hat{\theta}_i$ by MLE on this disambiguated complete corpus (*M-step*). As this concerns maximising the likelihood of a corpus where for every data point all values of the model's variables are observed, this optimisation is in most cases feasible, and many times is equivalent to RFE.

Remarkably, this new estimate is guaranteed to better fit the training data.

The process then continues iteratively, each time using the previous estimate to arrive at a better one until convergence, as illustrated in Figure 2.10.

Let us now more thoroughly describe the two steps that each EM iteration consists of. We switch to the equivalent optimisation criterion of *log*-likelihood maximisation which is sometimes easier to formulate and solve. Also, to simplify the exposition, we describe the two steps for a single data point $\mathbf{x}$. The relevant equations easily extend to the full training corpus by summing[6] through all the i.i.d sampled data points of the corpus $\mathcal{X}$.

Expectation Step In the Expectation step (E-step), we formulate the *expected* log-likelihood $Q(\theta|\hat{\theta}_{r-1})$ of the complete-data $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$, given the observed incomplete-data point $\mathbf{x}$ and the parameter estimate from the previous iteration $\hat{\theta}_{r-1}$.

$$Q(\theta|\hat{\theta}_{r-1}) = E\left[\log p(\mathbf{z}|\theta)|\mathbf{x}, \hat{\theta}_{r-1}\right] = \sum_{\langle \mathbf{x},\mathbf{y}\rangle \in Z(\mathbf{x})} \log \{p(\mathbf{x},\mathbf{y}|\theta)\} \; p(\mathbf{y}|\mathbf{x},\hat{\theta}_{r-1}) \quad (2.33)$$

In the somewhat abstract equation above, it is crucial to notice that $p(\mathbf{y}|\mathbf{x}, \hat{\theta}_{r-1})$, the expected counts of the complete-data expansions of $\mathbf{x}$ given $\hat{\theta}_{r-1}$, can be readily computed and will function as a constant in the M-step that follows. Substituting it with $q(\mathbf{x}, \mathbf{y}|\hat{\theta}_{r-1})$ to denote this, we have:

$$q(\mathbf{x},\mathbf{y}|\hat{\theta}_{r-1}) = \frac{p(\mathbf{y}|\mathbf{x},\hat{\theta}_{r-1})}{\sum_{\langle \mathbf{x},\mathbf{y}'\rangle \in Z(\mathbf{x})} p(\mathbf{y}'|\mathbf{x},\hat{\theta}_{r-1})} \quad (2.34)$$

In practice, E-step implementations involve computing these expected counts. While it involves going through all possible complete-data expansions of $\mathbf{x}$ which can be exponential in number, for many practical applications dynamic programming algorithms allow them to be efficiently computed.

Maximization Step With the $q(\mathbf{x}, \mathbf{y}|\hat{\theta}_{r-1})$ counts already computed in the E-step, the Maximization step (M-step) of the EM algorithm involves maximising $Q(\theta|\hat{\theta}_{r-1})$ with respect to θ to retrieve the next parameter estimate $\hat{\theta}_r$.

$$\hat{\theta}_r = \arg\max_{\theta} Q(\theta|\hat{\theta}_{r-1}) = \arg\max_{\theta} \sum_{\langle \mathbf{x},\mathbf{y}\rangle \in Z(\mathbf{x})} \log \{p(\mathbf{x},\mathbf{y}|\theta)\} \; q(\mathbf{x},\mathbf{y}|\hat{\theta}_{r-1}) \quad (2.35)$$

Equation (2.35) involves optimising the model parameters from what is now, with the help of the counts computed during the E-step, a complete-data corpus. This is usually much easier than the original incomplete-data optimisation problem of (2.32) and many times translates to the usually easy to implement and compute Relative-Frequency Estimation.

[6] We are employing log-likelihood optimisation.

Theoretical Guarantees The EM algorithm's appeal is strengthened, by the fact that it is coupled with theoretical guarantees concerning its operation and output. Dempster et al. (1977) show[7] that the iterations of the EM algorithm provide:

Guarantee to Non-Decrease Likelihood After every iteration, the new estimate raises or leaves equal the likelihood of the incomplete-data training corpus in comparison with the estimate of the previous iteration, i.e. $\mathcal{L}(\mathcal{X};\hat{\theta}_r) \geq \mathcal{L}(\mathcal{X};\hat{\theta}_{r-1})$.

Guarantee to Converge The iterative process will converge to a local maximum of the likelihood function.

These guarantees distinguish EM as a well-understood and powerful optimisation algorithm for MLE using incomplete data. While other heuristic optimisation procedures might somehow work for specific tasks, we are still left unsure of the exact pre-conditions that favour their use, as well as over the quality of their output for various input data. In contrast, the two guarantees discussed above clearly delineate what EM can do for the modeller. Namely, given a starting point it will climb and converge to the 'nearest' local maximum of the incomplete data's likelihood under the model.

Discussion In addition to being well-understood, the EM algorithm has proven its effectiveness as an essential estimation tool for various machine learning tasks for more than three decades. Its popularity however has sometimes led to the formulation of iterative estimation procedures, which are then casually presented as 'EM'-algorithms. This confusion is further increased by the fact that EM is more of an algorithmic framework, which still needs to be applied for every estimation problem, than a concrete set of instructions. However, only true instances of the Expectation-Maximization algorithm, following the principles described in this section, inherit the algorithmic guarantees associated with EM. It is important for the Machine Learning practitioner to distinguish between EM and 'EM-like' algorithm instantiations.

EM's magnificent ability to fit latent variables over observed data has in the past sometimes led to its promotion as an omnipotent out-of-the-box tool to perform unsupervised induction of hidden patterns in data, like parses, part-of-speech tag sequences and others. EM is certainly effective in this task, as showcased in its application for the induction of word-alignments under the IBM Models discussed in this chapter. However, EM merely provides us with a tool to fit models involving latent variables to incomplete data. The extent to which this will be helpful in inducing interesting latent patterns is linked to the following factors:

[7] Somewhat more accessible proofs of EM's algorithmic properties can be found in (Beal, 2003; Chen and Gupta, 2010).

Model We need to consider the model's ability to effectively describe the data through latent variables. The increasingly more refined IBM Models again showcase how better models induce better latent patterns.

Maximum Likelihood Objective As EM performs MLE, we must assess the appropriateness of the ML objective for our problem. For models using more parameters than necessary to capture the regularities underlying the data, MLE might overfit the data producing degenerate estimates, as we further discuss in Chapter 3. Also MLE coupled with the model at hand might reveal latent patterns that have little to do with what we were after, e.g. sentence bracketings similar to human-derived references.

Initialisation Point EM converges towards a local maximum beginning from the initialisation parameters setting, which, depending on the form of the likelihood function, can greatly affect the algorithm's output (e.g. see (Goldberg et al., 2008)).

Number of Iterations Depending on the application, sometimes it is better to stop the algorithm well before convergence to avoid overfitting, as is usually performed during word-alignment with the IBM Models. Other times, EM needs surprisingly many iterations to arrive at a good estimate, as discussed in (Johnson, 2007).

Overall, Expectation-Maximization stands out as a highly potent estimation algorithm, whose careful empirical application can lead to discovering latent patterns that go well beyond the surface of the observed training data.

2.7 Generalisation Error and Cross-Validation

Most statistical estimators for parametric models, including Maximum-Likelihood Estimation with which we are primarily occupied in this thesis, select model parameters by fitting the model to the training data. That entails optimising the parameters so that training data error, as computed according to an error function[8], is minimised. In general however, our interest in the model estimate goes well beyond the training data, as our primary concern relates to its *prediction* capability on independent test data. The problem is that frequently, minimising loss on the training data does not necessarily translate to reducing error on test data. Below we try to localise the reasons for this, by distinguishing between two sources of estimator errors, bias and variance.

In addition, we discuss Cross-Validation (CV), a method to arrive at an estimate of the *Generalisation Error* (GE) of a model estimate, i.e. the expected error over all the independently drawn test sets. As for most applications GE

[8] Also sometimes called a loss function.

is exactly the error we would like to minimise, CV can aid in model assessment, evaluating the expected performance of a particular model estimate derived from the training data on yet unseen data. In the next chapter we will also present how CV can be used directly for model estimation, finding the model estimate which is most expected to perform well with future data.

2.7.1 Estimator Bias and Variance

An estimator can be defined as a function of the data which, for a particular data set, returns an estimate of a given quantity. This quantity can be for example a number, like an estimate of the true average of a random variable, but it can also be our estimate of the true distribution from which we are sampling. Two characteristics of estimators that relate to their generalisation error are estimator bias and variance.

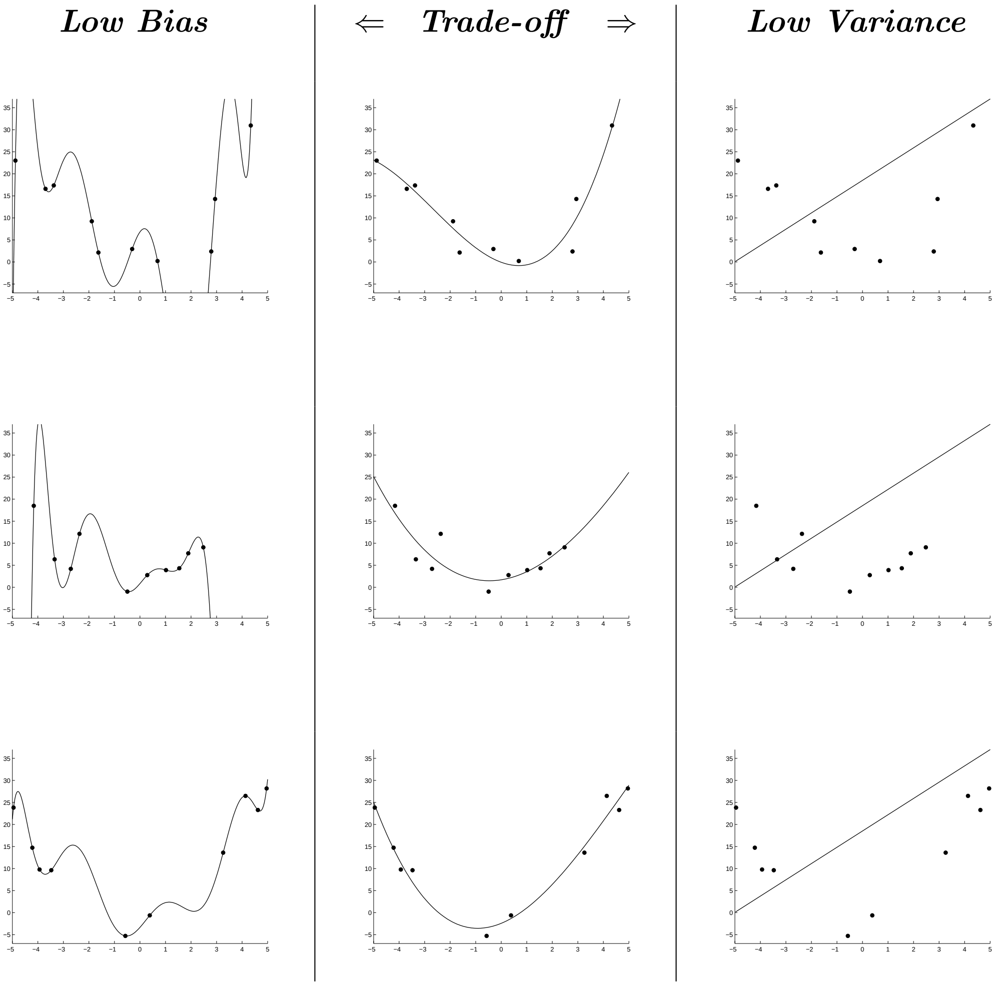
Estimator *bias* is the accuracy of our average estimate (as measured by our error function), when we average over all training sets $\mathcal{X}$ that we can sample. *Asymptotically unbiased* estimators converge to the true value of the quantity estimated as the training set size approaches infinity. This is an appealing property as it guarantees that an estimator will ultimately arrive at an accurate estimate given enough data.

However, we never possess training data of sizes close to infinite. Somewhat surprisingly, an unbiased estimator for smaller training sample sizes frequently produces a large generalisation error. Bias relates to the strength of prior assumptions employed by the estimator, with an unbiased estimator enforcing no such assumptions over the quantity estimated, opting instead to completely rely on the training input. The estimates may become then too sensitive to the training input.

This can lead to increased *variance* between them for different training sets, which entails that many of them will deviate significantly from the true value, leading to generalisation errors. Lowering GE due to estimate variance usually entails increasing our assumptions over the quantity estimated and in this way abstracting away from the training data, i.e. increasing the estimator's bias. Still, at the other extreme, an estimator with zero GE due to estimate variance always outputs the same estimate irrespective of the training data. Unless our strict assumptions behind this estimate are somehow correctly guessed, this is bound to lead to a large GE.

This establishes a trade-off between errors due to estimator bias and those due to variance, where decreasing errors due to bias increases errors due to variance and vice versa. The curve-fitting example of Figure 2.11[9] showcases this trade-off. Both low bias as well as low variance estimators return estimates which widely deviate from the true function $f(x) = x^2$ behind the three noisy samples.

[9] Adapted version of a similar example in (Duda et al., 2001).



Low Bias ⇐ ***Trade-off*** ⇒ ***Low Variance***

Figure 2.11: A *low bias* estimator (9-th degree polynomial) precisely fits each sample, but is penalised in terms of estimate variance. A *low variance* estimator (fixed linear) has zero estimate variance, but its high inherent bias results in a bad estimate of the underlying function. A *trade-off* between bias and variance (cubic) is needed to lower the Generalisation Error.

The challenge then lies in finding the correct balance between estimator bias and variance which minimises generalisation errors overall, as is the case in the second column in the figure.

Bias-Variance Decomposition We may gain additional understanding in the source of an estimator's errors using the GE's *bias-variance decomposition*, breaking down the GE into terms attributed to estimator bias and variance respectively. This decomposition relies on the kind of the estimation and the error function used. In the context of this thesis, it is interesting to consider estimators where the target of our estimation efforts is the distribution generating the data we model.

Assume that we wish to recover the target distribution q by means of an estimator $\hat{p}$ returning the probability estimate $\hat{p}(\mathcal{X})$ when trained on training set $\mathcal{X}$. A sensible error function in this setting can be the Kullback-Leibler (KL) divergence between the target q and estimate $\hat{p}$. For the distinct random variable case this is:

$$KL(q, \hat{p}(\mathcal{X})) = \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{\hat{p}(\mathbf{x}; \mathcal{X})} \tag{2.36}$$

Denoting with $E_{\mathcal{X}}$ the expectation over all training samples, generalisation error is then the expected KL-divergence between q and $\hat{p}$.

$$Err = E_{\mathcal{X}} KL(q, \hat{p}) = E\left[KL(q, \hat{p}(\mathcal{X}))|\mathcal{X}\right] \tag{2.37}$$

Heskes (1998) shows that the GE Err can then be decomposed in bias and variance terms. The bias term is the KL-divergence between q and the mean estimate over all training data $\bar{p} = E_{\mathcal{X}}\hat{p}(\mathcal{X})$. Variance is the expected divergence between the average estimate and the estimator's actual choice for each training input $\mathcal{X}$.

$$Err = \overbrace{KL(q, \bar{p})}^{bias} + \overbrace{E_{\mathcal{X}} KL(\bar{p}, \hat{p})}^{variance} \tag{2.38}$$

An example of an unbiased estimator in this setting would be one which only predicts the training data according to their frequency in the training set. It is easy to show that the average over all sampled training sets $\bar{p}$ would then coincide with the target distribution q leading to a zero bias term. However, excluding random variables taking only a handful of values or having access to extremely large training sets, the variance term of an unbiased estimator becomes unboundedly large, leading to a large GE. In Chapter 3, we shall revisit the bias-variance decomposition in the context of Fragment Models.

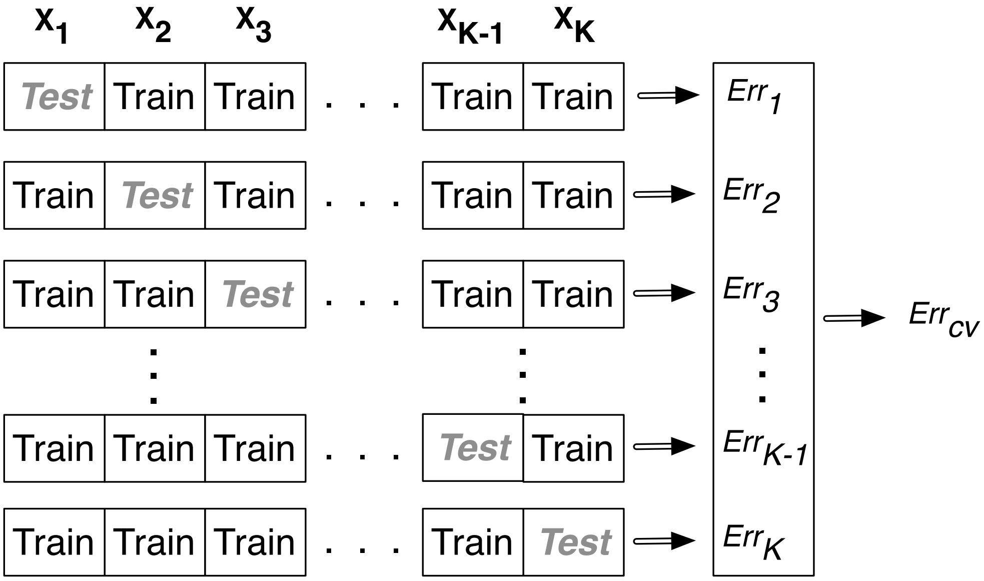


Figure 2.12: K-fold Cross-Validation.

2.7.2 Cross-Validation

The Bias-Variance analysis of estimator Generalisation Error highlights that low training data error alone does not always translate to low GE, due to training data *overfitting*. While providing low training set error remains an intuitive model selection criterion, we need to discriminate between model-estimator combinations which are able to capture the underlying random variable's statistical properties and those which fixate on the training sample's peculiarities.

Given enough data, we could set aside a *validation* set, the error on which can aid in assessing GE. Nevertheless, reserving data for the validation set reduces the size of the training set. This is further aggravated from the fact that often, in order to attain a reasonable estimate of the GE, the size of the validation set must be substantial. Validating thus on a reserved data set is inefficient in employing training data and its use is often prohibitive when assembling training sets is particularly costly.

A simple but highly effective method to arrive at an estimate of the GE without sacrificing possibly scarce training data is K-fold Cross-Validation (CV) (Hastie et al., 2001; Duda et al., 2001). The basic concept is using part of the training data to fit our models and a different *holdout* part to test them, while rotating which part functions as the holdout set. This allows a more efficient usage of our data, as in the end we allow all data points to take part in both fitting the model as well as validating its generalisation capacity over new data.

More precisely, we begin by splitting the data in K roughly equal-sized parts $X^1 \dots X^K$. For every $1 \leq k \leq K$, we test against part X^k a model trained on the rest of the data X^{-k}. In this manner, K estimates of the generalisation error are computed, which we can further combine together to output an overall estimate of the GE, for example by averaging them together. This process is depicted in Figure 2.12.

Usual settings for K which have been shown to work well for a range of modelling problems (Kohavi, 1995) are in the range of 5 to 20, with 10 a popular choice. The case when K is equal to the size of the training set X is referred to as *leave-one-out* CV, as in each CV-round we hold out a single training data point.

CV as an Estimator of GE Cross-Validation is itself an estimator of the Generalisation Error of model-estimator combinations. Due to the No-Free-Lunch Theorem (Wolpert, 1996), which is applicable to all estimators, we cannot of course prove that CV is an overall superior estimator of the GE under all circumstances.

Nevertheless, it has been shown (Kohavi, 1995) that, under some assumptions, CV is both an unbiased as well as low-variance estimator of the Generalisation Error, promoting CV as a highly accurate estimator of GE. The key assumption for this to hold is that the estimators tested under CV are *stable* under the perturbations of the training data set during CV. In other words, that their predictions do not change when trained on the training set with a CV holdout part removed. While this assumption does not strictly hold for most estimators and experimental settings, this result exhibits that CV is expected to be a good estimator of prediction error when for the estimator, training data and number of CV-folds chosen, the predictions of the estimator do not greatly change when presented with the CV holdout parts removed.

Practical Applications Apart from these theoretical properties, CV has been also shown to provide a low-bias, low-variance estimator of GE for a host of 'real-life' problems (see e.g. (Kohavi, 1995; Schaffer, 1993)). In addition, CV has also found numerous applications for NLP problems. Examples include estimating back-off parameters of Language Models (Jelinek and Mercer, 1980; Kneser and Ney, 1995), as well as estimating the parameters of Data-Oriented Parsing models (Zollmann and Sima'an, 2006) and selecting the feature set of a discriminative parsing model (Collins, 2000).

Cross-Validation is mostly applied in the context of *model selection*, picking out the model which the best prediction properties. In the next chapter, we take advantage of the theoretical and practical appealingness of CV as an estimator of Generalisation Error to formulate a model *parameter estimation* objective, which aims at increased generalisation over yet unseen data. We find that this learning objective is a preferred alternative to plain Maximum-Likelihood Estimation for models with a strong tendency to overfit training data. We then discuss this in detail for Fragment Models: a family of models which has already been employed with success for syntactic parsing and machine translation.

Chapter 3

Fragment Models Estimation with the CV-EM Algorithm

Machine Learning problems frequently involve data with an unobserved hidden structure, and these data typically cannot be described by a mere low-dimensional vector. Examples include face and character recognition that work with matrices of image pixel values, financial fraud detection operating on sequences of financial transactions, and automated medical diagnosis systems accepting as input vectors of patient medical variables. In the same category fall many of the problems in NLP like language modelling, speech recognition, parsing and machine translation.

In the first part of this chapter we occupy ourselves with modelling such *complex data*. We begin by a discussion of the overall challenges involved, examining in more detail the interesting case posed by natural language data. We further concentrate on generative models and treat the case of Fragment Models. These define distributions over the data modelled, by specifying how *fragments* of arbitrary sizes extracted from the training data can be combined together to produce novel data instances.

The highly expressive Fragment Models are nevertheless notoriously difficult to train, as they are known to have a strong tendency to overfit training data. Motivated by this, we propose a Cross-Validated MLE (CV-MLE) estimation objective and contribute the Cross-Validated Expectation-Maximization (CV-EM) algorithm. This is a general estimation algorithm, which employs the Cross-Validation criterion to induce models generalising well on yet unseen data. We show that CV-EM enjoys an array of appealing algorithmic properties, preparing the grounds for its application in the following chapters of this thesis.

3.1 Fragment Models

3.1.1 Modelling Complex Data

We employ the term ‘complex data’ to refer to instances whose representation demands a large number of numerical values, with complex patterns between these that cannot trivially be captured in a low-dimensional space. We contrast these to simpler data involving a handful of values per data point, like temperature readings from a single sensor, a collection of basic measurements of people like height and weight or a country’s key financial variables. While working with simpler data can also be highly non-trivial, modelling problems involving complex data share a number of common challenges.

Perhaps the most fundamental issue with regard to complex data modelling is that we can never hope to have access to enough training instances to model the data straightforwardly as atomic data points. For example, one can consider the height, width and weight of various animals in a 3-dimensional space and distinguish humans among them using a Gaussian model or the k-Nearest-Neighbor algorithm. In contrast, the same approaches cannot be routinely applied for the much higher-dimensional spaces of complex data such as the pixels of an image.

We may respond to this challenge by moving past the surface of the data to take advantage of their internal structure and the relations between the data’s variables. In contrast to random data where modelling efforts are in any case futile, real-life data related to ML applications often exhibit such internal organisation. Models or learning algorithms which introduce the right assumptions over these internal interdependencies are able to overcome data sparsity and adequately describe or classify the data when trained on the limited amount of available training instances.

There are multiple methods we might follow to take advantage of these internal data patterns. For example, to perform face recognition we may exploit the correlations between the pixel values to map the image data in a much lower-dimensional space using Principal Component Analysis (Turk and Pentland, 1991). In financial fraud detection we might classify transactions as fraudulent by establishing a hierarchy over the transaction’s variables using a decision tree. Medical variables are often related by encoding conditional independence assumptions between them in a Bayesian Network. Notwithstanding the differences, all of the aforementioned methods coincide in viewing the data as instances whose internal organisation can be exploited to model them from reasonably-sized training sets.

Complex Data in NLP Most of the Natural Language Processing tasks accept as input complex data, e.g. natural language sentences, and in this way are susceptible to the issues highlighted above. The space of possible values is so large that even billions of training instances are not enough to cover a substantial part of it. An interesting experiment to exemplify data sparseness when working with NLP data is searching for an exact match of medium length, well-formed sentences or even phrases against all the content of the web using a search engine, resulting in most cases in zero matches. Fortunately, despite all the talking we have only explored a small fraction of what can be stated through natural language.

However, this does render modelling NLP data challenging. For example, due to this data sparsity a language model based on the relative frequency of whole sentences in the training data can hardly be effective at all, assigning zero probability to most of the yet unseen well-formed sentences. In the same way, for parsing we cannot escape by merely learning the conditional distributions of full parse-trees for every training sentence.

One way to overcome this is to introduce independence assumptions between the variables of the data in a generative model, or employ features based on patterns between the values of these variables in a featured-based approach. For a generative parsing model, we might for example assume that expansions of parse-tree non-terminals are independent of the rest of the previous derivation steps given the non-terminal being expanded, as in Probabilistic Context-Free Grammars.

A typical solution for the purpose of language modelling is to assume a Markovian LM. For example, in a second-order LM, every new word w_i generated is independent of the previous ones given the last two words w_{i-2}, w_{i-1} before it, as in equation (3.2) below. With $w_1 = \langle s \rangle$ and $w_N = \langle /s \rangle$ the start and stop symbols delimiting a sentence, we can write:

$$\begin{aligned} p(w_1 w_2 w_3 \dots w_N) &= p(w_1)p(w_2|w_1)\prod_{i=3}^{N} p(w_i|w_1^{i-1}) && (3.1) \\ &\simeq p(w_1)p(w_2|w_1)\prod_{i=3}^{N} p(w_i|w_{i-2}^{i-1}) && (3.2) \end{aligned}$$

Modelling with Data Fragments The case of Markovian LMs exemplifies how we can address the challenges of modelling complex NLP data by breaking through their surface and considering assumptions over their internal organisation. The exact chain-rule application of equation (3.1) above is associated with an opaque, rigid view of data generated as one piece. In contrast, imposing in the context of generative models independence assumptions like those of a second-order Markovian model, allows us to take advantage of the local nature of many linguistic phenomena to disentangle, as conditionally independent, the words of the sentence that are longer than 2 words apart.

These assumptions, aside from their probabilistic modelling impact on simplifying (3.1) in (3.2), crucially establish tri-grams, word sequences of length 3, as the partially overlapping building blocks of sentences. The result is that, since gathering statistics over tri-grams is less affected by data sparsity than doing so for full sentences, a tri-gram based language model generalises much better.

Along the same lines, we can visualise the generative process of a model employing conditional independence assumptions between the data variables as building the data from partially overlapping data *fragment, data*, with the overlapping part matching the conditioning context between them. The size of these fragments and the way that they are combined depends on the assumptions of the model, with second order Markov LMs employing tri-grams, third order models employing four-grams etc. When training such models over data whose variables are discrete, as is the case for most NLP problems, training often consists of *extracting* such fragments and their associated statistics from the training data. Apart from the n-grams of Markov LMs, further examples include subtrees of depth one for PCFG models trained from treebanks, word-pairs for the IBM SMT models (Brown et al., 1990), contiguous phrase-pairs for the phrase-based SMT models (Och et al., 1999; Koehn et al., 2003) and synchronous subtrees of depth one for Synchronous CFG hierarchical translation models (Wu, 1997; Chiang, 2005a).

Fixed-size and Arbitrary Fragments As discussed above, a lot of the models employed in NLP are based on extracting and learning to recombine fragments of the training data. A common trait of the majority of these models is that they employ elementary fragments of fixed sizes, such as tri-grams, subtrees of depth one or word-pairs. They then define probability distributions over derivations combining these fragments together to generate the data being modelled.

In contrast, a particularly interesting family of models is characterised by the utilisation of fragments of arbitrary sizes. For example, the Data-Oriented Parsing models (Bod et al., 2003) are based on subtrees of arbitrary depth, while phrase-based and hierarchical SMT employs phrases, contiguous and non-contiguous respectively, of arbitrary length. In principle, the size of the fragments combined to generate the modelled data can vary up to the full size of the data points, allowing derivations generating a data point as a single fragment and in a single generative step. We refer to these models in the rest of this work using the term *Fragment Models* (FMs).

3.1.2 Data-Oriented Processing

Studying the Data-Oriented Processing paradigm (Scha, 1990; Bod, 1992), one of the earliest frameworks leading to the formulation of Fragment Models, is interesting both to highlight the potential of FMs as well as discuss the challenges involved in their training. Data-Oriented Processing was initially applied to the supervised learning of natural language parsing and later employed for unsupervised parsing (Bod, 2006) as well as translation (Poutsma, 2000; Way, 1999) among others. The application of this paradigm for NLP tasks stems from the basic assumption that human language perception and production works with representations of concrete past language experiences rather than with abstract linguistic rules. These are maintained in the form of memorised fragments of arbitrary sizes from previous language utterances that a language user has been exposed. New linguistic input can be analysed or novel linguistic output can be produced by combining these fragments together.

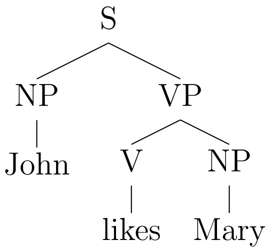
The same assumptions can be employed in Artificial Intelligence terms to arrive at a Fragment Modelling approach. Here, the role of the human language user is substituted by an empirically estimated probabilistic model, with prior linguistic experiences embodied in the training corpus. From this corpus arbitrarily-sized fragments of data points are extracted, which a generative process can combine together to arrive at new data. A stochastic model instance over this process provides a distribution over these novel combinations, distinguishing between highly probable and less probable ones. In total, a framework to define such Fragment Models, along the lines it was first developed for Data-Oriented Parsing can be drawn in terms of the following components (Bod, 1995).

- A definition of a formal representation for data analyses.
- A definition of the fragments of the analyses that may be used as units in constructing an analysis of a new data point. The size of the fragments varies up to covering the full data point analysis as a single fragment. This last property crucially distinguishes FMs from other modelling paradigms.
- A definition of the operations that may be used in combining fragments.
- A probabilistic model over the derivations of data points through the combination of fragments.

Data-Oriented Parsing The first application of the Data-Oriented Processing framework was in the context of Data-Oriented *Parsing* (DOP) (Bod et al., 2003) which can also function as an interesting example of a Fragment Model. In DOP we are interested in modelling the constituency parsing analyses of natural language sentences.

The more traditional approach in modelling the latter is through Probabilistic Context-Free Grammars (PCFGs). Each PCFG $\mathbf{G}$ is a 5-tuple $\langle V, T, S, R, P\rangle$: a finite set of Non-Terminal (NT) symbols V, a finite set of terminal symbols T, a designated 'start' symbol $S \in V$ and a finite set of rewrite rules R expanding a left-hand side (a single NT) to a right-hand side string of terminals and NTs. P is a set of probabilities $\{p(r|\text{LHS}(r))\}$, one for each rule in R, with the probabilities of all rules with the same left-hand side summing up to one.

The PCFG explains the derivation of a parse tree starting from the start symbol S, by recursively rewriting NTs using rules of the grammar. Each time, the rewriting operation is applied to the left-most NT which has not yet been



$$\begin{aligned} r_1 &: \quad \text{S} \rightarrow \text{NP VP} \\ r_2 &: \quad \text{NP} \rightarrow \text{John} \\ r_3 &: \quad \text{VP} \rightarrow \text{V NP} \\ r_4 &: \quad \text{V} \rightarrow \text{likes} \\ r_5 &: \quad \text{NP} \rightarrow \text{Mary} \end{aligned}$$

$$p(T) = \prod_{i=1}^{5} p(r_i|\text{LHS}(r_i))$$

Figure 3.1: Constituency parse tree T with the PCFG rules r of its derivation. The probability $p(T)$ of the parse tree is the product of the derivation's rule probabilities.

expanded, to avoid the spurious ambiguity between derivations employing the same rules in a different order. The rule applications are considered independent of the rest of the derivation given the NT that they expand. The probability of a full derivation is the product of the probabilities of the rules that were employed, with these probabilities summing up to one for all rules with the same NT as their left-hand side, as dictated by the aforementioned independence assumptions. An example of a constituency parse tree together with the PCFG rules taking part in its derivation and the computation of its probability can be seen in Figure 3.1.

We already discussed in the previous section that PCFG derivations can be seen as combining together parse fragments to derive a full parse tree, albeit of a fixed size: subtrees of depth one. In DOP, we move past this constraint to extract tree fragments of arbitrary sizes from the training corpus of sentence constituency parses and learn how to combine them together in derivations of novel sentence instances. As fragments we consider subtrees (i.e. tree fragments with a single root) of arbitrary depths, with the conditioning context remaining as in the case of the PCFGs the root of the subtree. Considering subtrees of arbitrary depth implies also including the *complete* parse tree in the set of subtrees. Figure 3.2 lists a subset of the subtrees that can be extracted from the tree of Figure 3.1.

More formally, a DOP probabilistic grammar is again defined as a 5-tuple $\langle V, T, S, R, P\rangle$ along the same lines as a PCFG. However, each rule $r \in R$ replaces a left-hand side single NT, to a right-hand side *sub-tree* of terminals and NTs having the left-hand side as root, in contrast with PCFGs where expansions lead to terminal and NT strings. For this, DOP grammars are categorised in the literature in the family of Tree-Substitution Grammars.

Each rule has an associated probability attached to it, with these probabilities again summing up to one for all rules with the same left-hand side NT. Each derivation, starting from the start symbol S, rewrites left-to-right non-terminals

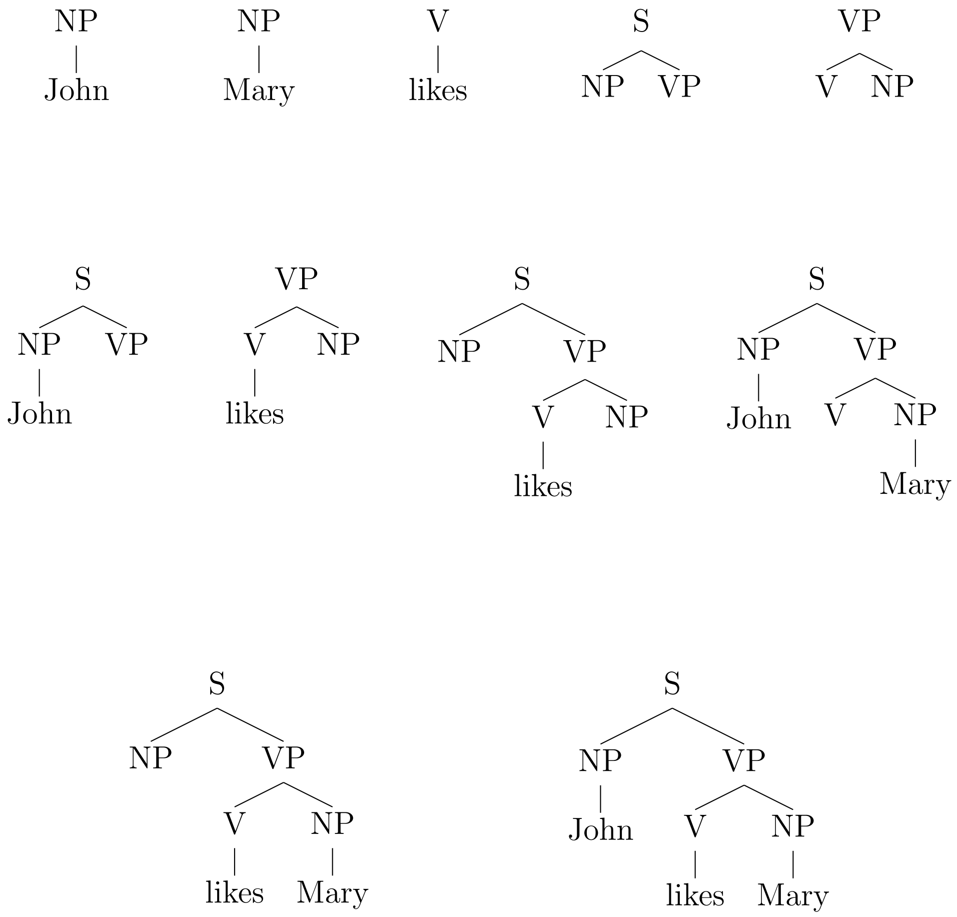


Figure 3.2: Some of the subtrees that can be extracted from the constituency parse in Figure 3.1. The first row depicts subtrees of depth one that are also the units of PCFG derivations. However, DOP extracts and reuses in derivations also subtrees of arbitrary depth, up to the complete parse tree.

to their subtree expansion, by applying a rule with the current leftmost NT as its left-hand side. The probability of a derivation D is, along the same lines as PCFGs, equal to the product of the probabilities of the rules r taking part in it.

$$p(D) = \prod_{r \in D} p(r|\mathrm{LHS}(r)) \tag{3.3}$$

Three derivations of a novel sentence based on a subset of the fragments of Figure 3.2 are listed in Figure 3.3. The first derivation reuses the same fragments as a PCFG derivation of the parse would employ, highlighting that PCFG derivations with elementary fragments is also part of the space of derivations that DOP considers. The second and third derivations however employ a subtree reaching down to depth two, which in the second derivation encodes the dominant Subject-Verb-Object sentence structure of English, while the third one memorises the argument structure of the verb 'likes'.

Latent Segmentation The introduction of Data-Oriented Parsing, apart from showcasing the descriptive power of Fragment Models, also gradually revealed the challenges involved into estimating such models from training data. Context-Free Grammars provide only a single derivation for each parse tree, if we agree to only substitute NTs in a left-to-right fashion as mentioned above. This is a crucial point, as it allows us to relate a training parse tree to a single derivation behind it and in this way translate the observation of the tree to the observation of the unique sequence of CFG rules taking part in its derivation. For the purpose of PCFG training, this enables us to treat a corpus of training parse trees (a treebank) as *complete* data[1], which simplifies training under a Maximum-Likelihood objective to an instance of Relative Frequency Estimation.

However, while the same left-to-right constraint allows us to avoid spurious ambiguity between DOP derivations employing the same subtrees, as Figure 3.3 reveals under DOP we can still arrive at the same parse tree under multiple derivations, each employing a different set of fragments for this purpose. The probability of a full parse T is then the sum of all derivations $D \overset{*}{\Rightarrow} T$ leading to T.

$$p(T) = \sum_{D \overset{*}{\Rightarrow} T} p(D) \tag{3.4}$$

Most importantly, DOP and Fragment Models in general introduce in this manner a latent *segmentation* variable. This dictates how a data point is segmented into fragments, as there is more than a single way to do this. While in DOP latent segmentation is not explicitly encoded in a separate model variable, it still is embedded in the model in the form of subtree expansion probabilities, which indicate preferences for substitution of NTs by larger or smaller fragments.

[1] For a discussion of complete vs. incomplete data please refer to section 2.6.

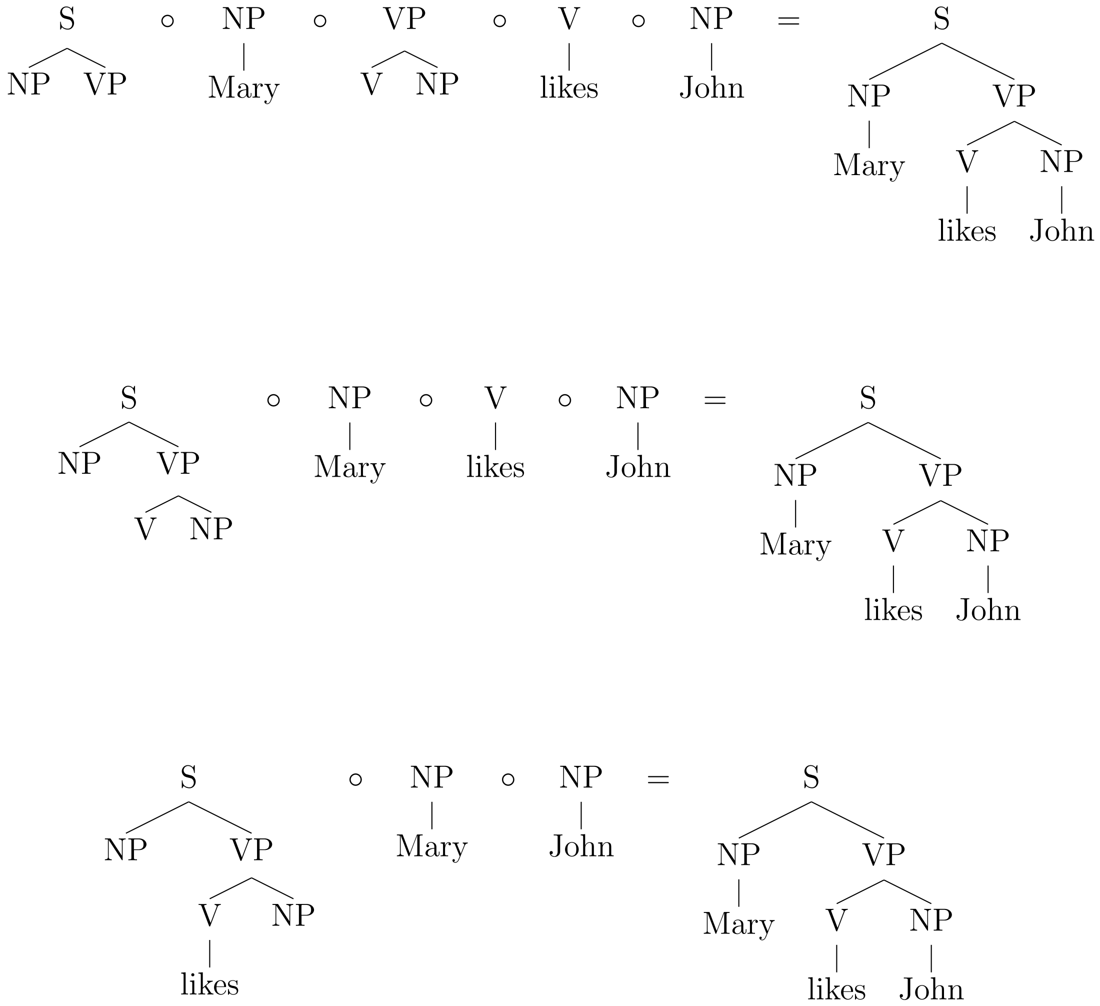


Figure 3.3: Three DOP derivations of the same parse tree.

Because of the latent segmentation, the treebank training set must be considered as *incomplete* data, as it does not contain information on the segmentation of parse trees into DOP subtrees. For parameter estimation, we need thus to disambiguate between the segmentations of data points into fragments.

3.1.3 DOP Estimation

At first sight, one could apply Maximum Likelihood Estimation for DOP, as it is successfully applied for PCFGs in the form of Relative Frequency Estimation. The fact that treebanks are incomplete data for the purpose of estimating the parameters of a DOP model, disallows the application of the relatively easy RFE. Nonetheless, we can still formulate an MLE estimation objective, as we already discussed in section 2.6, and maximise the likelihood $\mathcal{L}(\mathcal{T})$ of the training data $\mathcal{T}$ by summing through the alternative derivations of each training instance T.

$$\mathcal{L}(\mathcal{T}) = \prod_{T \in \mathcal{T}} \sum_{D \overset{*}{\Rightarrow} T} p(D) \tag{3.5}$$

However, this vanilla MLE objective is of little use to estimate the parameters of DOP models, as we discuss in more detail in the wider context of Fragment Models in the next section. In rough terms, the problem is that the MLE objective of fitting the training data leads to allocating all probability mass to full training parse trees, as these are also part of the subtrees extracted from the training set (Prescher et al., 2004). Doing so allows the model to exactly predict the training treebank. Crucially, while it seems superficially desirable to arrive at parameters which fit well the training data, in this case it completely defeats the purpose of learning such a model, as it assigns no probability mass to any analyses of sentences past those included in the training parses (Zollmann and Sima'an, 2006). The MLE estimate of DOP probabilistic grammars has an extremely limited ability to generalise to yet unseen data instances.

In the face of this, right from the introduction of DOP, there has been a barrage of work on estimating DOP models, resulting in a fair amount of progress and increased understanding of the issues involved, but still failing to decisively resolve the problem of estimation for DOP models. We briefly examine the key points of such estimation approaches, as an interesting overview of possible solutions to Fragment Model estimation that have already been tried.

DOP1 Estimator The introduction of the DOP model for parsing was coupled with the DOP1 estimator (Bod, 1995). Under DOP1, the probability of a DOP subtree is set to the relative frequency of its 'appearance' in the training treebank. More accurately, this estimator belongs to the family of estimators employing *extraction* heuristics, that we discuss in the next section. Namely, we compute the DOP1 estimate of a treebank by first extracting all DOP subtrees of the training parses, counting how many times each fragment can be extracted. The DOP1 estimate is then set to the relative frequency of the subtree fragments in this multiset of subtrees constructed in the previous step.

A somewhat subtle but crucial point is that the DOP1 estimate has no connection to the relative frequency of any events in the training data, as the training treebank does not contain any information relating to the segmentation of parses into fragments as we already mentioned. As the extraction step lacks any clear link to the training data by means of an optimisation objective, the DOP1 estimate is a heuristic estimator, with an appealingness relating only to the strength of the results from its empirical application. Its arbitrary nature from a theoretical standpoint is further highlighted by (Johnson, 2002) who shows that DOP1 is a biased and inconsistent estimator. Additionally, (Bonnema et al., 1999) notice that the DOP1 extraction heuristic in practice favours larger extracted subtrees over smaller ones.

Bonnema et al. Estimator Bonnema et al. (1999) propose instead an alternative estimator which assigns to every appearance of a subtree in the training data, a count equal to the fraction of the number of possible DOP derivations using the subtree fragment, against the total number of DOP derivations of the parse where it appears. Essentially, this boils down to Maximum Likelihood estimation on a treebank whose segmentation of every parse tree in DOP subtree fragments has already been disambiguated by the strong assumption that all segmentations are equally likely. Sima'an and Buratto (2003) show that this estimator is inconsistent and discuss that it is biased towards smaller subtrees and does not perform well in practice.

Back-Off Estimator Starting from the DOP1 estimate, (Sima'an and Buratto, 2003) propose a back-off estimator based on Katz smoothing technique (Katz, 1987) to discount probability mass from the larger subtrees towards smaller ones. As (Zollmann and Sima'an, 2006) discuss, this estimator both inherits the weaknesses of the DOP1 estimate, as well as loses some of the appealing properties of Katz smoothing through problems related to the estimator's practical implementation.

Parsimonious-DOP Estimator Starting from the PCFG relative-frequency estimate of the training treebank, which employs minimal subtrees of depth one, (Zuidema, 2007) propose an estimator which distributes probability mass to larger subtrees by evaluating their extraction frequency against its expectation. It also includes a counter-balancing bias against large trees to avoid completely overfitting the treebank. While the Parsimonious-DOP estimator works in the opposite manner than the Back-Off estimator, moving overall probability mass from smaller to larger subtree fragments instead, it shares with it the weakness of being based on the DOP1 heuristic, as it compares model subtree extraction expectations against the DOP1 extraction counts.

Shortest Derivation Parsing While not involving the estimation of a model, an alternative approach to the problems related to estimating DOP model parameters is brought forward by (Bod, 2000): shortest derivation parsing. It abandons probabilistic modelling altogether, in favour of parsing by recovering the shortest DOP derivation of a parse tree covering a test sentence, by employing subtree fragments extracted from the training treebank. This parsing heuristic objective favours large subtrees as does the DOP1 estimator and, perhaps for exactly this reason, it is shown to perform competitively against the latter.

DOP* Estimator The DOP* estimator proposed in Zollmann and Sima'an (2006) is of particular interest to this work. The authors briefly consider the possible use of Cross-Validation (CV) to avoid overfitting towards a Maximum Likelihood estimate of a DOP model's parameters which fails to generalise. However, they do not pursue this approach. They argue that while such a learning objective might be theoretically appealing, in practice it involves employing hill-climbing algorithms such as the Expectation-Maximization algorithm, which do not guarantee arriving at the overall ML estimate but only at a local likelihood optimum.

With a primary objective of arriving at a consistent estimator, the DOP* estimator instead couples the shortest derivation principle with CV to disambiguate the segmentation of the treebank parses in DOP derivations: after partitioning the treebank in 10 parts, for every part they consider the shortest DOP derivation(s) of each parse utilising exclusively subtree fragments from the remaining parts of the treebank. From this subtree-segmented treebank they arrive at a DOP estimate by ML estimation which, operating on DOP derivations in place of unsegmented parse trees, boils down to relative frequency estimation. This estimation approach is shown to be consistent, i.e. to arrive with a probability approaching one at an accurate estimate of the true distribution of parse trees when the training treebank size grows towards infinity.

Despite the asymptotic consistency properties of the DOP* estimator, its application on treebanks of real-life sizes still demands a 'leap-of-faith' concerning the use of the shortest derivation principle. In contrast, in this thesis we draw inspiration from certain aspects of the work on the DOP* estimator to pursue and study the implications of a Cross-Validated Maximum-Likelihood learning objective, implemented in the form of our Cross-Validating Expectation-Maximization algorithm.

3.1.4 Modelling with Fragment Models

In the previous section we reviewed in some detail the developments in employing a Fragment Model for natural language parsing in the form of the DOP framework. However, models which can be categorised as FMs have been also introduced for other tasks, with contiguous and non-contiguous phrase-based Statistical Machine Translation being the most relevant in respect to this work. Irrespective of the particularities linked to every kind of data or process that needs to be modelled, FMs enjoy certain common properties which we will now discuss.

A Fragment Model can be considered a *hybrid* between more traditional *generative* models on the one hand and *example-based* models on the other. Typically, generative models describe the derivation of data points through generative steps involving minimal, usually fixed-size, units. Defining what constitutes such a unit rests with the modeller: for example, a language model can operate at the word, morpheme or even letter level. Still, the crucial property of these units is that they are considered atomic, as they cannot be in turn constructed from other such modelling units. Furthermore, learning and applying a generative model is usually a two-step process. First, generative models are induced from the training data according to a learning objective or using a training algorithm, and the learnt model is subsequently applied to process the test data.

In contrast, frameworks categorised as memory-based or example-based process novel data by reusing memorised training examples. For instance, Example-Based Translation (EMBT) (Nagao, 1984) strives to employ these to translate novel sentences by recombining translation fragments from the memorised examples. These fragments can be of arbitrary sizes, with the larger fragments preferred (Sato and Nagao, 1990). Part of an EBMT system is the process of choosing which translation fragments to use and how to recombine them. Typically, this process does not depend on a prior model induction step, accepting the entire training data as input and outputting a model describing translation by means of such fragments, as is usually the case with generative models. On the contrary, the related decisions are considered and scored for every test data point.

Fragment Models stand in the middle, aiming to pick the 'best of both worlds'. On the one side, in comparison to traditional generative models employing minimal atomic units, they crucially consider generating data from units whose involvement in the derivation of a data point can be replaced by an alternative analysis employing other, smaller units. As examples, a parse analysis involving a larger DOP subtree can be substituted by an analysis which arrives at the same constituency structure by a combination of smaller subtree fragments; a translation phrase-pair can be constructed as a combination of smaller phrase-pairs or even word-pairs. On the other side, in comparison to memory-based frameworks, their formulation by means of a stochastic generative process, allows us to encode the construction of novel data instances from fragments extracted from the training corpus, by means of the perhaps better-founded and easier to understand probabilistic models.

By moving past atomic units to larger fragments, while crucially remaining within the context of a generative framework, Fragment Models have a chance to overcome the inherent weaknesses due to the blanket independence assumptions behind traditional generative models. An interesting example of when this is desirable is the modelling of the translation of idiomatic expressions in natural language such as 'kick the bucket'. Such expressions typically exhibit correlations between their words and their translations that severely violate the independence assumptions behind word-based models; word-to-word translation statistics are not enough to direct us toward an adequate translation for the aforementioned example phrase. An FM enjoys the ability to reserve some probability mass from more fine-grained analyses that assume some degree of conditional independence between their generative steps. It can then assign this mass to directly model the translation of such idiomatic expressions as single units, in this case translate 'kick the bucket' as a single contiguous phrase.

3.1.5 MLE and Fragment Models

As already highlighted above through following the development of the literature on DOP estimation, using a Fragment Model does not come without disadvantages. The central issue is that due to the introduction of the latent segmentation variable, estimating the parameters of FMs is far from straightforward. In addition, FMs ability to bypass the independence assumptions of models employing smaller fragments allows it to fit the training data arbitrarily well. This is a blessing that, unless properly treated, can easily develop into a curse in the form of overfitting.

For example it is easy to see that the often successful Maximum Likelihood Estimation objective completely overfits the training data when applied to estimate FMs, as discussed in (Prescher et al., 2004). The MLE objective of equation (2.30) can be interpreted as minimising the Kullback-Leibler divergence between the empirical relative-frequency distribution $\tilde{p}(\mathbf{x})$ of values of the data variable X in the training data $\mathcal{X}$, and the model estimate $p(\mathbf{x};\theta)$ parameterised by θ.

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \mathcal{L}(\mathcal{X};\theta) = \arg\max_{\theta} \prod_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x};\theta) \\ &= \arg\min_{\theta} KL(\tilde{p}(\mathbf{x}) \mid\mid p(\mathbf{x};\theta)) = \arg\min_{\theta} \sum_{\mathbf{x}\in\mathcal{X}} \tilde{p}(\mathbf{x}) \log \frac{\tilde{p}(\mathbf{x})}{p(\mathbf{x};\theta)} \end{aligned} \tag{3.6}$$

We already mentioned that we considered a key feature of FMs to be able to model the training data with fragments of arbitrary sizes, up to considering a data point as a single fragment emitted in a single generative step. This entails that there exists a parameter setting $\tilde{\theta}$ for which all values $\mathbf{x}$ of random variable X are emitted in a single generative step, according to their relative frequency in training data $\mathcal{X}$. As these relative frequencies sum up to one, this leaves no probability mass for analyses employing smaller fragments than whole data points.

But for $\theta = \tilde{\theta}$ we have $KL(\tilde{p}(\mathbf{x}) \mid\mid p(\mathbf{x}; \tilde{\theta})) = 0$ and since the KL divergence between any two distributions is always larger or equal to zero, $\tilde{\theta}$ is the ML estimate of θ as it minimises (3.6). Overall, the MLE estimate of Fragment Models completely overfits the training data by predicting nothing more than the training points according to their empirical relative frequency.

3.1.6 Expected Error of MLE for Fragment Models

Taking the result of the previous section into account allows to shed some light on the source of the generalisation error incurred by the MLE estimates of FMs, as analysed in terms of estimator bias and variance. In section 2.7.1 we discussed how the expected generalisation error for an estimator of a distribution $\hat{p}(\mathcal{X})$ trained on data $\mathcal{X}$, can be analysed in estimator *bias* and *variance* terms. Let us measure the estimator's expected error $Err(\hat{p})$ in terms of the expected Kullback-Leibler (KL) $E_{\mathcal{X}} KL(q, \hat{p})$ divergence between the estimator's output and the target distribution q. Then, the error $Err(\hat{p})$ can be analysed in a bias and a variance term according to equation (2.38), which we repeat here for the reader's convenience.

$$Err(\hat{p}) := E_{\mathcal{X}} KL(q, \hat{p}) = \overbrace{KL(q, \bar{p})}^{bias} + \overbrace{E_{\mathcal{X}} KL(\bar{p}, \hat{p})}^{variance}$$

The bias term is the KL-divergence between q and the mean estimate over all training data $\bar{p} = E_{\mathcal{X}} \hat{p}(\mathcal{X})$. Variance is the expected divergence between the average estimate and the estimator's actual choice for each training input $\mathcal{X}$.

The fact that the MLE estimate $\hat{p}(\mathcal{X})$ in the case of FMs predicts exactly the training data according to their empirical relative frequency, has consequences for both terms of the estimator's error. On one hand, MLE is a *zero-bias* estimator for FMs: The average estimate $\bar{p}$ will coincide with the target distribution q when we average over all training sets $\mathcal{X}$, which themselves are sampled from q. On the other hand, since every estimate assigns zero probability to any value not appearing in the training data $\mathcal{X}$, in all but trivial cases the variance term will be *unboundedly large*.

The end result is that the expected error of the MLE estimate is extremely large even though the estimator is zero-biased. This is a typical instance of the estimator bias-variance trade-off, when merely aiming to minimise one of the two error terms severely increases the other. To arrive at an overall low expected error, we need to trade bias error by relaxing how closely our estimator fits the training data, in order to reduce the error attributed to the variance between the estimates. For Fragment Models this translates to shifting probability mass from excessively large fragments to smaller, more reusable ones, according to our expectation for them to appear in further samples from the target distribution than the training data we currently have at hand.

3.1.7 Fragment Models and Generalisation

The overfitting behaviour of MLE estimators for FMs should not discourage us from employing them. Part of the model space of FMs are also estimates which solely employ atomic fragments in their analyses. This means that an FM always includes an estimate which fits the data as well as models employing atomic, fixed-size fragments. Combining this with the result above, what we essentially learn is that FMs have the ability to provide model estimates which cover the continuum between the best data fit provided by their traditional generative counterparts and completely overfitting the training data, by shifting probability mass from more generic explanations of the data to more specific ones. FMs should be thus seen as very versatile models which can accurately describe any training data set. Our focus should then be directed towards employing this ability to also accurately model yet unseen data.

Choosing how closely the training data should be fit is, as is often the case in Machine Learning, a *data-centric* issue. The more training data we have at hand, the larger the probability mass we can reserve for larger fragments, with the overfitting unconstrained MLE estimate we discuss above being the optimal choice as data grows towards infinity and $\tilde{p}(\mathbf{x})$ converges towards the true data distribution. Smaller training data sizes demand focusing probability mass on derivations which employ smaller fragments which we hope will generalise better. But not all fragments of the same size are equally good at generalising: some of them might come forward in data as noise, other as particular instantiations involving combinations of smaller fragments which roughly follow the independence assumptions assumed, while others might signify a departure from the same assumptions that needs to be captured. We believe that navigating this treacherous field demands a data-driven approach, as the one we propose in the form of the CV-EM algorithm later in this chapter and apply empirically in the rest of this work.

3.1.8 Inducing Fragment Models

A direct application of Maximum Likelihood Estimation, apart from the unconstrained case, has also been empirically shown to perform poorly even when the size of the fragments is constrained so that complete overfitting is avoided (DeNero et al., 2006). Given the difficulties of establishing an ML objective that leads to estimates which generalise well, there has been an array of research directions towards alternative FM estimation approaches. We discuss below two of these: the extraction heuristic, which despite its heuristic nature is still applied in most state-of-the-art implementations of FMs, and Bayesian induction of FMs which employs probabilistic priors to arrive at reasonable FM estimates in a more principled manner.

The Extraction Heuristic The Extraction Heuristic estimates the parameters of a Fragment Model in two steps. Firstly, a corpus of extracted fragments is constructed from the training data, extracting all fragments assumed from the FM from every training data point and assigning them a frequency equal to the number of times they were extracted. Then, we arrive at an FM estimate by applying Relative Frequency Estimation on this fragments corpus. This leaves the estimate only related to the original training corpus of complete data points by means of the heuristic extraction process. As we have already commented, the Extraction Heuristic has nothing to do with RFE on the training corpus itself, as the latter does not provide information on any events that are related to its segmentation in fragments. We cannot then just 'count' fragment appearances on the corpus, as the segmentation variable is hidden.

Nevertheless, while it remains difficult to understand what the Extraction Heuristic optimises, its straightforward implementation and the relatively strong results obtained through its employment resulted in it being the estimator proposed during the introduction of both DOP (Bod, 1995), as well as phrase-based translation by means of both contiguous (Och et al., 1999; Koehn et al., 2003) and non-contiguous phrase-pairs (Chiang, 2005a). Going further, the Extraction Heuristic remains a competitive estimator for state-of-the-art systems up to this day.

However, due to their heuristic nature, these estimates have limited theoretical appeal and leave open the question how much better an estimate that maximises some meaningful objective function can do. Also, as they operate on the surface of the training data ignoring the latent variables of FM models, the risks involved in their application grow as the latent variable of the proposed Fragment Models becomes more involved, as we have discussed when examining the Syntax Augmented MT models in section 2.4.3. For this, there has been a significant amount of work on alternative, better founded and understood approaches to induce FMs, such as a large part of this work, as well as the work on the Bayesian induction of FMs which we discuss next.

Bayesian Induction One way to address the inherent tendency of Fragment Models to overfit the training data is by means of a Bayesian prior over the model space. This allows us to encode in the prior a certain preference over parts of the model space which we believe might better generalise. The actual parameterisation of the model is then typically marginalised out. Two practical Bayesian inference approaches that have been applied to induce FMs are Variational Bayes and Gibbs sampling. Variational Bayesian EM (also named Variational Bayes)[2] provides an iterative algorithm to arrive at a local maximum of the marginal likelihood:

$$p(\mathcal{X}) = \int_{\theta} p(\theta)\ \ p(\mathcal{X};\theta) \tag{3.7}$$

A Gibbs sampler (Geman and Geman, 1984) is a Markov Chain Monte Carlo method to sample from the model distribution where the model parameters have also been marginalised out.

There is no doubt that prior knowledge reaching past the training data can often prove highly successful. However, in this case there is no ‘expert’ to consult on which fragments or which parts of the model space to prefer and no clear reason to favour one model estimate over the other *prior* to observing the training data, apart from our experience that models which favour extremely large fragments tend to overfit and do not generalise well. There have been two main directions on choosing such priors, both of which aim to avoid models reserving too much probability mass for overly large fragments: preferring sparse fragment distributions and preferring smaller fragments.

The first direction is employing priors preferring *sparse* fragment distributions which assign most probability to a small subset of the data fragments, favouring more parsimonious model formulations. For example, this can be achieved by means of a sparse Dirichlet prior, with (Zhang et al., 2008a) employing such sparse prior in a Variational Bayesian approach to disambiguate the segmentation of sentence pairs in contiguous phrase-pairs. The second direction is priors preferring *smaller* fragments. For example, (Blunsom et al., 2009) employ a Dirichlet Process prior with a base distribution with a preference for smaller phrase-pairs in a hierarchical translation model.

However, the strong overfitting behaviour of FMs employing large fragments entails that they can assign extremely large likelihood values to the training data. Examining equation (3.7) above where we marginalise over the product of data likelihood and prior probability, reveals that for a Bayesian prior to counter this it needs to penalise models employing large fragments equally strongly. Zhang et al. (2008a) find that a good choice for the Dirichlet hyperparameter α (which must satisfy $\alpha > 0$) is the extremely low value $\alpha = 10^{-100}$. Blunsom et al. (2009) use a base distribution including a Poisson distribution over the phrase-pair length with unit mean.

Overall, for both prior designs, small fragments are strongly preferred, with larger fragments having a higher chance to be sampled only when they appear very frequently in the data. While this approach does allow to expand to fragments past the minimal set without overfitting, imposing a blanket preference for small fragments is a bias which might prevent discovering larger fragments that could

[2] An overview of Variational Bayesian approaches and applications can be found in (Beal, 2003).

be nevertheless useful to better model yet unseen data.

In the next section, we introduce the CV-EM algorithm, a data-driven approach towards estimation which explicitly aims at model estimates which generalise well. In later chapters, we see how the CV-EM can be applied to arrive at Fragment Model estimates for Machine Translation which perform well on test data, with both a clear optimisation criterion (in contrast to the Extraction Heuristic) and a data-driven approach to avoid overfitting (in contrast to enforcing an external prior).

3.2 Cross-Validated Expectation-Maximization

The central problem in Machine Learning is bridging the gap between the limited sample that makes up our training data and the yet unseen test data. This necessarily involves abstracting away from the actual training sample to capture the general properties of the data being modelled, so that what is learnt from the training set can hopefully extend to novel data instances. If no abstraction from the training data is necessary to solve a problem and merely looking them up is sufficient, then this falls more into the realm of databases and could hardly be considered an ML problem. The art of Machine Learning then lies in successfully choosing the level of abstraction from the training points and sorting out the characteristics of the underlying data distribution from the peculiarities of the training instances. Cross-Validation (CV) provides a simple yet powerful method to evaluate how well a learner does in this respect.

As we presented in more detail in section 2.7, given a model for the random variable behind the training data and an estimator for its parameters, K-fold Cross-Validation (CV) provides a method to estimate the Generalisation Error (GE), the error over yet unseen data of the model instances selected by the estimator, by employing the training data itself. It is able to do so, by first partitioning the training data in K parts. Then, in K rounds, each time a different part from the training data is held out, to assess on it the prediction error of the model instance selected by training on the rest of the data. The outcomes of these K rounds are then combined together to arrive at a single estimate of the GE. Notably, as we discussed in the previous chapter, CV is a low bias, low variance estimator of the GE, allowing a fairly accurate prediction of how useful a learner applied on the training data at hand is expected to be for yet unseen data points.

These features have established CV as a widely used approach for *model selection*, i.e. choosing which model, out of a limited set of possible options, is best suited for a particular learning problem, by picking the model which offers the lowest GE as estimated by CV. Here in this work, we move further to describe how Cross-Validation can be employed for *parameter estimation* of models employing latent variables.

We begin by discussing how the training data themselves are used in practice during the modelling process to define which hypotheses over the values of the model latent variables we will consider, with the risk of overfitting the training corpus. To avoid this, we formulate a Cross-Validated MLE (CV-MLE) learning objective, aiming at model estimates which generalise better. CV for GE estimation considers the error on each part of the training data of a model trained by excluding that part. In the same way, CV-MLE seeks the estimate which maximises the likelihood of each training data part, by excluding hypotheses over the values that the latent variables take for it, and which we would not consider if we excluded this part from the training data.

As for plain MLE, it is frequently not possible to compute the CV-MLE estimate analytically for models with latent variables. With this in mind, we propose CV-EM, a Cross-Validated instance of the EM algorithm to allow CV-MLE parameter optimisation from incomplete data. In the rest of this chapter, we present both the CV-MLE estimation criterion and the CV-EM algorithm that allows us to optimise parameters according to it. We compare our framework with related approaches on estimation towards increased generalisation and discuss how CV-EM can be applied to estimate the parameters of Fragment Models.

3.2.1 Pitfalls of Model Extraction

Maximum Likelihood Estimation, estimating the parameters of statistical models so as to maximise the likelihood of the training data, is one of the most widely applied estimators in the Machine Learning literature. When a model with no latent variables is trained from complete data, MLE boils down to the familiar Relative Frequency Estimation. However the MLE estimation objective can also be applied to train models with latent variables from incomplete data. In these cases, an MLE estimator is frequently implemented as an instance of the Expectation-Maximization algorithm, which allows us to climb the likelihood with respect to the model parameters until a local maximum is reached. Pairing the MLE optimisation objective with the EM-algorithm in this way allows us to discover latent data patterns, such as the word-alignments between sentence pairs in Statistical Machine Translation (see section 2.2).

Crucially, for many models explaining complex data, the parameter estimation process where MLE is applicable is preceded by an implicit step, where the training data are used to establish the model parameter space. A particular modelling framework, for example Phrase-Based Statistical Machine Translation, can be seen as a function which when applied on the training data returns a parametric model, using the training data in this way to set up the model's parameter space. For our example, the output of this function would be the space of conditional distributions between source phrases and their possible target phrase translations, as they appear on a word-aligned training parallel corpus. We will refer to this as the *model extraction* step.

A model extracted in this way establishes a set of hypotheses over the target distribution $q(\mathbf{x})$ we are trying to model, with every parameter setting mapping to one such hypothesis. Interestingly, all of these hypotheses can be considered to be *suggested* by the training data themselves, as these are employed to set the model parameter space during the model extraction step. This renders choosing between these hypotheses during estimation by fitting the training data (as we do in MLE) dangerous, as we might be *testing hypotheses suggested by the data*. In this way, we risk arriving at estimates which succeed in little more than predicting the particular instances contained in the training data, missing the chance to discover the underlying patterns.

In the case of models with latent variables, each such hypothesis over the model parameter space also leads to an expectation over the values of these hidden variables for the training data points. Since the values for the hidden variables that we will consider in practice frequently arise from examining the training data, we need to make sure that our estimators are not misled into preferring hypotheses over the values of the latent variables which overly fit the training material. For example, if we assume a generative model where the derivation of each training data point is (partially) hidden, every parameter setting disambiguates between all the different derivations of a data point by establishing a distribution over them. For such a model extracted using the training data, the danger then lies at erroneously preferring derivations which overfit the training data points, such as derivations which generate the data by combining large data fragments instead of smaller, more reusable ones.

These are issues that apply in various extents to the training of most models of complex data. For example, the large majority of models in NLP, such as probabilistic CFGs, language models etc, are extracted from training data which are also used to estimate their parameters. The actual extent that the pitfalls discussed above affects empirical work relates to the level that the model extracted from the training data abstracts from them. As we move from coarse-grained models to more fine-grained models (e.g. as we increase the maximum history size of an interpolated Markovian LM), the risk of overfitting the training data increases, with Fragment Models standing at the far end of this continuum.

FMs are based on analyses of the data of arbitrary granularity, and for these models estimation by fitting the training data leads to degenerate estimates which fail to generalise. This is not surprising, given that an extracted FM includes hypotheses over the data distribution which closely predict the training corpus, as we discussed in sections 3.1.5 and 3.1.6. When these degenerate hypotheses are then tested against the same data they were extracted from, they easily emerge as the strongest (best fitting) ones. For many modelling frameworks, such as the Fragment Models family or interpolations of models of different granularities, straightforwardly maximising the likelihood of the training data as an estimation criterion fails to arrive at model parameter values which generalise well.

3.2.2 Cross-Validated MLE

The issues highlighted above render the application of the training data Maximum Likelihood optimisation objective problematic for models (or the relevant part of the model parameters) whose estimation needs to also establish how closely they should predict the training data. Nevertheless, we need not abandon ML estimation altogether, as it is a well-understood, widely applied estimator enjoying desirable statistical properties. In contrary, we will formulate here an alternative, Cross-Validated Maximum-Likelihood estimation objective, which avoids the problems arising from establishing model hypotheses from the training data.

Deleted Estimation Some of the issues and core principles behind the solution we propose here are long established in NLP research, as exemplified by the literature on LM parameter estimation. Jelinek and Mercer (1985) observe that the n-gram relative frequencies of the training data can diverge from those in test data, especially as the length of the n-grams increases. Furthermore, the MLE estimate of the interpolation weights for a linear interpolation of Markovian language models of different orders does not generalise well (Jelinek and Mercer, 1980).

In both cases, validation or Cross-Validation, with the application of the latter often referred to as 'Deleted Estimation' in LM literature, are successfully employed to address the problems related to the skewed n-gram distributions suggested by the training data. Jelinek and Mercer (1985) cross-validate the n-gram LM, extracting in each CV-round the model from one part of the data while estimating its parameters from the held-out part. Additionally, the interpolation weights of interpolated LMs are trained by maximising the likelihood of a held-out corpus or by means of cross-validation, so as to avoid training both the n-gram models *and* the interpolation weights from the same part of the corpus. Below, we base ourselves on these approaches to formulate a comprehensive CV-MLE optimisation criterion.

Re-examining MLE on Incomplete Data Let us now examine how we can formulate a Cross-Validated Maximum Likelihood Estimation objective for generative models estimated from an incomplete data corpus $\mathcal{X}$ made of observed data points $\mathbf{x}$. As a first step however, it is interesting to begin by revisiting MLE in this setting (see also section 2.6).

This time however, we make explicit the use of training data to establishing the hypotheses $Z(\mathbf{x})$ over how each observed data point $\mathbf{x}$ can be completed with its unobserved part $\mathbf{y}$ to arrive at a complete data point $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$. Towards this aim, as part of the input of the function Z which maps incomplete to complete data, we will include the training data $\mathcal{X}$ which are implicitly employed by this function, using the notation $Z(\mathbf{x}; \mathcal{X})$. We can then rewrite equation 2.32, this time highlighting the role of the training corpus itself as part of the process to arrive at the hypotheses over the missing information $\mathbf{y}$.

$$\hat{\theta} = \arg\max_{\theta} \prod_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{z}=\langle \mathbf{x},\mathbf{y} \rangle \in Z(\mathbf{x};\mathcal{X})} p(\mathbf{x},\mathbf{y};\theta) \tag{3.8}$$

For example, in the case of Phrase-Based SMT, the incomplete data corpus $\mathcal{X}$ refers to the parallel training data, a corpus of observed word-aligned sentence-pairs $\mathbf{x}$, where each $\mathbf{x}$ misses the hidden phrase-pair segmentation $\mathbf{y}$. To train the PBSMT model, which assumes a segmentation in phrase-pairs, we must establish hypotheses $Z(\mathbf{x};\mathcal{X})$ over the phrase-segmentation, by considering which phrases can be translations of each other. Crucially, these hypotheses are constructed by examining during the model extraction step the training corpus $\mathcal{X}$ in its entirety and extracting all phrase-pairs according to the phrase-extraction heuristic, as presented in section 2.3. As discussed in section 3.2.1 above, for some model frameworks and in particular for fragment models such as those employed in PBSMT, this entails the danger of favouring hypotheses suggested by the training data itself, leading to estimates which overfit the training corpus.

Cross-Validated Likelihood To avoid this pitfall, we will employ K-fold Cross-Validation during the process of establishing the hypotheses $Z(\mathbf{x};\mathcal{X})$ on the complete data $\mathbf{z} = \langle \mathbf{x},\mathbf{y} \rangle$ from which the incomplete observations $\mathbf{x}$ might stem. In more detail, we begin by splitting the training corpus $\mathcal{X}$ in K roughly equal-sized parts $\mathcal{X}^1 \dots \mathcal{X}^K$. For every $1 \leq k \leq K$, we consider for the data points $\mathbf{x}$ belonging to part $\mathcal{X}^k$ only hypotheses $Z(\mathbf{x};\mathcal{X}^{-k})$ over the completion of the observed data which stem from the rest of the data $\mathcal{X}^{-k} = \{\mathcal{X}_1 \dots \mathcal{X}^{k-1}, \mathcal{X}^{k+1} \dots \mathcal{X}^K\}$. For the example of Phrase-Based SMT, this would translate into considering for every $\mathbf{x} \in \mathcal{X}^k$ only phrase-pair segmentations $Z(\mathbf{x};\mathcal{X}^{-k})$ which employ phrase-pairs extracted from the rest of the training corpus $\mathcal{X}^{-k}$, *excluding* the part where the data point which we currently examine belongs. We will refer to the likelihood of the incomplete corpus according to the model when only the cross-validated hypotheses over the unobserved data are considered, as the cross-validated likelihood $\mathcal{L}^{CV}$.

$$\mathcal{L}^{CV}(\mathcal{X};K,\theta) = \prod_{k=1}^{K} \prod_{\mathbf{x} \in \mathcal{X}^k} \sum_{\mathbf{z}=\langle \mathbf{x},\mathbf{y} \rangle \in Z(\mathbf{x};\mathcal{X}^{-k})} p(\mathbf{x},\mathbf{y};\theta) \tag{3.9}$$

Cross-Validated MLE Cross-Validated MLE aims at arriving at the parameter set which maximises the likelihood of the incomplete training data just as plain MLE does. However, by maximising the cross-validated incomplete data likelihood during CV-MLE we are more selective when choosing which hypotheses over the hidden part of the data to consider, by cross-validating the set of these hypotheses as just described. The CV-MLE estimate $\hat{\theta}^{CV}$ is then computed according to the following equation.

$$\begin{aligned} \hat{\theta}^{CV} &= \arg\max_{\theta} \mathcal{L}^{CV}(\mathcal{X}; K, \theta) \\ \hat{\theta}^{CV} &= \arg\max_{\theta} \prod_{k=1}^{K} \prod_{\mathbf{x} \in \mathcal{X}^k} \sum_{\mathbf{z}=\langle \mathbf{x}, \mathbf{y} \rangle \in Z(\mathbf{x}; \mathcal{X}^{-k})} p(\mathbf{x}, \mathbf{y}; \theta) \end{aligned} \tag{3.10}$$

Properties of CV-MLE On the one hand, CV-MLE deviates from standard applications of MLE little enough so as to retain the well-understood and desirable properties of MLE as an estimator. Firstly and most importantly, the estimation objective is still *likelihood maximisation*, albeit operating on a more constrained space of hypotheses $Z(\mathbf{x}; \mathcal{X}^{-k})$ on how the complete training corpus might look like. When working with incomplete data, choosing which such hypotheses to consider is a modelling choice and after this choice is made, CV-MLE proceeds as plain MLE would.

Moreover, let us assume that the incomplete to complete data mapping function $Z(\mathbf{x}; \mathcal{X})$ utilises only the set of training points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathcal{X}$ and not their frequencies in $\mathcal{X}$. This assumption holds for all models discussed in later chapters of this thesis and makes sense as the mapping function Z must output only which hypotheses must be considered and not how probable these are, the latter task assigned to the estimator.

For these mapping functions Z, CV-MLE likewise with plain MLE can be shown to be *asymptotically consistent*, following similar steps as the related proof in (Zollmann and Sima'an, 2006) for the asymptotic consistency of the DOP* estimator which also employs Cross-Validation. Intuitively, this property holds because, as the size of the training corpus grows, the probability that every training point in $\mathcal{X}^k$ is also in $\mathcal{X}^{-k}$, rendering equal the outputs of $Z(\mathbf{x}; \mathcal{X})$ and $Z(\mathbf{x}; \mathcal{X}^{-k})$, increases towards one. In that case the estimate of CV-MLE in equation (3.10) converges towards the consistent estimate of plain MLE of equation (3.8).

On the other hand, the crucial step of cross-validating the hypotheses over the unobserved part of the data *avoids overfitting* towards hypotheses which do not generalise well. On the contrary, CV-MLE favours estimates that prefer such hypotheses which, according to the cross-validation criterion, are similar to those that can be employed to model yet unseen data. These properties promote CV-MLE as a well-understood estimator with good statistical properties that directly aims towards estimates which generalise well.

3.2.3 Cross-Validated EM

Similarly to the estimate of equation (3.8) for plain MLE using incomplete data, we are in most cases not able to analytically compute the CV-MLE estimate

$\hat{\theta}^{CV}$ of equation (3.10). For this, in this work we formulate the Cross-Validated Expectation Maximization (CV-EM) algorithm, which iteratively maximises the cross-validated likelihood of the incomplete data until convergence towards a local optimum. CV-EM is a true instance of the EM algorithm, fully enjoying the same algorithmic and statistical estimation properties as those presented in section 2.6. In a nutshell, we could say that CV-EM is for CV-MLE the equivalent of EM for MLE: an iterative algorithmic optimisation framework with a well-understood operation and favourable properties.

The CV-EM algorithm as an instance of the EM algorithm follows the same algorithmic workflow of initialisation, followed by iterations between an E-step and an M-step until convergence. The crucial difference with a standard application of EM is that, as we are now climbing the cross-validated likelihood of the incomplete training data as defined above; we will only consider the cross-validated set of complete data hypotheses for every training point.

In the description of the algorithm below, we will follow the same notation as we used already for the CV-MLE in section 3.2.2. Namely, to employ K-fold cross validation during CV-EM, we begin by splitting again the training corpus in K equal sized parts $\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^K$. An essential part of an application of the EM algorithm is establishing the ambiguous complete data hypotheses by employing the incomplete to complete data mapping function Z. Since CV-EM optimises the parameters according to the CV-MLE estimation criterion, we will employ for every data point $\mathbf{x} \in \mathcal{X}^k$, the cross-validating mapping function $Z(\mathbf{x}; \mathcal{X}^{-k})$, returning complete data hypotheses from the rest of the training data after $\mathcal{X}^k$ has been excluded.

Along the same lines as for standard EM in section 2.6, the iterative procedure of CV-EM is as follows.

Initialisation We begin by initialising the model's parameter set by an initial setting $\hat{\theta}_0^{CV}$. As for all instances of the EM algorithm, initialisation can sometimes crucially determine the outcome of CV-EM's output, given that the latter climbs towards a local optimum of the cross-validated likelihood starting from the initialisation point. In cases where the shape of the CV-likelihood function in respect to the model parameters is complex, random restarts might provide a solution to the sensitivity of CV-EM to the initial parameter set.

After initialisation, the algorithm proceeds to iteratively compute estimates which raise the CV-likelihood of (3.9) (or equivalently its logarithm) until convergence. Every iteration r entails two steps, the E-step and the M-step.

E-step In the Expectation step (E-step), we formulate the *expected cross-validated* log-likelihood $Q^{CV}(\theta|\hat{\theta}_{r-1}^{CV})$ of the incomplete corpus $\mathcal{X}$ given the parameter estimate from the previous iteration $\hat{\theta}_{r-1}^{CV}$, by marginalising out the cross-validated set of complete data hypotheses $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$ provided by $Z(\mathbf{x}; \mathcal{X}^{-k})$.

$$\begin{aligned} Q^{CV}(\theta|\hat{\theta}^{CV}_{r-1}) &= E\left[\log \mathcal{L}^{CV}(\mathcal{X}; K, \theta)|\mathcal{X}, K, \hat{\theta}^{CV}_{r-1}\right] \\ &= \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{X}^k} \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in Z(\mathbf{x}; \mathcal{X}^{-k})} \log\{p(\mathbf{x}, \mathbf{y}|\theta)\} \; p(\mathbf{y}|\mathbf{x}, \hat{\theta}^{CV}_{r-1}) \end{aligned} \tag{3.11}$$

If from a mathematical point of view the E-step involves formulating the expectation over the CV log-likelihood in (3.11), from an implementation one, as for standard EM, it relates to computing the expected counts q^{CV} of the cross-validated complete data hypotheses given $\hat{\theta}^{CV}_{r-1}$.

$$q^{CV}(\mathbf{x}, \mathbf{y}|\hat{\theta}^{CV}_{r-1}) = \frac{p(\mathbf{y}|\mathbf{x}, \hat{\theta}^{CV}_{r-1})}{\sum_{\langle \mathbf{x}, \mathbf{y}' \rangle \in Z(\mathbf{x}; \mathcal{X}^{-k})} p(\mathbf{y}'|\mathbf{x}, \hat{\theta}^{CV}_{r-1})} \tag{3.12}$$

The expected counts q^{CV} disambiguate between the complete data expansions of each incomplete training point employing the current parameter estimate $\hat{\theta}^{CV}_{r-1}$ and their computation prepares the ground for the M-step that follows.

Maximization Step In the Maximization step (M-step) of the CV-EM algorithm, we maximise the objective function $Q^{CV}(\theta|\hat{\theta}^{CV}_{r-1})$ of (3.11) with respect to θ to retrieve the next parameter estimate $\hat{\theta}^{CV}_{r}$.

$$\begin{aligned} \hat{\theta}^{CV}_{r} &= \arg\max_{\theta} Q^{CV}(\theta|\hat{\theta}^{CV}_{r-1}) \\ &= \arg\max_{\theta} \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{X}^K} \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in Z(\mathbf{x}; \mathcal{X}^{-k})} \log\{p(\mathbf{x}, \mathbf{y}|\theta)\} \; q^{CV}(\mathbf{x}, \mathbf{y}|\hat{\theta}_{r-1}) \end{aligned} \tag{3.13}$$

The optimisation step of equation (3.13) above is much easier than that of (3.10), given the expected counts q^{CV} computed during the E-step and which are kept constant during the arg max operation in (3.13). In many applications, as is the case in all of our own work using CV-EM in later chapters of this thesis, the M-step of the CV-EM framework will translate in Relative-Frequency Estimation over the disambiguated corpus of complete data hypotheses.

In any case, the new current estimate $\hat{\theta}^{CV}_{r}$ computed in (3.13) is then fed as input to the E-step of the following iteration, similarly to the the workflow of Figure 2.10.

Convergence Given that CV-EM is an instance of the EM algorithm as we discuss in more detail below, its application is paired with the guarantee that the iterative process will converge when a local optimum of the CV log-likelihood is reached. A stop condition can then terminate the algorithm when the parameters have sufficiently converged or when the increment of the CV log-likelihood is smaller than a predefined value.

CV-EM as an instance of EM A comparison of the algorithmic workflow and the equations related to the E-steps and M-steps of the CV-EM algorithm presented in this section and the EM algorithm discussed in section 2.6 easily reveals that the two are closely related. However, a crucial point is that CV-EM is not an EM-*like* algorithm, somehow reminiscent of EM because of its iterative nature. On the contrary, CV-EM is an *instance* of the EM algorithm and as such inherits all the algorithmic and statistical estimation properties of the latter.

The single but essential difference between standard MLE and CV-MLE is that the latter employs a different, cross-validated set of hypotheses over the complete data $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$ that every observed data point $\mathbf{x}$ stems from. After splitting the training data in K parts, we arrive at this new set of hypotheses by making sure that the mapping function $Z(\mathbf{x}; \mathcal{X}^{-k})$ between incomplete and complete data excludes the training data part $\mathcal{X}^k$ for every $\mathbf{x} \in \mathcal{X}^k$ to compute its output, but rather relies on the rest of the training data $\mathcal{X}^{-k}$.

Similarly, the single but again essential difference between CV-EM and EM is that the latter employs a mapping function $Z(\mathbf{x}; \mathcal{X})$ examining all the training data to output complete data hypotheses, while CV-EM replaces this with the cross-validating function $Z(\mathbf{x}; \mathcal{X}^{-k})$. Employing $Z(\mathbf{x}; \mathcal{X}^{-k})$ as a mapping function in place of $Z(\mathbf{x}; \mathcal{X})$, leads from the formulation of the EM algorithm in section 2.6 to that of CV-EM here.

Crucially, the mapping function Z is not part of the EM algorithm's internals but part of its *input*. For this reason, CV-EM is a *true* EM-instance, where we alter the EM algorithm's input so as to maximise the CV-likelihood of (3.9). This is far from an observation of a purely theoretical nature. On the contrary, it guarantees the Machine Learning practitioner that CV-EM inherits the highly desirable properties of the EM algorithm as set out in section 2.6. In our case, this translates to:

Guarantee to Non-Decrease Cross-Validated Likelihood After every iteration, the new estimate raises or leaves equal the *Cross-Validated* likelihood of the incomplete-data training corpus in comparison with the estimate of the previous iteration, i.e. $\mathcal{L}^{CV}(\mathcal{X}; K, \hat{\theta}_r^{CV}) \geq \mathcal{L}^{CV}(\mathcal{X}; K, \hat{\theta}_{r-1}^{CV})$.

Guarantee to Converge The iterative process will converge to a local maximum of the Cross-Validated likelihood function $\mathcal{L}^{CV}(\mathcal{X}; K, \theta)$.

Overall, these features promote CV-EM as an algorithm with both a clear objective and a well-understood operation. The objective of CV-EM is to discover parameter estimates for generative models with latent variables, which maximise the likelihood of an incomplete data corpus when the set of complete data hypotheses for it is cross-validated. During the iterative operation of CV-EM a series of such estimates is output, each increasing this cross-validated likelihood until a guaranteed convergence towards a local optimum is reached. The combination of Maximum Likelihood Estimation and a Cross-Validated space of complete data hypotheses, as practically implemented in terms of the CV-EM algorithm, aims towards strong parameter estimates which generalise well, something that we empirically validate successfully in the following chapters of this thesis.

3.2.4 Related Approaches

Cross-Validation Based As we have already discussed in section 3.2.2, we trace the origins of our work in the applications of Deleted Estimation for estimating Language Model parameters. Jelinek and Mercer (1985) employ CV in the process of estimating the parameters of an LM under Maximum Likelihood Estimation. However their model has no hidden variables and the use of CV is confined in identifying n-grams and estimating their conditional probabilities from different parts of the data. After the n-grams participating in the language model have been identified in one part of the training data, the LM estimate is computed analytically from the rest of the data under Relative Frequency Estimation, i.e. complete-data estimation.

Jelinek and Mercer (1980) use CV and an instance of the EM algorithm, the Baum-Welch algorithm (Baum et al., 1970), to estimate model interpolation weights while other model parameters remain constant. Here we describe in general terms both a CV-MLE estimation objective and a Cross-Validated instance of the EM algorithm aiming to estimate all parameters of a model with latent variables. This is of particular importance for cases like the Fragment Models, which do not employ a distinct set of parameters which regulate the balance between fitting the training data and generalising over yet unseen data.

The possibility to employ cross-validation in an iterative estimation procedure has also been explored in (Shinozaki and Ostendorf, 2008). The authors propose an EM-*like* iterative procedure which keeps a separate model estimate for each of the K parts of the training data $\mathcal{X}^k$, which is specifically estimated from and applied on different parts of the training data. The end result is a heuristic estimation algorithm which, while somehow inspired by the workflow of the Expectation Maximization algorithm, is not an instance of the latter and does not inherit its properties. The authors acknowledge this and observe that the training data likelihood can decrease after some iterations and that there is no guarantee for convergence. In contrast, the Cross-Validated EM presented in this chapter is based on a clear learning objective and enjoys desirable properties inherited from EM by being an instance of the latter.

Information Theoretical Apart from solutions employing Cross-Validation, the problem of avoiding overfitting the training data has been widely addressed in the context of model selection: selecting among a host of models the one which, taking the training data that we have available in consideration, is most likely to generalise well. Of particular interest in the context of this thesis are model selection approaches which examine models belonging in the same model family which differ in their complexity (e.g. as measured by the number of parameters), which usually intuitively translates to how fine or coarse grained is the view that they take on data.

Examples of such approaches are the Akaike Information Criterion (AIC) (Akaike, 1974) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). Both model selection criteria are derived starting from information theoretical arguments and penalise models with a large number of parameters. Typically, increasing the number of free parameters in a model leads to estimates which further increase the likelihood of the training data. When selecting a model, both AIC and BIC add a penalising term for the number of parameters in the model in addition to the likelihood that their estimates assign to the training data, aiming to mitigate the danger of overfitting them.

A further model selection criterion that also aims to balance how closely a model fits the training data with the model's complexity is the Minimum Description Length (MDL) (Rissanen, 1978; Rissanen, 1983). MDL evaluates the description length of a model as an indication of its complexity together with the length of the description it assigns to the training data, which relates to how closely it predicts them, and it selects the model which minimises the sum of the two. Model selection criteria such as these have found applications in NLP such as (Grünwald, 1996; Goldsmith, 2001; Adriaans and Jacobs, 2006; Poon et al., 2009), either by being directly applied or by inspiring approaches which try to counterbalance the increasingly better fit of the training data offered by large models, aiming towards larger generalisation capacity. Nevertheless, the application of such criteria is often challenging (Adriaans and Vitanyi, 2007), and the difficulties mount up when applying MDL learning for the frequently complex models employed in NLP.

Regularisation A wider framework that, from a technical and a formalisation perspective, encompasses the AIC and BIC criteria as well some applications of the MLD principle (e.g. 'crude', two-part MDL (Grünwald, 2007)), is regularisation. The regularisation of the log-likelihood optimisation objective involves adding a penalty term $R(\theta)$, whose role is to penalise models of increased complexity or model instances which overfit the training data. The impact of the penalty term in the optimisation criterion is weighted by a parameter α, which allows us to adjust the tradeoff between data fit and model complexity.

$$\hat{\theta} = \arg\max_{\theta} \log \mathcal{L}(\mathcal{X}; \theta) - \alpha R(\theta) \tag{3.14}$$

The frequently applied L_p-norm regularisation involves using the L_p-norms $||\theta||_p$ of the parameter vector θ in the regularisation term $R(\theta)$.

$$||\theta||_p = \left(\sum_i |\theta_i|^p \right)^{\frac{1}{p}} \tag{3.15}$$

A member of this family that has been widely applied in Machine Learning literature is L_1-norm regularisation, which uses a penalty term equal to the sum of the parameter values. This has been shown to prefer in practice model instances with many parameters equal to zero (Tibshirani, 1996); when this happens, an L_1-norm optimisation objective essentially performs model selection and model smoothing at the same time. The AIC and BIC criteria can be interpreted as instances of L_0-"norm" regularisation; $||\theta||_0$, as p approaches zero and under certain simplifying assumptions, merely counts the number of non-zero parameters in vector θ.

Bayesian Instead of penalising whole model spaces, as the model selection criteria do based on their complexity as measured by their number of parameters or description length, the use of Bayesian inference allows us to employ external preferences over the structure and the parameter space of a model. These are stated in terms of probabilistic priors which allow preferring certain model structures, or parts of the parameter space for which the modeller hopes that estimates situated there generalise better. Frequently used choices for such priors for models employing multinomial distributions, such as is commonly the case in NLP, are the Dirichlet distribution (Johnson et al., 2007; Zhang et al., 2008a) and the Dirichlet Process (Liang et al., 2007; DeNero et al., 2008; Blunsom et al., 2009), which can both be tuned with the help of hyperparameters to prefer more compact model estimates, as discussed for the case of Fragment Models in section 3.1.8. Inference with Bayesian priors such as these typically involves marginalising out the model parameters given the training data.

While Bayesian inference provides an interesting theoretical framework to employ external biases to arrive at models and model estimates which generalising better, its practical application does not come without shortcomings. Bayesian methods are centred around the modelling step of introducing external knowledge, which while it can prove beneficial in many cases, still entails the dangers of arriving at suboptimal solutions and overwhelming the empirical evidence under the strength of the external prior imposed. Furthermore, in practice Bayesian approaches are frequently sensitive to the choices involved in approximating model parameter marginalisation using sampling or variational methods, as well as in the selection of the prior's hyperparameters.

Overall, CV-EM takes a *data-driven* approach on the problem of finding estimates which generalise well, focusing on avoiding formulating and validating hypotheses using the same data set. In contrast to CV-EM's concentration on the empirical evidence, most of the approaches mentioned above either emphasise employing external knowledge for the task, or take an information theoretic view on the problem. It is interesting to note that CV-EM is not mutually exclusive with these alternative approaches. While we believe that on one hand the data-driven nature of CV-EM safeguards against arbitrary modelling choices, we still find interesting investigating the crossroads between ours and alternative approaches on the problem of model estimation for increased generalisation.

3.2.5 CV-EM for Fragment Models

In the next chapters of this thesis, we will experiment with applying the CV-MLE optimisation criterion as implemented through the CV-EM algorithm for an array of Statistical Machine Translation models, all of which belong to the family of Fragment Models. Before we delve into the details of applying CV-EM for each particular problem, we close this section by discussing what we might expect from employing our methods for the estimation of Fragment Models in more abstract terms.

Balancing MLE Bias and Variance In section 3.1.6, we showed that the large expected error of the MLE estimator for Fragment Models can be attributed to an unbalanced correspondence between errors attributed to estimator bias and variance. The generalisation error due to estimator bias is zero, or very low in case we constrain the size of the fragments, which however gives rise to a large error due to the variance of the estimates in respect to the training data. CV-EM reduces the overall expected test error by increasing the estimator's bias in a targeted manner.

For Fragment Models, hypotheses over the value of the unobserved data variables relate to the segmentation of training instances in data fragments and the generative steps that are followed to arrive at the observed data points. CV-EM cross-validates this hypothesis space, so that all the hypotheses that will be considered employ both reusable fragments and reusable derivational steps according to the cross-validation criterion. This brings about an increase in estimator bias error due to moving away probability mass from the largest fragments, most of which will fail to survive the application of CV.

However, in contrast to arbitrary constraints such as fragment length cut-off points, increasing the generalisation error's bias term in this way directly aims to greatly reduce the error due to estimate variance. The CV-MLE estimates, focusing on reusable fragments and derivation steps, will differ with each other much less than the MLE estimates, each of which only predicts each different sampled training data set. The end result is a better trade-off between the expected test error's bias and variance terms[3], which lowers the overall generalisation error of CV-EM Fragment Model estimates.

[3] See section 2.7.1 for more details on the trade-off between bias and variance.

Model Selection and Estimation A further interesting facet of CV-EM estimation that is relevant to FM estimation, is that it combines features of both model selection and model estimation. This property of CV-EM is highly applicable to Fragment Model estimation, where parameters related to the model's level of abstraction from the training data are not clearly separated from the rest of the model parameters, as is for example the case for mixture models. Easy solutions such as separating FMs into an array of models which each employs fragments of a certain size are not working either, as some larger fragments capture data regularities while others simply overfit training data particularities. This makes it difficult for FMs to precede estimation with a clearly separated model selection step. CV-EM addresses it by applying features of model selection during estimation of the model's parameters.

On the one hand, the cross-validated complete data hypothesis space which CV-EM considers is a subset of that employed in standard applications of EM. Assuming this hypothesis space when maximising training data likelihood as CV-EM does, effectively shapes the model set that will be considered, by not considering models leading to hypotheses which do not survive cross-validation. While no single model is selected, models employing fragments which do not appear to generalise well according to CV are either eliminated or penalised, depending on the smoothing choices made in each application of CV-EM. On the other hand, when the extent of the model space and the preferences over it have been set, estimating model parameters by maximising training data likelihood allows us to discover which estimate seems to better capture the latent patterns of the training data, cross-validating our hypotheses over them to safeguard against overfitting.

In total, these features of CV-EM together with the algorithmic and statistical estimation properties inherited by the Expectation-Maximisation algorithm promote it as a well-founded and highly suitable estimation framework for Fragment Models. In the next chapters of this thesis, we empirically evaluate CV-EM for Fragment Model estimation, for three Statistical Machine Translation models of increasing sophistication belonging to this family.

Chapter 4

Phrase Translation Probabilities Estimation

The introduction of phrase-based Statistical Machine Translation (SMT) took advantage of the foundations laid by the work on word-based translation models to bringing about a forward leap for SMT. The registered improvement was both in terms of translation performance as well as of its acceptance by the research and business communities. Importantly, moving from word-based towards phrase-based translation marked the transition of SMT into the realm of the Fragment Models (FMs) family, to which Phrase-Based SMT (PBSMT) models belong. The fragments in the case of PBSMT correspond to contiguous phrase-pairs. These are considered as our translation units and modelling the correspondence between their phrases allows SMT to tap into the modelling potential of FMs. This allows PBSMT models to combine the generalisation capacity of word-based translation models, with the ability to forego the independence assumptions behind them when translating certain phrases.

However, translating with phrasal fragments also exposed SMT to the estimation problems faced by FMs. Direct application of trusted and well-understood approaches such as Maximum Likelihood Estimation (MLE) and the Expectation-Maximization (EM) algorithm is almost useless for PBSMT, turning researchers into employing heuristic estimators instead. While these estimators do perform relatively well, their heuristic nature leaves open the question of whether alternative estimators can deliver competitive performance from premises that are better understood.

In this chapter, we will apply the CV-EM algorithm for the estimation of the conditional phrase translation probabilities that form the core of PBSMT models, based on work first presented in (Mylonakis and Sima'an, 2008). We do this by employing CV-EM to estimate the parameters of a phrase-based translation model. Our model directly addresses the latent segmentation of sentence-pairs in phrase-pairs, which the heuristic estimators fail to take into account. It considers a restricted *binary* segmentation space with a prior over the segmentation variable, based on linguistic as well as computational premises.

In the context of this thesis, this empirical investigation serves two aims: (a) to propose an estimation algorithm for phrase-based models with both a clear learning objective, in the form of CV-MLE as well as a well-understood implementation like CV-EM and (b) to empirically evaluate applying the CV-EM algorithm to estimate the parameters of a state-of-the-art Fragment Model.

4.1 Problem Setting

The Phrase-Based SMT modelling framework (Och et al., 1999; Koehn et al., 2003), which we introduce in section 2.3, is based on the notion of establishing *phrases* instead of words as the basic translation units. Given an input source sentence $\mathbf{f}$, the key intuitive assumption is that, after $\mathbf{f}$ has been *segmented* into K source phrases $\tilde{f}_1^K$, each source phrase $\tilde{f}_i$ is *translated* independently of the rest into a target phrase $\tilde{e}_i$. The resulting set of target phrases $\tilde{e}_1^K$ is then further *reordered* according to a reordering pattern $\boldsymbol{\pi}$ between the indexes of the source and target phrase vectors $\tilde{f}_1^K$ and $\tilde{e}_1^K$ to arrive at the target output $\mathbf{e}$, similarly to the process in Figure 2.5.

Crucially, even though the concept of phrase segmentation is central to the assumptions behind phrase-based translation, most of the Phrase-Based SMT systems fail to account for it. Instead, they are based around translation models which assign probabilities to translations of already segmented source sentences. These models are log-linear interpolations of Φ feature functions ϕ, whose feature scores are interpolated together under feature weights λ and normalised by $Z(\mathbf{f})$.

$$p(\mathbf{e}, \tilde{e}_1^K, \boldsymbol{\pi} | \mathbf{f}, \tilde{f}_1^K) = \frac{1}{Z(\mathbf{f})} \sum_{i=1}^{\Phi} \lambda_i \; \phi_i(\mathbf{e}, \mathbf{f}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi}) \tag{4.1}$$

During decoding, a PBSMT system selects the output translation $\hat{\mathbf{e}}$ through a Viterbi search on the space of all source phrase segmentations $\tilde{f}_1^K$ and subsequent constructions of target sentences from phrase translations and reordering operations.

$$\langle \hat{\mathbf{e}}, \widehat{\tilde{e}_1^K}, \widehat{\tilde{f}_1^K}, \hat{\boldsymbol{\pi}} \rangle = \operatorname*{arg\,max}_{\mathbf{e}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi}} \sum_{i=1}^{\Phi} \lambda_i \; \phi_i(\mathbf{e}, \mathbf{f}, \tilde{e}_1^K, \tilde{f}_1^K, \boldsymbol{\pi}) \tag{4.2}$$

In the model of (Koehn et al., 2003), which will form the baseline for the experiments we present later in this chapter, the feature set includes an array of features examining the correspondence between $\mathbf{f}$ and $\mathbf{e}$ from different perspectives. These features range from those that examine the translation of phrases and reordering, to others which consider target sentence well-formedness using a monolingual target language modelling feature. They are further complemented by additional smoothing features, with the overall feature set described in detail in section 2.3.

While all members of the feature set contribute in recovering $\hat{\mathbf{e}}$ in equation (4.2), the backbone of a Phrase-Based SMT model is formed by the conditional phrase translation features $\phi^{\mathbf{e}|\mathbf{f}}_{\text{PHR}}$ and $\phi^{\mathbf{f}|\mathbf{e}}_{\text{PHR}}$.

$$\phi^{\mathbf{e}|\mathbf{f}}_{\text{PHR}} = \log \prod_{k=1}^{K} p(\tilde{e}_k|\tilde{f}_k) \qquad \phi^{\mathbf{f}|\mathbf{e}}_{\text{PHR}} = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \tag{4.3}$$

These examine the correspondence between **f** and **e** under the assumption that each phrase is translated independently from the rest of the input, with each of the two features examining one of the two translation directions (**f** to **e**, as well as **e** to **f**). Their score is computed based on the conditional phrase translation distributions: $p(\tilde{e}|\tilde{f})$ for each source phrase $\tilde{f}$, and $p(\tilde{f}|\tilde{e})$ for each target phrase $\tilde{e}$. Estimating these distributions is thus an essential step of training a PBSMT model and this will be the focus of this chapter.

4.1.1 PBSMT and Fragment Models

Phrase-Based SMT models employing conditional distributions $p(\mathbf{e}, \tilde{e}_1^K, \boldsymbol{\pi}|\mathbf{f}, \tilde{f}_1^K)$, belong in the Fragment Models family along the following lines, in correspondence with how FMs were defined in section 3.1.2:

Data A target sentence **e** can be analysed given a source sentence **f** and a segmentation of the latter in source phrases $\tilde{f}_1^K$, through the reordering $\boldsymbol{\pi}$ of target phrases $\tilde{e}_1^K$. Each such target phrase $\tilde{e}_i$ emerges as the translation of the corresponding source phrase $\tilde{f}_i$.

Fragments The model employs contiguous phrase-pairs as data fragments extracted from a word-aligned parallel corpus. These are extracted following simple heuristics, the most important of which is that word alignments originating in either the source or target phrase of the pair must be contained within the phrase-pair. These fragments vary in size as measured by the length of the phrases that comprise every such pair. This fragment size can grow up to the full sentence lengths of training points, so that whole sentence-pairs are also conceived as phrase-pairs.

Derivations Every phrase-pair fragment $\langle \tilde{e}, \tilde{f} \rangle$ can be employed to supply the target phrase translation $\tilde{e}$ for a source phrase $\tilde{f} \in \tilde{f}_1^K$. The resulting target phrase vector $\tilde{e}_1^K$ is reordered according to the reordering variable $\boldsymbol{\pi}$ to derive **e**.

Model Each such derivation of **e** from $\tilde{f}_1^K$ is assigned a probability by the model of equation (4.1).

A phrase-based Fragment Model for translation memorises and recombines extracted contiguous phrase-pairs from the training data. This allows it to memorise and reuse instances of local translation phenomena like local reordering, as well as encompass correlations between the translations of adjacent words as in the case of translating idioms. However, these models are now also exposed to the same overfitting issues plaguing the estimation of all Fragment Models according to the frequently used MLE estimation objective. Attempts to train the conditional phrase translation probabilities $p(\tilde{e}|\tilde{f})$ as part of a generative translation model and estimating their parameters so as to maximise training data likelihood, lead to degenerate estimates which perform poorly (DeNero et al., 2006).

For this reason, right from the initial introduction of PBSMT, model parameters are set to heuristic estimates, a practice still used in state-of-the-art systems up to this day.

4.1.2 Heuristic Estimation

The heuristic estimates for the conditional phrase translation probabilities $p(\tilde{e}|\tilde{f})$ and $p(\tilde{f}|\tilde{e})$ used in the feature functions of (4.3) are set to values based on the counts $C(\langle\tilde{e},\tilde{f}\rangle)$, which register how many times each phrase-pair can be extracted from the training data.

$$p(\tilde{e}|\tilde{f}) = \frac{C(\langle\tilde{e},\tilde{f}\rangle)}{\sum_{\tilde{e}'} C(\langle\tilde{e}',\tilde{f}\rangle)} \qquad\qquad p(\tilde{f}|\tilde{e}) = \frac{C(\langle\tilde{e},\tilde{f}\rangle)}{\sum_{\tilde{f}'} C(\langle\tilde{e},\tilde{f}'\rangle)} \tag{4.4}$$

This heuristic solution to the estimation problem of PBSMT models is reminiscent of the DOP-1 estimator, the first estimator proposed for Data Oriented Parsing (DOP) models, which also belong in the family of Fragment Models (see section 3.1.2). Similarly to the PBSMT heuristic, under DOP-1, tree fragments are assigned probabilities in proportion to their extraction counts from the training corpus. However, this heuristic choice leads to estimates which overfit towards large tree fragments, leading to both weak statistical properties and weaker performance in relation to later proposed estimators.

Examining the relation of the estimates in equations (4.4) to the training corpus reveals their heuristic nature. While these estimates are sometimes informally referred to as relative frequencies, it is important to understand that they are totally unrelated to the frequency of events in the training data (word-aligned sentence-pairs). Indeed, the segmentation of sentence-pairs into phrase-pairs is not observed in the training corpus, as the latter is composed of incomplete-data in this respect. Instead, the heuristic estimates are relative frequencies of phrase translations in the multiset of extracted phrase-pairs, which is related to the training corpus only by means of the arbitrary extraction step. As a consequence of this, the heuristic estimates are not known to optimise any meaningful function of the training parallel corpus itself.

Despite these shortcomings of the heuristic estimator, the mounting number of efforts attacking the problem of PBSMT model estimation over the last few years (DeNero et al., 2006; Marcu and Wong, 2002; Birch et al., 2006; Moore and Quirk, 2007; Zhang et al., 2008a) exhibits its difficulty. So far, none has lead to an alternative method that performs as well as the heuristic on reasonably sized data.

4.1.3 Motivating an Intuitive Estimation Approach

In the face of the difficulty of coming up with an alternative estimation approach for PBSMT models, the heuristic estimates have long been dominant in state-of-the-art implementations of phrase-based translation systems. Still, there are multiple arguments motivating research in that direction.

Firstly, estimators which employ a clearer optimisation objective allow us to better understand how the estimates relate to the training data, which is by no means an argument of a merely theoretical nature. On the contrary, employing a well-founded estimator makes it easier to evaluate its statistical properties and pinpoint the source of its errors, or the conditions under which it performs well.

Further, estimating phrase-based models, as is the case with Fragment Models in general, allows us to fine-tune how closely the training corpus will be fit during estimation, something which will reflect in the generalisation capacity of the estimates. On one hand, smaller phrase-pairs provide in general higher coverage, but are difficult to combine together due to the strong independence assumptions of the models. On the other, larger phrase-pairs many times offer a relatively trusted translation for a large span of the input source sentence, but at the cost of low coverage and less flexibility in adapting their fixed translation to the surrounding context. While this results in a parameter space that can prove treacherous for estimators, it also makes PBSMT model estimation highly interesting by providing the chance to *learn* how to combine memorisation with re-use to perform well on novel source sentences.

Finally, employing alternative well-founded estimators for phrase-based models, apart from the chance to perhaps offer equivalent or better translation performance from better understood foundations, even more importantly lays the path to the future. Heuristics, being arbitrary in nature, can hardly be evolved to something more meaningful. Also, as they are ad hoc solutions with an applicability based only on empirical grounds. The implications of extending their use to novel models, or for translation between language pairs enjoying different properties, are far from being clear. Proposing estimators based on stronger principles builds both a better trusted estimation platform for novel and refined models, as well as allows the estimator itself to function as a starting point for further research.

4.2 Related Work

Marcu and Wong (2002) realise that the problem of extracting phrase pairs should be intertwined with the method of probability estimation. They formulate a joint phrase-based model in which a source-target sentence pair is generated jointly. However, the huge number of possible *phrase-alignments* prohibits scaling up the estimation by Expectation-Maximization to large corpora. Birch et al. (2006) provide soft measures for including word-alignments in the estimation process and obtain improved results, but only on small data sets.

More recently, (Blunsom et al., 2008a) attempt a related estimation problem to (Marcu and Wong, 2002), using the expanded phrase pair set of (Chiang, 2005a), working with an exponential model and concentrating on marginalising out the latent segmentation variable. In addition, (Zhang et al., 2008a) report on a multi-stage model, *without* a latent segmentation variable, but with a strong prior preferring sparse estimates. This prior is embedded in a Variational Bayes (VB) estimator and the authors concentrate their efforts on pruning both the space of phrase pairs and the space of (ITG) analyses. Blunsom et al. (2008a) and (Zhang et al., 2008a) report improved performance, albeit again on a limited training set (approx. 140K-170K sentences with sentence length constraints).

DeNero et al (DeNero et al., 2006) have explored estimation using EM of phrase pair probabilities under a conditional translation model based on the original source-channel formulation. This model involves a hidden segmentation variable that is set uniformly (or to prefer shorter phrases over longer ones). Furthermore, the model involves a reordering component akin to the one used in IBM Model-3. Despite this, the heuristic estimator remains superior because “EM learns overly determinized segmentations and translation parameters, overfitting the training data and failing to generalize”. Moore and Quirk (2007) devise an estimator working with a model that does not include a hidden segmentation variable but works with a heuristic iterative procedure (rather than MLE or EM). The translation results remain inferior to the heuristic, but the authors note an interesting trade-off between decoding speed and the various settings of this estimator.

Our work expands on the general approach taken by (DeNero et al., 2006; Moore and Quirk, 2007) but arrives at insights concerning the value of binary phrase-pair segmentations similar to those of (Zhang et al., 2006), albeit in a completely different manner. The present work differs from all preceding work in that it employs the set of *all phrase pairs* during training. It differs from (Zhang et al., 2008a) in that, while it does postulate a latent segmentation variable as well, it puts the prior directly over that variable rather than over the ITG synchronous rule estimates. Our method neither excludes phrase pairs before estimation nor does it prune the space of possible segmentations/analyses during training. As well as smoothing, we find (in the same vein as (Zhang et al., 2008a)) that setting effective priors and smoothing is crucial for EM to arrive at better estimates.

4.3 Our approach

In this chapter, we start out from a standard phrase extraction procedure based on word-alignment and aim solely at estimating the conditional probabilities for the phrase pairs in both translation directions. Unlike preceding work, we extract *all phrase pairs* from the training corpus and estimate their probabilities, i.e. without limit on length. After training, we can still limit the set of phrase pairs to those selected by a cut-off on phrase length. The reason for using all phrase pairs during training is that it gives a clear point of reference for an estimator, without implicit, accidental biases that might emerge due to length cut-off.

We employ a novel formulation of a conditional translation model that works with a *prior* over *bilingual* segmentations and a bag of conditional phrase pairs. We use binary Synchronous Context-Free Grammar (bSCFG) based on the Inversion Transduction Grammar (ITG) (Wu, 1997; Chiang, 2005a), to define the set of eligible segmentations for an aligned sentence pair. We also show how the number of spurious derivations per segmentation in this bSCFG can be used for devising a prior probability over the space of segmentations, capturing the bias *in the data* towards monotone[1] translation.

At the heart of the estimation process lies an instance of the Cross-Validating EM algorithm. Apart from a direct application of the CV-EM framework for this estimation problem, we also experiment with a Jackknife inspired variation of it, which averages the temporary probability estimates of multiple parallel EM processes at each joint iteration.

We evaluate against a state-of-the-art baseline system (Moses) (Hoang and Koehn, 2008), which works with the log-linear interpolation of feature functions of equation (4.2), with interpolation weights optimised by Minimum Error Rate Training (Och, 2003). We simply substitute our own estimates for the heuristic phrase translation estimates of (4.4) and compare the two within the Moses decoder. While our estimates differ substantially from the heuristic, their performance is on par with the heuristic estimates. This is remarkable given the fact that comparable previous work (DeNero et al., 2006; Moore and Quirk, 2007) did not match the performance of the heuristic estimator using large training sets. We believe that using CV-EM for the estimation of the model's parameters is the vital choice which allows to avoid overfitting while disambiguating the phrase-pair segmentation of the word-aligned training corpus, arriving in this way at strong estimates of the conditional translation probabilities.

4.3.1 The Translation Model

Heuristically estimated PBSMT systems mostly treat the latent segmentation of training sentence-pairs into phrase-pairs as an unnecessary nuisance. In contrast,

[1] Monotone translation produces target output which follows the word or phrase order of the source sentence.

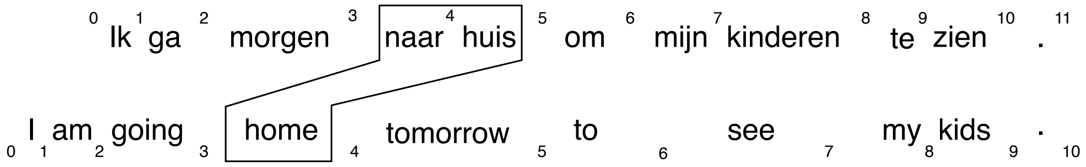

0 Ik 1 ga 2 morgen 3 naar 4 huis 5 om 6 mijn 7 kinderen 8 te 9 zien 10 . 11
0 I 1 am 2 going 3 home 4 tomorrow 5 to 6 see 7 my 8 kids 9 . 10

Figure 4.1: A bilingual segment $\sigma_k = \langle l_f, r_f, l_e, r_e \rangle$, covering source span $\langle l_f, r_f \rangle = \langle 3, 5 \rangle$ and target span $\langle l_e, r_e \rangle = \langle 3, 4 \rangle$

.

our model makes this part of the translation process explicit by incorporating a latent bilingual segmentation variable $\boldsymbol{\sigma}$. This takes values from a constrained *binary* space of segmentations, while a non-uniform prior $p(\boldsymbol{\sigma}; \mathbf{a})$ over the segmentations enforces a preference for more productive segmentation patterns.

We couple this modelling component with the conditional phrase translation probabilities, to arrive at a conditional phrase-based translation model. Estimating the parameters of this model by using CV-EM will lead us to phrase translation distribution estimates which enjoy a clear relation to the training corpus as local optima of the Cross-Validated Likelihood of the latter.

4.3.2 Generative Process

Given a word-aligned source-target sentence-pair $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle$, the generative story underlying our model goes as follows:

1. Abiding by the word-alignments in $\mathbf{a}$, segment the source-target sentence-pair $\langle \mathbf{e}, \mathbf{f} \rangle$ into a sequence of K non-overlapping containers $\boldsymbol{\sigma} = \sigma_1^K$. Each container $\sigma_k = \langle l_f, r_f, l_e, r_e \rangle$ consists of the start l_f and end r_f positions for a phrase in $\mathbf{f}$, and the start l_e and end r_e positions for an aligned phrase in $\mathbf{e}$, as in Figure 4.1.

2. For a given segmentation σ_1^K, for every container $\sigma_k = \langle l_f, r_f, l_e, r_e \rangle$ with $1 \leq k \leq K$, select a phrase $\tilde{e}_k$ for the output span $\langle l_e, r_e \rangle$ as a translation of the source phrase $\tilde{f}_k$ corresponding to the span $\langle l_f, r_f \rangle$, independently from the rest of the source input and according to the conditional phrase translation distribution $p(\tilde{e}|\tilde{f}_j)$.

This conditional translation process is depicted in Figure 4.2 and leads to the following probabilistic model:

$$p(\mathbf{e}|\mathbf{f}; \mathbf{a}) = \sum_{\sigma_1^K \in \boldsymbol{\Sigma}(\mathbf{a})} p(\sigma_1^K; \mathbf{a}) \prod_{\sigma_k \in \sigma_1^K} p(\tilde{e}_k | \tilde{f}_k) \tag{4.5}$$

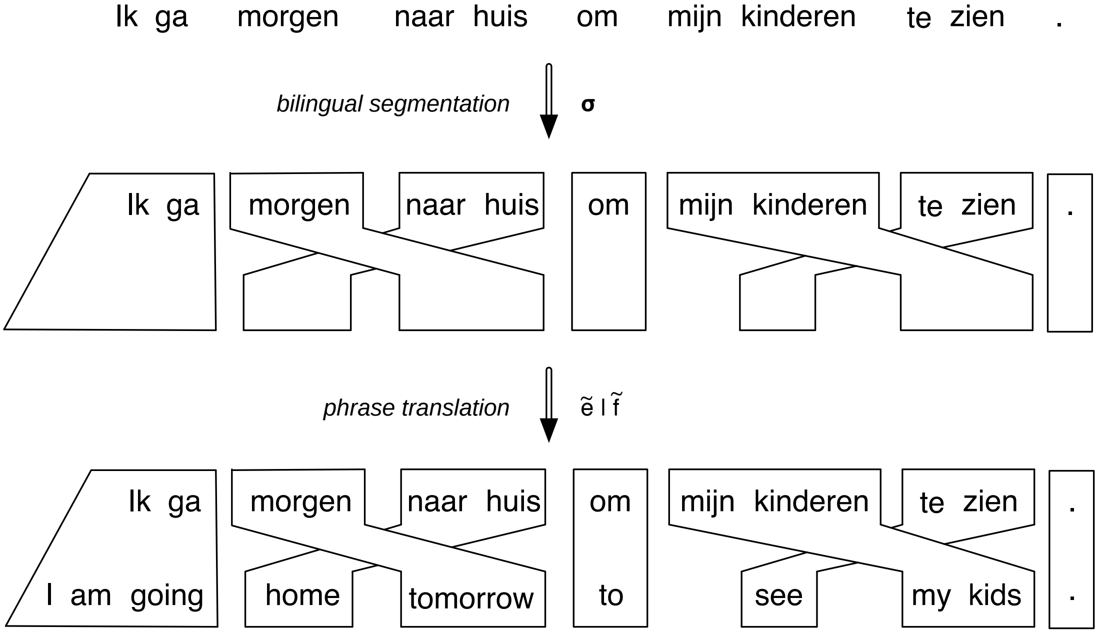


Figure 4.2: The conditional translation process for the model of equation (4.5).

In equation (4.5) above, $\mathbf{\Sigma}(\mathbf{a})$ is the set of *binary* segmentations (defined next) that are eligible according to the word-alignments $\mathbf{a}$ between $\mathbf{f}$ and $\mathbf{e}$; i.e. which only employ containers which delineate phrase pairs according to the usual PBSMT rules[2].

These segmentations into bilingual containers are different from the monolingual latent segmentation in phrases implied by Phrase-Based SMT models. They also differ from the segmentation variable used in earlier comparable conditional models (e.g., (DeNero et al., 2006)), which must generate the alignment on top of the segmentations. Our bilingual segmentations encompass both the process of partitioning the source sentence into phrase pairs, as well as establish the reordering pattern between the source phrases and their translations, as each container of the segmentation defines the position of the target translation in the output.

The process of formulating our model will be complete after we define the space of binary segmentations $\mathbf{\Sigma}(\mathbf{a})$ that we consider and discuss our choice of prior probability $p(\sigma_1^K)$ over these segmentations.

4.3.3 Binary Segmentations Space

Considering the unconstrained space of all segmentations $\boldsymbol{\sigma}$ between a sentence pair in our training data leads to an NP-hard problem, as it was shown for

[2] At least one alignment point between the two spans $\langle l_f, r_f\rangle$ and $\langle l_e, r_e\rangle$, no alignment points crossing the container's boundaries.

the similar Fragment Model of DOP (Sima'an, 1996). Given this, working with phrase-based models necessarily involves considering a certain subset of the segmentation space. For example, this can be done by imposing length constraints on the phrases which make up phrase-pairs, as well as by employing approximative search, such as the beam search algorithms which are frequently used in PBSMT decoder implementations.

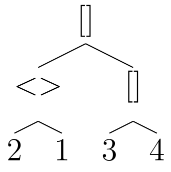
Instead of using arbitrary cut-off lengths to constrain the segmentation space, models based on binary Synchronous Context Free Grammars (bSCFGs), in the form of Inversion Transduction Grammars (ITG) (Wu, 1997), take an alternative approach which is motivated by both computational and linguistic arguments. bSCFGs occupy a part of the space of all SCFGs which is highly interesting to MT practitioners, as we discussed in section 2.4.1. Exploring the space of all derivations which can be described by a bSCFGs, can be usually performed using algorithms with polynomial computational complexities in respect to the length of the sentence-pairs. At the same time, as has been first identified by (Wu, 1997) and further confirmed empirically by (Huang et al., 2009) , bSCFGs seem to be able to cover most of the reordering patterns encountered in natural language pairs.

For these reasons, we will consider in the work in this chapter the space of *binary* segmentations. We thus denote as $\mathbf{\Sigma}(\mathbf{a})$ in equation (4.5) above the space of segmentations that can be produced from a binary phrase-based SCFG, which employs phrase-pair spans that abide by the alignments $\mathbf{a}$. This SCFG has two binary synchronous rules that correspond respectively to the contiguous monotone and inverted alignments, denoting with [ ] monotone and with $\langle\ \rangle$ swapping reordering of the target phrase translations.

$$\begin{aligned} \mathrm{XP} &\rightarrow [\mathrm{XP}\ \ \mathrm{XP}] \\ \mathrm{XP} &\rightarrow \langle \mathrm{XP}\ \ \mathrm{XP} \rangle \end{aligned} \tag{4.6}$$

These two synchronous rules are coupled in our grammar with a set of lexical, phrase-pair emitting rules $\{\mathrm{XP} \rightarrow \tilde{e}\ /\ \tilde{f} \mid \langle \tilde{e}, \tilde{f} \rangle$ is a phrase pair$\}$. In this bSCFG, every derivation corresponds to a phrase segmentation of the input and a binary re-ordering pattern for the translations of the source phrases according to the rules in (4.6).

Binarisable Reordering Patterns Following (Zhang et al., 2006; Huang et al., 2009), in phrase-based translation every sequence of alignments between K source and target phrases can be viewed as a sequence of integers $1, \dots K$ together with a permuted version of this sequence $\pi(1), \dots, \pi(I)$, where the two copies of an integer in the two sequences are assumed aligned/paired together. For example, possible permutations of $\{1, 2, 3, 4\}$ are $\{2, 1, 3, 4\}$ and $\{2, 4, 1, 3\}$. Permutations such as these can be used to describe the reordering pattern of a segmentation



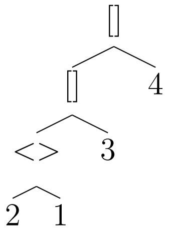


Figure 4.3: Multiple ways to binarise the reordering pattern $\{2, 1, 3, 4\}$.

σ_1^K for a sentence-pair, by indicating the order of the target phrases in relation to the source phrase that each originates from.

Binarisable reordering patterns are those which can be captured by binary SCFGs employing rules such as those in (4.6). Huang et al. (2009) describe a linear-time algorithm to find *binarisations*: derivations of reordering patterns $\pi(1), \ldots, \pi(I)$ from $1, \ldots K$, employing binary reordering steps which swap or keep intact the order of two adjacent spans in the sequence of integers, as the rules of (4.6) do. For most reordering patterns there is more than one way to binarise them. The number of possible binarisations of a binarisable permutation is a recursive function which reaches its maximum for fully monotone permutations. It is equal to the number all binary trees, which is a factorial function of the length of the permutation. Each such binarisation corresponds to a different way to derive the reordering pattern in terms of a bSCFG, with Figure 4.3 listing two such derivations for the reordering pattern $\{2, 1, 3, 4\}$.

Accordingly, for a given reordering pattern as indicated by the word alignments $\mathbf{a}$, the bSCFG of (4.6) which constraints the space of segmentations $\mathbf{\Sigma}(\mathbf{a})$ produces multiple derivations for every segmentation $\boldsymbol{\sigma} \in \mathbf{\Sigma}(\mathbf{a})$. It is possible to constrain this bSCFG such that it generates a single, canonical derivation per segmentation (Wu, 1997). However, in the next section we show that the number of such derivations is a good measure of phrase pair productivity, a feature we take advantage of to formulate a prior over the segmentation space.

Finally, while there is evidence that most of the reordering patterns for many natural language pairs are binarisable (Huang et al., 2009), there still exist non-binarisable reorderings which might be present in the training data, either as a result of less frequent translation phenomena or noise such as misaligned words. Segmentations which correspond to such reordering patterns (e.g. $\{2, 4, 1, 3\}$) are non-binarisable and are not included in $\mathbf{\Sigma}(\mathbf{a})$. Nevertheless, our Fragment Model of (4.5) takes an *all phrase-pairs* approach, including segmentations employing phrase-pairs of unconstrained lengths. This enables always finding a segmentation for a word-aligned sentence-pair in $\mathbf{\Sigma}(\mathbf{a})$, as we may encompass the

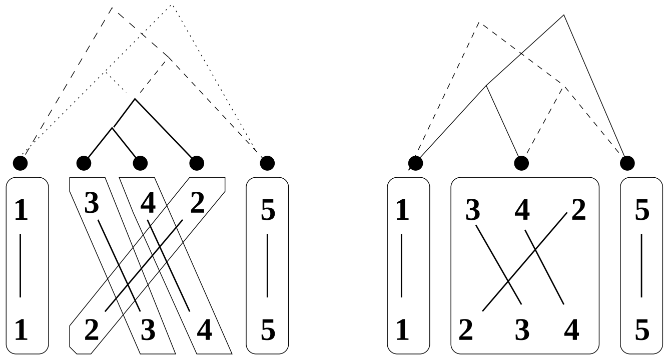


Figure 4.4: Two segmentations of an alignment/permutation. Both segmentations have the same number of binarisations despite differences in container sizes.

non-binarisable parts of the reordering pattern in longer phrase-pairs, if needed up to the full sentence-pair length.

4.3.4 Prior over Segmentations

As it has been found out by (DeNero et al., 2006), it is not easy to come up with a simple, effective prior distribution over segmentations that allows for improved phrase pair estimates. Within a Maximum-Likelihood estimator, preference for segmentations σ_1^I consisting of longer containers could lead to overfitting, as is the case for all Fragment Models. Alternatively, it is tempting to have a preference for segmentations σ_1^I that consist of shorter containers, because (generally speaking) shorter containers have higher expected coverage of new sentence pairs. However, mere bias for shorter containers will not give better estimates as observed by (DeNero et al., 2006). One case where this bias clearly fails is the case of a contiguous sequence of containers with a complex alignment structure (crossing alignments). For example as seen in Figure 4.4, for the word-alignment pattern $\{1, 3, 4, 2, 5\}$ there is a segmentation into five containers $\{1; 3; 4; 2; 5\}$, as well as a further one into three $\{1; 3, 4, 2; 5\}$. The first segmentation involves shorter containers that have crossing brackets among them, while the second one consists of three containers including a longer container $\{3, 4, 2\}$.

In the first segmentation, due to their crossing alignments, each of the containers $\{3\}$, $\{4\}$ and $\{2\}$ will not combine with the surrounding context ($\{1\}$ and $\{5\}$) on its own, i.e., without the other two containers. Furthermore, there is only a single binarisation of $\{3, 4, 2\}$. Hence, while the first segmentation involves shorter containers than the second one, these shorter containers are as *productive* as the large container $\{3, 4, 2\}$, i.e., they combine with surrounding containers in the same number of ways as the large container. In such and similar cases, there are no grounds for the bias towards shorter phrases/containers.

The notion of *container productivity* (the number of ways in which it combines with surrounding containers during training) seems to correlate with the expected number of ways a container can be used during decoding, which should be correlated with expected coverage. During training, containers that are often surrounded by other monotonically aligned containers, are expected to be more productive than alternative containers that are often surrounded by crossing alignments. Hence, the number of binarisations that a segmentation has under the bSCFG is a direct function of the ways in which the containers combine among themselves (monotone vs. swapping) within segmentations and provides a more accurate measure of container productivity than container length.

On these grounds we formulate a prior distribution over segmentations $p(\sigma_1^K; \mathbf{a})$ of a word-aligned sentence-pair as follows:

$$p(\sigma_1^K; \mathbf{a}) = \frac{N(\sigma_1^K)}{Z(\mathbf{\Sigma}(\mathbf{a}))} \tag{4.7}$$

Above, $N(\sigma_1^K)$ is the number of binary derivations that σ_1^K has in the binary SCFG (bSCFG) and $Z(\mathbf{\Sigma}(\mathbf{a})) = \sum_{\sigma_1^J \in \mathbf{\Sigma}(\mathbf{a})} N(\sigma_1^J)$. In total, this prior is the ratio of the number of bSCFG derivations of σ_1^K to the total number of derivations that $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle$ has under the bSCFG.

Hence, the final model we employ is the following:

$$p(\mathbf{e}|\mathbf{f}; \mathbf{a}) = \sum_{\sigma_1^K \in \mathbf{\Sigma}(\mathbf{a})} \frac{N(\sigma_1^K)}{Z(\mathbf{\Sigma}(\mathbf{a}))} \prod_{\sigma_k \in \sigma_1^K} p(\tilde{e}_k | \tilde{f}_k) \tag{4.8}$$

4.3.5 Contrast with Similar Models

In contrast with the model of (DeNero et al., 2006), who define the segmentations over the source sentence $\mathbf{f}$ alone, our model employs bilingual containers thereby segmenting both source and target sides simultaneously. Therefore, unlike (DeNero et al., 2006), our model does not need to generate the word-alignments explicitly, as the latter are embedded in the segmentations. Similarly, our model does not include *explicit* penalty terms for reordering/inversion but includes a related bias in the prior probabilities over segmentations $p(\sigma_1^K; \mathbf{a})$.

In a way, the segmentations and bilingual containers we use can be viewed as similar to the concepts used in the Joint Model of Marcu and Wong (Marcu and Wong, 2002). Unlike (Marcu and Wong, 2002) however, our model works with conditional probabilities and starts out from the word-alignments.

The novel aspects of our model are three: (a) it defines the set of segmentations using a bSCFG, (b) it includes a novel, refined prior probability over segmentations, and (c) it employs all phrase pairs that can be extracted from a word-aligned training parallel corpus. For these novel elements to produce reasonable estimates, we employ CV-EM for this Fragment Model for conditional phrase-based translation.

4.4 Estimation with CV-EM

The translation model of equation (4.8) coupled with the all-phrase pairs principle that we employ result in a conditional Fragment Model for translation. Directly applying MLE estimation with it is bound to overfit the training data, as we have both discussed from a theoretical perspective here in section 3.1.5, and as considered empirically in (DeNero et al., 2006). In our experiments, where we do not enforce a phrase-pair length cut-off value, plain EM strongly overfits towards considering the sentence-pair as a single large phrase-pair fragment. Other hypotheses receive fractional expected counts close to zero, merely as a result of stopping EM short of full convergence.

Avoiding the degenerate MLE estimates can take the form of: (a) employing probabilistic priors over the segmentation space which prefer more reusable fragments, or (b) smoothing the learning objective itself so that we are led to estimates which generalise better. In this work, we employ both solutions in tandem. The general-purpose, smoothing learning objective of Cross-Validated MLE is complemented by a model and application specific smoothing prior.

This prior over segmentations $p(\sigma_1^K; \mathbf{a})$, defined in section 4.3.4, counters overfitting by preferring segments which are more productive, in the sense of taking part together with their context in more derivations of the target sentence. Our formulation for this prior in (4.7) prefers shorter fragments that participate in monotone translations and is less inclined towards the preference for smaller phrase-pairs when these participate in complex reordering patterns as shown in Figure 4.4. While this prior counters overfitting up to a certain extent, in our experiments we find that it is not enough to avoid the degenerate EM estimates.

For this, we couple it with the Cross-Validated Expectation-Maximization algorithm we formulated in section 3.2. The Cross-Validated MLE estimation objective that our algorithm is based on, will help to avoid considering overfitting segmentation hypotheses which arise from single training instances and explore instead a segmentation space which favours reusable phrase-pairs. Apart from a standard application of the CV-EM framework, we further examine a variation of the algorithm which further cross-validates the parameter values themselves instead of only applying CV for the latent segmentation variable. After examining both CV-EM variations we discuss smoothing and implementation issues, preparing the grounds for our empirical evaluation.

4.4.1 Applying the CV-EM Framework

We begin the process of applying the CV-EM framework for the estimation of the parameters of the phrase-based conditional translation model in (4.8), by relating the concepts and notation of the problem to their more abstract counterparts in section 3.2.3. In relation to the estimation problem at hand, the available training data $\mathcal{X}$ are *incomplete*. These are made up of observed word-aligned sentence-pairs $\mathbf{x} = \langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle$, while the segmentation $\mathbf{y} = \boldsymbol{\sigma}$ of each sentence-pair in phrase-pairs remains unobserved.

Cross-Validated Segmentation We prepare the ground for Cross-Validating during estimation by splitting the corpus $\mathcal{X}$ in J parts $\mathcal{X}^1, \dots, \mathcal{X}^J$ of approximately equal sizes. The crucial difference between CV-EM and a direct application of EM is employing a mapping function $Z(\mathbf{x}; \mathcal{X}^{-j})$ from incomplete to complete data, which safeguards against overfitting by returning only hypotheses which arise from the rest of the training corpus $\mathcal{X}^{-j}$ for $\mathbf{x} \in \mathcal{X}^j$. In our case, this translates to substituting the unconstrained binary segmentations set $\boldsymbol{\sigma}(\mathbf{a})$ for each training point $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle \in \mathcal{X}^j$, with the set of hypotheses over the segmentation $\boldsymbol{\sigma}(\mathbf{e}, \mathbf{f}, \mathbf{a}; \mathcal{X}^{-j})$. The latter is equal to the subset of $\boldsymbol{\sigma}(\mathbf{a})$, for which for every $\sigma_1^K \in \boldsymbol{\sigma}(\mathbf{a})$ which segments the sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$ in phrase-pair segments $\sigma_k = \langle \tilde{e}, \tilde{f} \rangle$ $(1 \leq k \leq K)$, the phrase-pair $\langle \tilde{e}, \tilde{f} \rangle$ can be extracted from $\mathcal{X}^{-j}$. This cross-validated segmentation set allows us to avoid considering segmentations which demand from us to supply for the source phrases $\tilde{f}$, target translations $\tilde{e}$ which are solely suggested by the currently examined part of the corpus alone.

Optimisation Objective Following a Cross-Validated Maximum Likelihood Estimation (CV-MLE) objective, we wish to find the estimate $\hat{\theta}^{CV}$ which maximises the Cross-Validated conditional likelihood of the training corpus $\mathcal{X}$ which we have split in J parts, considering only hypotheses over the segmentation of the word-aligned sentence-pairs supplied by $\boldsymbol{\Sigma}(\mathbf{e}, \mathbf{f}, \mathbf{a}; \mathcal{X}^{-j})$.

$$\begin{aligned}\mathcal{L}^{CV}(\mathcal{X}; J, \theta) &= \prod_{j=1}^{J} \prod_{\langle \mathbf{e},\mathbf{f},\mathbf{a}\rangle \in \mathcal{X}^j} p(\mathbf{e}|\mathbf{f}, \mathbf{a}) \\ &= \prod_{j=1}^{J} \prod_{\langle \mathbf{e},\mathbf{f},\mathbf{a}\rangle \in \mathcal{X}^j} \sum_{\sigma_1^K \in \boldsymbol{\Sigma}(\mathbf{e},\mathbf{f},\mathbf{a};\mathcal{X}^{-j})} p(\sigma_1^K; \mathbf{a}) \prod_{\sigma_k \in \sigma_1^K} p(\tilde{e}_k|\tilde{f}_k) \end{aligned} \tag{4.9}$$

As usual with estimation from incomplete data, finding the CV-MLE estimate which maximises (4.9) above cannot be solved analytically and we resort to finding iteratively a local optimum of the likelihood function using an implementation of the EM algorithm, in this case CV-EM. The algorithmic steps of CV-EM from section 3.2.3 take the following form for this estimation problem.

Parameter Initialisation Before the iterative CV-EM process initiates, we first establish the model parameter set for the conditional phrase translation distributions $p(\tilde{e}|\tilde{f})$, by extracting all phrase-pairs from the parallel training corpus

$\mathcal{X}$ which abide by the alignments. We then initialise the model parameters to an initial estimate $\hat{\theta}_0^{CV}$, with all distributions $p(\tilde{e}|\tilde{f})$ set to uniform. After initialisation, the two steps which comprise the iterative part of the CV-EM algorithm take over.

Expectation Step In each iteration r, during the E-step of the CV-EM algorithm we set up the expected CV log-likelihood of the training data, in respect to the parameters $\hat{\theta}_{r-1}$ that were output after the previous iteration $r-1$, starting from $\hat{\theta}_0^{CV}$.

$$Q^{CV}(\theta|\hat{\theta}_{r-1}^{CV}) = E\left[\log \mathcal{L}^{CV}(\mathcal{X}; K, \theta)|\mathcal{X}, K, \hat{\theta}_{r-1}^{CV}\right] = \\ \sum_{j=1}^{J} \sum_{\langle \mathbf{e},\mathbf{f},\mathbf{a}\rangle \in \mathcal{X}^j} \sum_{\sigma_1^K \in \mathbf{\Sigma}(\mathbf{e},\mathbf{f},\mathbf{a};\mathcal{X}^{-j})} \log \left\{ p(\sigma_1^K; \mathbf{a}) \prod_{\sigma_k \in \sigma_1^K} p(\tilde{e}_k|\tilde{f}_k) \right\} q^{CV}(\sigma_1^K|\mathbf{e}, \mathbf{f}, \mathbf{a}, \hat{\theta}_{r-1}^{CV}) \tag{4.10}$$

In preparation of optimising Q^{CV} in the next step, we compute the expected fractional counts q^{CV} of each segmentation as follows, employing the parameter estimates for the conditional translation probabilities $p(\tilde{e}|\tilde{f}; \hat{\theta}_{r-1}^{CV})$ from iteration $r-1$.

$$q^{CV}(\sigma_1^K|\mathbf{e}, \mathbf{f}, \mathbf{a}; \hat{\theta}_{r-1}^{CV}) = \frac{p(\sigma_1^K; \mathbf{a}) \prod_{\sigma_k \in \sigma_1^K} p(\tilde{e}_k|\tilde{f}_k; \hat{\theta}_{r-1}^{CV})}{\sum_{\sigma'^{K'}_1 \in \mathbf{\Sigma}(\mathbf{e},\mathbf{f},\mathbf{a};\mathcal{X}^{-j})} p(\sigma'^{K'}_1; \mathbf{a}) \prod_{\sigma'_{k'} \in \sigma'^{K'}_1} p(\tilde{e}'_{k'}|\tilde{f}'_{k'}; \hat{\theta}_{r-1}^{CV})} \tag{4.11}$$

Maximization Step In the M-step of the CV-EM algorithm, we maximise (4.10) in respect to θ to retrieve the next parameter estimate $\hat{\theta}_r^{CV}$.

$$\hat{\theta}_r^{CV} = \arg\max_{\theta} Q^{CV}(\theta|\hat{\theta}_{r-1}^{CV}) \tag{4.12}$$

Solving the optimisation problem of (4.12) results in each parameter taking a value $p_r(\tilde{e}|\tilde{f})$ proportional to the fractional counts $q(\langle \tilde{e}, \tilde{f}\rangle)$ of the appearance of phrase-pair $\langle \tilde{e}, \tilde{f}\rangle$ in the segmentations of the training data, as these are counted according to the $\hat{\theta}_{r-1}^{CV}$. With δ the Kronecker delta function, we have:

$$q(\langle \tilde{e}, \tilde{f}\rangle; \hat{\theta}_{r-1}^{CV}) = \\ \sum_{j=1}^{J} \sum_{\langle \mathbf{e},\mathbf{f},\mathbf{a}\rangle \in \mathcal{X}^j} \sum_{\sigma_1^K \in \mathbf{\Sigma}(\mathbf{e},\mathbf{f},\mathbf{a};\mathcal{X}^{-j})} q^{CV}(\sigma_1^K|\mathbf{e}, \mathbf{f}, \mathbf{a}; \hat{\theta}_{r-1}^{CV}) \sum_{k=1}^{K} \delta(\tilde{e}_k, \tilde{e})\delta(\tilde{f}_k, \tilde{f}) \tag{4.13}$$

```
INPUT: Word-aligned parallel training data $\mathcal{X}$
OUTPUT: Estimates $\hat{\theta}^{CV}$ for all $p(\tilde{e}|\tilde{f})$

Partition training data $\mathcal{X}$ into $J$ equal parts $\mathcal{X}^1, \ldots, \mathcal{X}^J$
Initialise $\hat{\theta}_0^{CV} = \left\{p_0(\tilde{e}|\tilde{f})\right\}$ to uniform conditional probabilities
Let $r = 0$          // EM iteration counter
Repeat
   Let $r = r + 1$
   E-step:
      For $1 \leq j \leq J$ do
           calculate expected counts for segmentations $\mathbf{\Sigma}(\mathbf{e}, \mathbf{f}, \mathbf{a}; \mathcal{X}^{-j})$
           of each $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle \in \mathcal{X}^j$ using $\hat{\theta}_{r-1}^{CV} = \left\{p_{r-1}(\tilde{e}|\tilde{f})\right\}$
   M-step: calculate probabilities $\hat{\theta}_r^{CV} = p_r(\tilde{e}|\tilde{f})$
       from the expected counts of the E-step
Until $\hat{\theta}_r^{CV}$ has converged
```

Figure 4.5: CV-EM implementation pseudocode.

$$p_r(\tilde{e}|\tilde{f}) = \frac{q(\langle \tilde{e}, \tilde{f} \rangle; \hat{\theta}_{r-1}^{CV})}{\sum_{\tilde{e}'} q(\langle \tilde{e}', \tilde{f} \rangle; \hat{\theta}_{r-1}^{CV})} \tag{4.14}$$

The new set of parameters $\hat{\theta}_r^{CV}$ is fed into the next iteration and the process continues until convergence. Figure 4.5 summarises in pseudocode the application of CV-EM for the estimation of this phrase-based translation model.

4.4.2 Jackknife Averaging

In addition to the standard application of CV-EM that we present above, we also formulate and employ in our empirical experiments a variation of it inspired by the Jackknife re-estimation method (Quenouille, 1949; Tukey, 1958)[3]. Similarly to CV, the Jackknife estimate is equal to the average of the estimator output when applied on different parts of a partition of the training data, when each time a single part is excluded from the estimator's input. As with CV, Jackknife estimation also aims at trading estimator bias to reduce the error due to variance of the estimates in relation to the input training sample.

We employ the Jackknife approach to arrive at a variation of the CV-EM which aims to reduce the variance error of our estimates. While CV-EM focuses on the cross-validation of the segmentation hypothesis space, which has an indirect

[3] An up-to-date presentation of the Jackknife can be found in (Duda et al., 2001).

smoothing effect on the resulting estimates, our application of Jackknife directly applies smoothing on the temporary parameter estimates themselves during every iteration.

Namely, during every iteration's E-step, instead of collecting a single expected counts value $q(\langle\tilde{e},\tilde{f}\rangle;\hat{\theta}^{CV}_{r-1})$, we compute and store the expected counts of each phrase-pair in the segmentations of each part of the corpus $\mathcal{X}^j$ *separately.*

$$q^j(\langle\tilde{e},\tilde{f}\rangle;\hat{\theta}^{CV}_{r-1}) = \sum_{\langle\mathbf{e},\mathbf{f},\mathbf{a}\rangle\in\mathcal{X}^j} \sum_{\sigma_1^K\in\mathbf{\Sigma}(\mathbf{e},\mathbf{f},\mathbf{a};\mathcal{X}^{-j})} q^{CV}(\sigma_1^K|\mathbf{e},\mathbf{f},\mathbf{a};\hat{\theta}^{CV}_{r-1}) \sum_{k=1}^{K} \delta(\tilde{e}_k,\tilde{e})\delta(\tilde{f}_k,\tilde{f}) \quad (4.15)$$

From these counts, we compute during the M-step a separate estimate $p^j_r(\tilde{e}|\tilde{f})$ of each parameter value $p(\tilde{e}|\tilde{f})$ from every part of the training data. The Jackknife estimate $p_r(\tilde{e}|\tilde{f})$ for each iteration is then computed by averaging together the respective J estimates, one from each training data part. These values comprise the temporary estimate $\hat{\theta}^{CV}_r$ for iteration r, which is then fed to the following iteration of the algorithm.

$$p^j_r(\tilde{e}|\tilde{f}) = \frac{q^j(\langle\tilde{e},\tilde{f}\rangle;\hat{\theta}^{CV}_{r-1})}{\sum_{\tilde{e}'} q^j(\langle\tilde{e}',\tilde{f}\rangle;\hat{\theta}^{CV}_{r-1})} \quad (4.16)$$

$$p_r(\tilde{e}|\tilde{f}) = \frac{\sum_{j=1}^{J} p^j_r(\tilde{e}|\tilde{f})}{J} \quad (4.17)$$

This averaging of the estimates has a smoothing effect on the final output $\hat{\theta}^{CV}_r$ of each iteration. It reduces the impact of observations or hypotheses over latent variables whose appearance in the training data is characterised by a large variance. In contrast, parameter values relating to events or hidden values (e.g. use of phrase-pairs in segmentations) whose frequencies are similar across the different parts $\mathcal{X}^j$, will stay largely unaffected by averaging through the data partition.

The disadvantage of applying this averaging operation between the estimates from the different parts of the data is that, by tampering with the workflow of the EM algorithm, we cannot lay claim anymore to the important algorithmic properties that CV-EM inherits from EM. It is not clear from a theoretical perspective that the Jackknife variation of CV-EM that we present here converges towards an estimate that satisfies a specific condition, as it is the case with the direct CV-EM implementation of section 4.4.1. Nevertheless, in all the experiments that we present later in this chapter, this variation of CV-EM always converged and the estimates produced were relatively strong in relation to both the direct CV-EM estimates as well as those of the baseline.

4.4.3 Smoothing and Implementation Details

There are two special boundary cases which demand our attention during estimation.

Completing the Derivations Enforcing the segmentation space $\mathbf{\Sigma}(\mathbf{e}, \mathbf{f}, \mathbf{a}; \mathcal{X}^{-j})$ strictly will not allow us to find segmentations for many sentence pairs. A striking example is any sentence-pair with a source or target word that appears only once in our training corpus. In order to avoid this, in practice we allow all binary segmentations $\mathbf{\Sigma}(\mathbf{a})$ to be considered, while strongly penalising segmentations which are not included in $\mathbf{\Sigma}(\mathbf{e}, \mathbf{f}, \mathbf{a}; \mathcal{X}^{-j})$.

In more detail, while our primary estimation target are parameters $p(\tilde{e}|\tilde{f})$ for phrase-pairs $\langle \tilde{e}, \tilde{f} \rangle$ which survive cross-validation by appearing in at least two different parts of the data, we also allow the rest of the phrase-pairs to take part in sentence-pair derivations with a fixed conditional translation probability. This is set to $10^{-5 * \tilde{m}}$, where $\tilde{m}$ is the length of the source phrase $\tilde{f}$, in essence employing a word-level conditional translation model with a fixed word translation probability equal to 10^{-5}.

This choice puts at a severe disadvantage derivations of a sentence-pair using long penalised phrase-pairs, strongly promoting derivations which employ the cross-validated phrase-pairs as much as possible. The role of the smoothing phrase-pairs is then to merely complete derivations for the synchronous spans which could not be otherwise covered.

Zero distributions In the case of the Jackknife CV-EM variation, when a phrase $\tilde{f}$ does not occur in $\mathcal{X}^j$, all its pairs $\tilde{e}$ in the phrase table will amass zero counts from this part of the training data. Its CV-MLE estimate from $\mathcal{X}^j$ is then undefined, since it is irrelevant for computing the likelihood of $\mathcal{X}^j$. This creates a problem during each iteration when the estimates from each $\mathcal{X}^j$ are averaged together to compute the Jackknife estimate. While in this case any choice of a distribution $P(\cdot|\tilde{f})$ will constitute an MLE solution for $\mathcal{X}^j$, we choose to set this case to the minimally informed uniform distribution when averaging.

Algorithmic Implementation Since our model is formulated within the binary SCFG framework, we employ a bilingual CYK parser (Younger, 1967). This parser uses a grammar which includes the rules in (4.6), complemented with the phrase emitting productions. It builds for every input $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle$ all binarisations/derivations for every segmentation in $\boldsymbol{\sigma}(\mathbf{a})$. For implementing CV-EM, we employ the bilingual extension of the Inside-Outside algorithm (Lari and Young, 1990; Goodman, 1998) on top of the parser. During estimation, because the input, output and word-alignment are known in advance, the time and space requirements remain manageable despite the worst-case complexity $O(n^6)$ for sentence-pair length n.

4.5 Experiments

To evaluate the strength of our CV-EM estimates computed according to the CV-MLE optimisation objective in comparison to their heuristic counterparts, we integrate these in turn in a Phrase-Based SMT decoder. We use French to English translation as our primary language pair to test the performance of an array of estimation and decoding configurations. We then further evaluate our best performing configurations against the baseline for a German to English translation task.

4.5.1 Decoding and Baseline Model

In this work, we compare against a state-of-the-art PBSMT baseline, based on the feature-based model of equation (4.1). The feature functions ϕ employed are: a 5-gram target language model, the standard reordering scores, the word and phrase penalty scores, the conditional lexical estimates obtained from the word-alignment in both directions and the conditional phrase translation heuristic estimates of equations (4.3) in both directions $p(\tilde{e}|\tilde{f})$ and $p(\tilde{f}|\tilde{e})$. The feature weights λ are optimised by Minimum Error-Rate Training (MERT) (Och, 2003). For decoding according to (4.2) we use the Moses decoder (Hoang and Koehn, 2008).

We compare our estimates of $p(\tilde{e}|\tilde{f})$ and $p(\tilde{f}|\tilde{e})$ to the commonly used heuristic estimates, by substituting the latter with the values obtained through CV-MLE estimation and decoding while keeping the rest of the feature functions fixed. Even though the exposition in this chapter follows the estimation of translation probabilities $p(\tilde{e}|\tilde{f})$ of target phrases given the source phrases, the estimates for the opposite translation direction can be readily computed by reversing the roles of the two languages in the language-pair. We use estimates computed using both variations of the CV-EM algorithm: the standard application of it (CV-EM) and the Jackknife variation (J-CV). Before we decode with each translation model parameter set, we recompute the feature weights with MERT.

Because our model employs a latent segmentation variable, this variable should be marginalised out during decoding to allow selecting the highest probability translation given the input. This could turn out crucial for improved results, as noted by (Blunsom et al., 2008a). However, such a marginalisation can be NP-Complete, in analogy to the similar problem in Data-Oriented Parsing (Sima'an, 2002). Since we do not have access to a decoder that can approximate this marginalisation efficiently, we employ the standard Moses decoder for this work which searches for the highest scoring phrase-based derivation.

4.5.2 Experimental Setup

The training, development and test data all come from the French-English translation shared task of the ACL 2007 Second Workshop on Statistical Machine

Phrase Length	**System**	**BLEU**
≤ 7	Baseline	33.03
≤ 10	Baseline	33.03
All	Baseline	33.00
≤ 7	EM + ITG Prior	32.50
≤ 7	CV-EM	32.67
≤ 7	CV-EM + ITG Prior	32.73
≤ 7	J-CV + ITG Prior	33.02
≤ 10	J-CV + ITG Prior	**33.14**
All	J-CV + ITG Prior	32.98

Table 4.1: French to English Translation: A comparison of the the heuristic estimates (Baseline) with estimates of the CV-EM algorithm and its Jackknife variation (J-CV). BLEU scores are computed by integrating each parameter set in the feature-based Moses decoder with weights trained by MERT.

Translation [4]. After pruning sentence pairs with word length more than 40 on either side, we are left with 949K sentence pairs for training. The development and test data are composed of 2K sentence pairs each. All data sets are lower-cased.

For both the baseline system and our method, we produce word-level alignments for the parallel training corpus using GIZA++. We use 5 iterations of each IBM Model 1 and HMM alignment models, followed by 3 iterations of each Model 3 and Model 4. From this aligned training corpus, we extract the phrase pairs according to the heuristics in (Koehn et al., 2003). The baseline system extracts all phrase-pairs up to a maximum length 7 on both sides and employs the heuristic estimator. The language model used in all systems is a 5-gram language model trained on the English side of the parallel corpus. Minimum-Error Rate Training (MERT) is applied on the development set to obtain optimal log-linear interpolation weights for all systems. Performance is measured by computing the BLEU scores (Papineni et al., 2002) of the system's translations, when compared against a single reference translation per sentence.

4.5.3 Results

We compare different versions of our system against the baseline system using the heuristic estimator. We observe the effects of the ITG segmentation prior in the translation model as well as the method of estimation: (a) direct application of CV-EM vs. (b) using Jackknife averaging during CV-EM's iterations (J-CV).

Table 4.1 exhibits the BLEU scores for the systems. Our own system (with ITG segmentation prior and J-CV estimation with a maximum phrase-length

[4] http://www.statmt.org/wmt07

Phrase Length	System	BLEU
≤ 7	Baseline	28.18
≤ 10	Baseline	28.34
All	Baseline	28.27
≤ 10	J-CV + ITG Prior	**28.46**
All	J-CV + ITG Prior	28.30

Table 4.2: German to English Translation.

used during decoding of ten words) scores (33.14), slightly outperforming the best baseline system (33.03). When using straight CV-EM, this leads to a lower score (32.73). When also the ITG prior is excluded (by having a single derivation per segmentation) this leads to a further score reduction (32.67). By directly applying EM with an ITG prior (turning off the cross-validation of the segmentation hypotheses space), performance goes down to 32.50. Estimation in this last case severely overfits. The only reason that the BLEU score does not completely collapse is that Moses falls back on the rest of the feature functions, such as the lexical smoothing translation probabilities. We did not explore mere EM without any smoothing or ITG prior, as we expect it will directly overfit the training data as reported by (DeNero et al., 2006). Overall, these results exhibit the crucial role of the estimation by smoothing, with CV-EM estimation and the ITG segmentation prior clearly emerging as key components behind the improved phrase translation estimates.

As table 4.1 shows we also varied the phrase length cut-off (seven, ten or none={all phrase pairs}) during decoding, with this cut-off value pertaining to both sides of a phrase-pair. It is important to distinguish this decoding-time cut-off from what applies during estimation-time for our model. We always train *all* phrase-pairs and apply the length cut-off only during decoding (no renormalisation is applied at that point).

Interestingly, we find out that in this case the heuristic estimator cannot benefit performance by including longer phrase pairs. Our estimator does benefit performance by including phrase pairs of length up to ten words, but then it degrades again when including all phrase pairs. We take the latter finding to signal remaining overfitting that proved resistant to the smoothing applied by our estimator. The heuristic estimator exhibits a similar degradation.

German to English Translation After comparing in detail the baseline system against different estimation algorithms and decoding-time phrase length limits for French to English translation, we perform further experiments on the German to English translation direction. In this way we wish to confirm the applicability of our estimation methods to translate between languages characterised by more pronounced differences than our primary French-English language-pair. The parallel training data are again part of the ACL 2007 Second Workshop on Statistical Machine Translation and after similar pre-processing as for French to English, we are left with 996K training sentence-pairs. The development and test sets are made also in this case from 2K sentence-pairs each.

We follow the same methodology to train our estimates and decode, resulting in the BLEU scores listed in Table 4.2 as first presented in (Sima'an and Mylonakis, 2008). Examining the results reveals that the combination of the Jackknife CV-EM process with the ITG segmentation prior, scores better than the best performing baseline system, although the margin between them is small. The best configuration in relation to the decoding-time phrase length cut-off value is 10 for both the baseline system as well as when using our estimates. Nevertheless, the performance of both systems degrades after the phrase-pair length constraint is removed altogether. These results indicate that the default length cut-off value of 7 might be sub-optimal for some language-pairs. Even though completely removing this constraint is not the optimal choice, investigating empirically what is a good choice for it could pay off.

4.6 Discussion

In this chapter, we explored training phrase-based translation models which explicitly employ phrase-pair segmentation variables. This is in contrast with most work on PBSMT which chooses to bypass the problem of disambiguating the segmentation of sentence-pairs into phrase-pairs and opts for heuristic estimates instead. The most similar efforts to ours, mainly (DeNero et al., 2006), conclude that segmentation variables in the generative translation model lead to overfitting while attaining higher likelihood of the training data than the heuristic estimator.

In this work we also start out from a generative model with latent segmentation variables. However, we find out that concentrating the learning effort on increasing the generalisation capacity of our estimates is crucial for good performance. While we do find employing probabilistic priors as our ITG segmentation prior useful, we do not centre our method on employing external knowledge in the form of priors over the parameter space neither we switch to a Bayesian formulation of the learning problem as in (Blunsom et al., 2008a). Following this research path is a choice which, while promising, still poses significant issues and challenges as we discuss in 3.2.4, as well as demands a significant departure from approaches such as MLE and EM which have long proven themselves successful in both NLP as well as SMT in particular. Instead, here we choose to remain in the domain of the Maximum Likelihood Estimation, while taking a data-driven approach to address the issues of MLE application for Fragment Models such as those employed in phrase-based translation.

Our approach is centred around the application of CV-EM which aims to maximise the likelihood of the training data while considering a cross-validated set of hypotheses over the missing phrase-pair segmentation of the training sentence-pairs. This application of cross-validation allows us to employ the training data itself to essentially simulate maximising the likelihood of yet unseen data instances and directly aim at increased generalisation. Pursuing this objective takes place within a well-founded and clear learning framework whose implementation is appealing computationally. The fact that our results (at least) match the heuristic estimates on a reasonably sized data set (947k parallel sentence pairs) is rather encouraging.

Another aspect of the work presented in this chapter is the employment of the the binary segmentation space considered by phrasal Inversion Transduction Grammars. This greatly reduces the number of segmentations considered and allows to efficiently pack together using dynamic programming the derivations of sentence-pairs from phrase-pairs. Our results indicate that constricting the segmentation space in this way is a reasonable choice when learning phrase-based translation models, connecting under a different perspective with the work of (Wu, 1997; Huang et al., 2009) and others.

While the best scoring estimates were computed by augmenting CV-EM with a Jackknife estimation step which further smooths the parameter estimates during every iteration, in the following chapters we opt to rely on the standard application of CV-EM for each learning problem. Our primary aim in this thesis from a learning perspective is understanding the impact of the learning framework of CV-MLE under CV-EM rather than tweaking for maximum performance. Since the implications of combining Jackknife with CV-EM to the algorithmic and statistical estimation properties of the latter are not clear, an issue which is further aggravated as our models become more complex, we choose to focus further on CV-EM proper in the following chapters. Nevertheless, the results in this chapter indicate that solutions such as Jackknife to complement the effort of CV-EM to increase the generalisation capacity of our estimates should be empirically evaluated when applying CV-EM for particular estimation problems.

Overall, the results in this chapter signify in the context of this thesis the establishment of a first set of empirical evidence highlighting the merits of CV-EM for Fragment Model estimation in the context of SMT. Estimates computed with CV-EM can substitute successfully the heuristically estimated ones without loss of translation performance, for two different language pairs and when training on a reasonably sized corpus. These positive results motivate the work in the following chapters, where we employ CV-EM for a pair of translation models where the focus gradually shifts towards the latent structure of phrase-based translation. Based on the success of binary SCFGs to efficiently constrict the segmentation space for PBSMT, we move on to explore syntactic approaches to SMT, where we employ probabilistic synchronous grammars as the central component of our translation models.

Chapter 5

Learning Stochastic Synchronous Grammars

In this chapter we build on the results of Chapter 4 to extend our method to the estimation of syntactically driven translation models, using binary Synchronous Context-Free Grammars (SCFGs) as our formalism of choice. As part of our work on estimating the parameters of phrase-based translation models, we already focused on the space of binary phrase reorderings to constrain successfully the segmentation space and define a prior over it. Now, we move further to bring the recursive nature of binary SCFGs at the centre of our attention, formulating a probabilistic SCFG joint model to describe translation.

Increasing the modelling stress on the latent translation structure necessitates a closer examination of its role and possible deficiencies. For this reason, we empirically consider alternatives in order to identify a translation structure strong enough to function as the backbone holding together the source and target sides of our modelling problem. Noting certain deficiencies of the independence assumptions behind SCFGs, we contribute a lexically sensitive reordering structure which propagates reordering decisions to higher and lower levels of a derivation, in order to widen the role of the abstract recursive translation structure past the rudimentary use that it finds in (Chiang, 2005a).

In comparison with the state-of-the-art, this work as first presented in (Mylonakis and Sima'an, 2010) contributes a method to learn phrase-based synchronous grammars for machine translation, aiming to discover reusable lexical and structural translation patterns which generalise well. We further contribute a particular grammar formalism which puts the focus on orchestrating phrase reordering across the full length of the sentence-pair.

We do not explore synchronous grammars which enrich synchronous productions with lexical context and which allow modelling translation with discontiguous phrases. While these grammars have been shown to offer competitive translation performance (Chiang, 2005a), in this chapter we choose to focus on the implications of learning the unlexicalised recursive structure of synchronous grammars. Nevertheless, our learning objective and implementation based on CV-EM, together with the grammar design we contribute, allows us to reach the same level of performance as our hierarchical phrase-based translation baseline which does use lexicalised recursive reordering.

In the context of the thesis, in this chapter we move further than merely estimating phrase translation probabilities as we did in Chapter 4, and integrate them as part of a more comprehensive model which handles all other aspects of translation such as reordering. We proceed towards learning a full translation model capturing both phrase translation and reordering patterns, which will be the key component taking care of the lexical and structural transfer from source to target during decoding. We consider the implications of describing the latent translation structure using the synchronous grammar formalism and take the first step in formulating a learning environment for a relatively simple grammar design. Our findings form a crucial step before proceeding in the following chapter towards discovering intricate, linguistically motivated grammatical structures capturing the translation process.

5.1 Focusing on Translation Structure

Probabilistic phrase-based synchronous grammars are currently considered promising devices for Statistical Machine Translation (SMT), with systems based on this formalism achieving state-of-the-art translation performance. This is especially true when these models are applied to translate between languages with significant differences in their syntax such as Chinese-English. Modelling translation using phrase-based Synchronous Context-Free Grammars (Wu, 1997; Chiang, 2005a) builds upon the strengths of Phrase-Based SMT (PBSMT) (Och et al., 1999; Koehn et al., 2003), while bringing together and extending the different components of a phrase-based system under a single modelling component.

On the one hand, probabilistic SCFGs inherit from the PBSMT models the ability to build models that can reuse memorised multi-word fragments and their translations. This is a powerful feature that essentially allows *forfeiting* for certain translation patterns the strong independence assumptions posed by word-based models. On the other hand, using synchronous grammars for SMT recasts the reordering problem in terms of establishing a syntactic correspondence between the two languages, unifying the usually separately conceived phrase translation and reordering components of PBSMT systems in a single grammatical formalism (Wu, 1997). In addition, the recursive nature of SCFGs coupled with the concept of modelling translation on the phrase level allows the formulation of hierarchical phrase-based models (Chiang, 2005a) making use of recursive lexical translation patterns, sometimes colloquially referred to as phrase-pairs with ‘gaps’.

In general, discontiguous phrase translation patterns need not necessarily be modelled through a synchronous grammar formalism. This is exhibited by

(Simard et al., 2005), where a standard contiguous PBSMT framework is extended to allow non-contiguous phrase-pairs with missing word placeholders. Still, the recursive nature of SCFGs, apart from the mechanics to allow modelling translation with discontiguous phrase-pairs, puts in place the necessary descriptive power to capture the impact of the hierarchical nature of language in translation.

Nevertheless, following their introduction, the focus on applying hierarchical SCFG-based models was mostly concentrated on the ability to model translation using discontiguous phrase-pairs, leaving the capacity of the model to handle short and long-range syntactic constraints in abstract terms mostly unexploited. This has been largely handled in lexical terms, through the use of recursive phrases with gaps which can trigger certain reorderings in relation with lexical context patterns. However, developments over the last few years have shifted the attention of the research community towards the ability of SCFGs to describe the structural aspects of translation on a level further afield than the lexical surface.

Crucially, the transition from PBSMT to SCFG-based translation was not marked by a similar step towards a better-founded learning framework, leaving the stochastic part of hierarchical models to be founded on the same heuristic methods used in PBSMT. Estimation based on the extraction counts of phrase-pairs was extended to their discontiguous counterparts (Chiang, 2005a), sometimes reaching past the lexical surface and up to the structural part of SCFG analyses (Zollmann and Venugopal, 2006). Some of the reasons behind opting for heuristic estimation in syntax-driven, phrase-based SMT approaches are similar to those encountered in their Phrase-Based SMT forerunners. Embedding the concept of modelling with multi-word fragments of arbitrary lengths in a syntactic framework does not make us any less liable to the same estimation challenges related to contiguous phrase-based models.

Even though the issues related to learning phrase-based models alone are daunting enough, learning synchronous grammars brings in additional aspects to the learning problem. Apart from lexical choice, it also involves training a structural component which takes over the reordering task from the reordering models of PBSMT. This modelling component is concerned with the syntactic well-formedness of the whole sentence, matching long-range syntactic preferences that the reordering models of PBSMT do not usually consider. In addition, the lexical and the structural parts of synchronous grammars can be tightly interlocked together, with the syntactic structure affecting the corresponding lexical choice and vice versa.

As a result of this interplay between the lexical and the structural aspect of synchronous grammars, the estimation challenges of phrase-based models reach out to the structural part as well. The tendency to overfit guides the estimator towards hypotheses translating as much of the source sentence as possible as part of long discontiguous phrase-pairs. This prohibits learning how to combine smaller fragments together and results in models which support only a trivial structure reaching up to the largest fragments allowed by the training constraints.

Avoiding such degenerate hypotheses will allow the estimator to discover not only reasonable phrase correspondences which we hope will be useful to analyse yet unseen data, but also to learn how to combine together these reusable building blocks recursively. The learning environment we will use to work towards this aim will again be the Cross-Validated EM algorithm of section 3.2.

5.2 Synchronous Grammars for SMT

Synchronous grammars extend the descriptive power of formal grammars from single strings to tuples of strings. They can be used to define a language over pairs of strings and are highly interesting for machine translation, as they can capture the correspondences between source sentences and their target language translations. Furthermore, each particular grammar formalism may offer an explanation of the compositional mechanics of translation which allows us to describe compactly the correspondences between a countably infinite set of sentence-pairs. While we may consider a wide range of such formalisms[1], the one which enjoys the widest acceptance in the MT community are the Synchronous Context-Free Grammars (SCFGs) of (Wu, 1997) and (Chiang, 2005a).

As we discuss in more detail in section 2.4, an SCFG defines a language over string-pairs by means of a recursive rewrite process. In a monolingual Context-Free Grammar, starting from a start symbol S we recursively expand left-hand side non-terminals according to the right-hand side of grammar production rules, rewriting each non-terminal as a string of terminals and novel non-terminals which need to be further expanded. This process continues until we end up with a string of terminal symbols, which then by definition belongs to the language of the grammar. In SCFGs, this rewrite process is *synchronous*, operating on a pair of strings of terminals and pair-wise linked non-terminals, expanding at every rewrite step a single such pair of non-terminals in both sides of the string-pair according to the grammar rules. These rules map a left-hand side of a single non-terminal pair towards a right-hand side consisting of a pair of strings of terminals and non-terminals, with the latter paired together across both sides of the right-hand side expansion.

An example on how the synchronous rewrite process can be employed to capture translation phenomena between language-pairs, repeated here from Chapter 2 for the reader's convenience, is presented in Figure 5.1. There, we denote linked non-terminals across both parts of the right-hand sides of the synchronous rules by attaching the same subscript indexes, while, to simplify notation, we assume without loss of generality that each linked pair involves non-terminals of the same type. The handful of rules in this small grammar already showcases how SCFG rewrite rules have the potential to encode the abstract syntactical transformations

[1] Some of the other formalisms to describe bilingual data are listed in section 2.4.1.

$$\begin{aligned}
S &\rightarrow X_{\boxed{1}} \;/\; X_{\boxed{1}} \\
S &\rightarrow \text{Do } X_{\boxed{1}} \text{ ? } / \; X_{\boxed{1}} \text{ ka ?} \\
X &\rightarrow NP_{\boxed{1}}\, VB_{\boxed{2}}\, NP_{\boxed{3}} \;/\; NP_{\boxed{1}}\, NP_{\boxed{3}}\, VB_{\boxed{2}} \\
NP &\rightarrow \text{I / watashi ga} \\
VB &\rightarrow \text{open / akemasu} \\
NP &\rightarrow \text{the book / hako o}
\end{aligned}$$

Figure 5.1: An SCFG rule set for SVO to SOV reordering and question construction from English to (romanised) Japanese, adapted from (Chiang, 2005b)

and reordering patterns as well as the lexical correspondences between the language pair, possibly combining both abstract and lexical aspects in single rewrite operations (rule 2).

5.2.1 Grammar Design

Approaches considering the use of recursive structure and formal grammars for MT draw inspiration from the related monolingual task of natural language parsing. Superficially, the flavour of syntactic MT that is relevant for this thesis seems highly related to the majority of research on natural language parsing, as they are both occupied with analysing human language manifestations drawing from a common pool of resources, i.e. formal grammars and the related algorithms and learning frameworks. Nevertheless, while a certain link and influence between the two fields undeniably exists, there are fundamental differences with respect to the role of syntax in the learning problems behind syntactic MT and parsing.

Firstly, the structure of Machine Translation is *latent*, rendering the problem of identifying it as an instance of unsupervised learning. On the contrary, the majority of research on natural language parsing is occupied with the supervised learning of a predefined flavour of language structure, using labelled corpora such as the Penn Treebank (Marcus et al., 1993). While all kinds of learning share common problems such as overfitting and treating yet unseen instances, the unsupervised nature of learning syntactic models for MT brings in the novel challenge of learning from incomplete data, in comparison to supervised monolingual parsing.

Still, one could argue perhaps that syntactic MT is more reminiscent then of the field of unsupervised parsing (van Zaanen, 2000; Clark, 2001; Klein and Manning, 2004; Bod, 2007), which considers the unsupervised learning of language structure from unlabelled corpora. However, while in both cases the task is to learn latent natural language structure training merely on the lexical surface, there is a crucial difference between the two in relation to the role of the latent structure in respect to the overall NLP system. It has been recognised (e.g. (Bod, 2007)) that, in the long term, attention in evaluating unsupervised parsing must be shifted towards more high-level tasks taking advantage of syntactic analyses, such as sense disambiguation and MT itself. Nevertheless, for the time being, most work on unsupervised parsing is fixated and evaluated on replicating certain linguistic annotations, like those (derived from) the aforementioned Penn Treebank. As long as discovering the syntactic structure of language remains the final aim of unsupervised parsing, it will boil down to discovering a *certain kind* of syntax, as otherwise a meaningful comparison between the different approaches seems impossible.

In contrast, in Machine Translation any latent variable assumed by a model is usually not interesting on its own, and is evaluated instead in the context of how well it captures the correspondence between the sentences of the language-pair. This extends to the syntactic variables used in MT models, such as those backed by SCFGs. Our aim is to raise the translation performance, by integrating as part of an MT model syntactic formalisms and annotations. The extent to which we will be successfull in this relies on our capacity to learn these latent variables from the incomplete parallel corpus and subsequently translate better employing them. While towards this end features of linguistic syntax as they evolved for monolingual parsing can be useful, overall syntax-based MT is not bound to a particular annotation scheme.

This leaves substantial space to consider different synchronous *grammar designs* to explain the translation process, and we venture to explore part of this space in this thesis. Figures 5.2 and 5.3 showcase two different views on a syntactic analysis of the translation of secondary clauses between English and German, using the sentence fragment pair '*which is the solution / der die Lösung ist*' as a particular example. The grammar of Figure 5.2 uses the linguistic structure of the English sentence to pivot between the two languages, while that of Figure 5.3 focuses on lexical cues to signal the characteristic reordering of verbs in these sentence-pairs. Finally, Figure 5.4 takes a hybrid approach, reducing the ambiguity when applying the lexically grounded rules of Figure 5.3 using linguistic constituency information.

One may argue about the merits of each grammar design on linguistic, cognitive or other grounds and these arguments are valid as long as we wish to move further than the machine translation task, e.g. by aiming to discover how the human brain translates and so forth. Still, as long as translating automatically is what we aim for and systems are evaluated on the quality of translations that they offer, establishing different grammar designs and choosing between them remains an empirical task. It involves assessing not only the descriptive powers of each synchronous grammar family, but also our ability to learn an effective synchronous

$$\begin{aligned}
SBAR &\rightarrow WHNP_{\boxed{1}}\ VP_{\boxed{2}}\ /\ WHNP_{\boxed{1}}\ VP_{\boxed{2}}\\
VP &\rightarrow VBZ_{\boxed{1}}\ NP_{\boxed{2}}\ /\ NP_{\boxed{2}}\ VBZ_{\boxed{1}}\\
WHNP &\rightarrow \text{which / der}\\
VBZ &\rightarrow \text{is / ist}\\
NP &\rightarrow \text{the solution / die Lösung}
\end{aligned}$$

Figure 5.2: An SCFG rule set for secondary clause verb reordering between English to German based on abstract linguistic structure.

$$\begin{aligned}
X &\rightarrow \text{which is } X_{\boxed{1}}\ /\ \text{der } X_{\boxed{1}}\ \text{ist}\\
X &\rightarrow \text{the solution / die Lösung}
\end{aligned}$$

Figure 5.3: An SCFG rule set for secondary clause verb reordering between English to German based on lexical context.

$$\begin{aligned}
SBAR &\rightarrow \text{which is } NP_{\boxed{1}}\ /\ \text{der } NP_{\boxed{1}}\ \text{ist}\\
NP &\rightarrow \text{the solution / die Lösung}
\end{aligned}$$

Figure 5.4: An SCFG hybrid rule set for secondary clause verb reordering between English to German, combining lexical context with linguistic constituency information.

grammar belonging to it from the available training data and determining the extent to which the grammars induced actually lead to strong translations during decoding.

In the end, a strong synchronous grammar design is the one which pairs well with the learning approach that we employ to learn from data and the decoding schemes in which we embed our grammars to translate. Evaluating the strengths and weaknesses of a grammar design should only be performed within the context of a specific MT system implementation, or even a particular language pair or training and test data domain. The synchronous grammar formalisms we employ here, when trained with plain MLE lead to *degenerate* models that translate extremely poorly yet unseen source sentences, as we discussed in the wider context of Fragment Models in section 3.1.5. The exact same synchronous grammar designs provide *state-of-the-art* results when trained with CV-MLE as we show later in this chapter.

5.2.2 SCFG Modelling & Its Pitfalls

Probabilistic SCFGs extend the synchronous grammar rules to arrive at a stochastic *joint* model over string pairs. A probability value is attached to every grammar rule, so that these probabilities sum up to one for all rules having the same left-hand side. The key assumption behind this SCFG model is, similarly to the case of monolingual CFGs, that each non-terminal pair rewrite operation is independent of the rest of the derivation of the string-pair, given this non-terminal pair that we currently expand. The probability of a derivation D of a string-pair $\langle \mathbf{e}, \mathbf{f} \rangle$ is then the product of the probabilities of all rules r used in D, and the probability of the string-pair itself is the sum of the probabilities of all derivations $D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle$ leading to it.

$$p(D) = \prod_{r \in D} p(r) \tag{5.1}$$

$$p(\mathbf{e}, \mathbf{f}) = \sum_{D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle} p(D) \tag{5.2}$$

This basic independence assumption behind SCFG models generalises the concept of a constituent from monolingual CFGs to their bilingual version. Right-hand side expansions covered by the same left-hand side non-terminal pair can be considered interchangeable, with the rule probabilities indicating how probable it is that they can be applied to rewrite this left-hand side non-terminal pair. For every right-hand side taking part in a synchronous rule, it is solely the left-hand side of the rule that will determine how the bilingual span covered by it will combine with the higher levels of the derivation. Accordingly, further expansions of the still abstract parts of the right-hand side are conditioned only on the non-terminal pairs that still need to be rewritten.

$$\begin{aligned}
NP &\rightarrow JJ_{\boxed{1}}\ NN_{\boxed{2}}\ /\ NN_{\boxed{2}}\ JJ_{\boxed{1}} && r_1 : p(r_1)\\
NP &\rightarrow JJ_{\boxed{1}}\ NN_{\boxed{2}}\ /\ JJ_{\boxed{1}}\ NN_{\boxed{2}} && r_2 : p(r_2)\\
NN &\rightarrow \text{box / boîte} && r_3 : p(r_3)\\
JJ &\rightarrow \text{blue / bleue} && r_4 : p(r_4)\\
JJ &\rightarrow \text{beautiful / belle} && r_5 : p(r_5)
\end{aligned}$$

Figure 5.5: An SCFG grammar rule-set categorising both word-pairs '*blue / bleue*' and '*beautiful / belle*' under the same non-terminal JJ, failing to take into account the different reordering patterns that these participate in. For $p(r_1) > p(r_2)$, the model will prefer to translate the input '*beautiful box*' wrongly as '*boîte belle'*.

Crucially, this leaves one of the most important components of the synchronous rule unaccounted for, when SCFG rules are combined to form a derivation. The *reordering pattern* between the non-terminals of the right-hand side is not an explicit part of the conditioning context in SCFG models, which is limited to the identity of the non-terminal pairs that function as left-hand sides. This may constitute a modelling pitfall that has received surprisingly little attention in the syntax-based MT community.

The concept of a constituent in monolingual CFGs describes strings of terminals and non-terminals which can substitute for each other as alternative expansions of the same covering left-hand side. This implied interchangeability in the monolingual case is justified by the ability of expansions covered by the same non-terminal to combine with similar surrounding contexts. For example, in English, noun phrases can combine to the left or to the right with verb clusters as subjects or objects of a sentence respectively, and nouns occur frequently after determiners or close to adjectives.

Importantly, when we move from monolingual to bilingual (or multi-lingual) grammars, the concept of a synchronous constituent and that of substitution must move further than taking into account the surrounding context in the two languages being modelled, for each of the two parts of the right-hand side expansions. An SCFG non-terminal pair must cover not only bilingual constituents whose two parts combine together similarly within each of the two languages of the language-pair, but they must *also* take part in similar reordering patterns.

A simple example illustrating this for translation between English and French can be seen in Figure 5.5. The word-pairs '*blue / bleue*' and '*beautiful / belle*' are both assigned the same non-terminal category JJ. This decision can be based on the observation that '*blue*' and '*beautiful*' can frequently substitute syntactically each other in English sentences and similarly '*bleue*' and '*belle*' do so in French sentences. However, this fails to take under account that while the monolingual parts of the two bilingual constituents behave similarly in each of the two languages, when joined as word-pairs they combine quite differently in regard to how they reorder in sentence-pairs. The result is that according to the SCFG model, translations of adjective-noun English phrases will always be swapped if the first more common reordering pattern in rule r_1 is correctly assigned a larger probability than the more infrequent r_2, even when encountering exceptions such as '*beautiful / belle*'.

As we see next, in practice SCFG-based models of translation are complemented in state-of-the-art syntax-based SMT systems such as (Chiang, 2005a) with an array of additional features including a target language model, which can counter to a certain extent this modelling weakness. However, these systems make limited use of the abstract recursive structure offered by SCFGs, based mostly on reordering based on lexical context. We believe that, when learning SCFG grammars which rely on a syntactical bilingual analysis of the sentence-pairs which investigates the structural aspects of the translation process, it is important to take the issues highlighted in this section into consideration. Later in this chapter we do so, by evaluating a grammar design which uses non-terminals which relate to the reordering behaviour of the string-pairs that they cover, propagating reordering decisions across the synchronous derivation.

5.2.3 The Hiero Baseline

The Hiero SMT system (Chiang, 2007) significantly popularised syntax-based MT and remains the yardstick that most other syntax-based models and implementations compare to. Hiero, which we introduce in detail in section 2.4.2, employs an SCFG as the backbone of a log-linear conditional translation model. The SCFG score is combined together with multiple other features ϕ, using weights λ to evaluate the quality of SCFG derivations D employing rules $r \in D$ and leading to translations $\mathbf{e}$ for input $\mathbf{f}$.

$$p(D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle) \propto \phi_{\mathrm{LM}}^{\lambda_{\mathrm{LM}}}(\mathbf{e}) \times \prod_{r \in D} \prod_{i \neq \mathrm{LM}} \phi_i^{\lambda_i}(r) \tag{5.3}$$

The SCFG grammar that Hiero employs treats translation as a hierarchical process, similar to the example of Figure 5.3. Namely, it focuses on lexicalised recursive translation rules, each of which translates a discontiguous source phrase-pair with 'gaps', while at the same time indicating the reordering pattern between the gaps on each side. However, as it covers all such discontiguous phrase-pairs under a single non-terminal X, the grammar employed by Hiero does not offer an abstract recursive explanation of the translation process and remains itself indiscriminate against the strings that each gap will be filled with.

An example of such rules can be seen in Figure 5.6. These rules are in practice allowed to recursively build sentence-segments up to a certain cut-off length (usu-

$$\begin{array}{ll} X \rightarrow & \text{do not } X_{\boxed{1}} \text{ / ne } X_{\boxed{1}} \text{ pas} \\ X \rightarrow & \text{financial } X_{\boxed{1}} \text{ / } X_{\boxed{1}} \text{ économiques} \\ X \rightarrow & \text{this } X_{\boxed{1}}\, X_{\boxed{2}} \text{ / cette } X_{\boxed{1}} \text{ de } X_{\boxed{2}} \\ X \rightarrow & X_{\boxed{1}} \text{ ' s common } X_{\boxed{2}} \text{ policy /} \\ & \text{politique } X_{\boxed{2}} \text{ commune de } X_{\boxed{1}} \end{array}$$

Figure 5.6: Hiero SCFG rules for English and French.

$$\begin{array}{l} S \rightarrow S_{\boxed{1}}\, X_{\boxed{2}} \text{ / } S_{\boxed{1}}\, X_{\boxed{2}} \\ S \rightarrow X_{\boxed{1}} \text{ / } X_{\boxed{1}} \end{array}$$

Figure 5.7: Hiero SCFG glue rules.

ally 10), which are later combined monotonically using the glue rules of Figure 5.7. This constraint together with the complementing features ϕ in the model of equation (5.3), and most importantly the target language model feature $\phi_{\mathrm{LM}}(\mathbf{e})$, aid in avoiding errors due to the absence of a less ambiguous recursive structure than that offered by the rules of Figure 5.6.

The list of features ϕ includes lexical translation scores judging translation on the word level, as well as scores considering the number of words in the target language output and the number of discontiguous phrase-pairs used in the SCFG derivation. Nevertheless, the core modelling elements related to the hierarchical phrase-based interpretation of translation assumed by the model are those employing conditional *discontiguous* phrase translation probabilities. These extend the similar concept of features based on conditional phrase translation probabilities, from the PBSMT models which employ contiguous phrases, to the Hiero models which use phrase-pairs with gaps.

Crucially, these discontiguous phrase translation probabilities for the rules like those in (5.6) are estimated with a heuristic rule of thumb, similarly to how the probabilities for the contiguous phrase-pairs in PBSMT models are set. Namely, they are set based on the extraction counts of contiguous phrase translation patterns from a training word-aligned parallel corpus. These extraction counts are distributed evenly across all discontiguous phrase translation patterns that can be formed by substituting aligned subphrase-pairs of a contiguous phrase-pair for the non-terminal X. The rule weights are computed after normalising these assigned extraction counts for each target part of each rule right-hand side. These scores do provide relatively strong translation performance for some language pairs. However, like their PBSMT analogues, their relation in statistical terms with the training corpus remains obscure.

Even though the impact of depending on surface extraction counts might be limited when computing such estimates for the largely lexically-grounded rules of Hiero, this approach can hardly extend to the estimation of more involved grammars including notions of abstract recursive translation structure. In that case, heuristic estimation would demand counting extraction events on the unobserved latent part of the translation process. This would seem exceedingly arbitrary as the assumed latent structure abstracts more from the observed lexical surface, as we already discussed in section 2.5. While the heuristic estimation of the key parameters of the Hiero translation system might have offered a solid starting point for the emergence of hierarchical translation, it may be a bottleneck in the process of extending syntax-based systems towards grammars abstracting more from the lexical surface.

Zollmann and Venugopal (2006) move in this direction by extending the Hiero system through the introduction of target-side linguistic information in the grammar design along the lines of Figure 5.4. Nevertheless, they also offer no advancements on the learning aspects of the problem, applying instead the same heuristic estimation regime as Hiero, while supporting the simple heuristic estimates with an array of further additional features.

Overall, Hiero introduces the employment of an SCFG as the backbone of a hierarchical translation system focusing on translating with discontiguous phrase-pairs with gaps. However, the SCFG's main contribution in Hiero implementations is to provide hierarchical derivations of target translations which are then scored by a feature-based model, while the ability of stochastic synchronous grammars to function as probabilistic models of translation as in equations (5.1) and (5.2) is not explored. In the rest of this chapter we follow this direction, and consider the *learning* of simple stochastic SCFGs as joint translation models. This features as a crucial intermediate step before moving on to induce the much more intricate linguistically motivated grammars of Chapter 6.

5.2.4 The Learning Problem

The Hiero system exemplifies the gains to be had by combining phrase-based translation (Och and Ney, 2004) with the hierarchical reordering capabilities of SCFGs, particularly originating from the binary Inversion Transduction Grammars (ITG) (Wu, 1997). Yet, the bulk of existing empirical work is largely based on the aforementioned heuristic techniques, and the learning of SCFGs remains an unsolved problem.

The difficulty of this problem stems from the need for simultaneous learning of many kinds of preferences under a single stochastic component.

- The translation of the *lexical* surface, either as part of dedicated phrase-emitting rules as those employed by our grammars later in this chapter or as part of lexicalised reordering rules as in the Hiero SCFGs.
- The *reordering* of phrase spans between the two languages.
- The overall *translation structure* as a mapping between the structure of both sides of the language pair.

The phrase-based analysis of the lexical correspondences between the source and target languages and the modelling of the reordering process have already been addressed in PBSMT models and the related translation systems with relative success, even if it was done only in relation to contiguous string elements. This leaves the modelling of the translation structure as the new exciting element in syntax-based MT. While the concept of a translation structure also includes the reordering patterns between segments of translated sentence-pairs, it moves further than this. It also considers the mapping between abstract syntactic elements of the source and target languages that can aid in explaining the translation process, and which do not necessarily coincide with the syntactic elements of monolingual linguistic analyses.

Crucially, it is exactly this novel aspect of syntax-based MT that learning through rule-of-thumb surface heuristics cannot support, due to the latent nature of translation structure. The approach of Chiang (2005a) however continued to base estimation of model parameters on extraction heuristics, mitigating the related issues by relying on the lexical, observable part of SCFGs, shunning at that time a richer syntactic analysis of the translation process while noting its importance as a future development.

Some efforts to *learn* a synchronous grammar for SMT concentrate on a part of the three translation preferences listed above. The problem of learning the hierarchical, synchronous grammar reordering rules is oftentimes addressed as a learning problem in its own right assuming all the rest is given (Blunsom et al., 2008b). A small number of efforts has been dedicated to the simultaneous learning of the probabilities of phrase translation pairs as well as hierarchical reordering, e.g., (DeNero et al., 2008; Zhang et al., 2008a; Blunsom et al., 2009). Of these, some concentrate on evaluating word-alignment, either directly such as (Zhang et al., 2008a), or indirectly by evaluating a heuristically trained hierarchical translation system from sampled phrasal alignments (Blunsom et al., 2009). However, very few evaluate on actual translation performance of induced synchronous grammars (DeNero et al., 2008). In the majority of cases, the Hiero system, which usually provides the baseline against which hierarchical systems are measured, remains superior in translation performance, see e.g. (DeNero et al., 2008).

5.3 Synchronous Grammar Learning

In the rest of the chapter, we tackle the problem of learning *generative phrase-based ITG models* as translation models assuming latent phrase segmentation and latent reordering: this setting is most similar to the training of Hiero. Unlike all other work that heuristically selects a subset of phrase-pairs, we start out from an SCFG that works with *all* phrase-pairs in the training set and concentrate on the aspects of learning. This problem is fraught with the risks of overfitting and can easily result in inadequate reordering preferences (DeNero et al., 2006).

We find that the translation performance of all-phrase probabilistic SCFGs induced in this setting crucially depends on the interplay between two aspects of learning:

- Defining a more constrained parameter space, where the reordering productions are phrase-lexicalised and made sensitive to neighbouring reorderings.
- Defining an objective function that effectively smoothes the maximum-likelihood criterion.

One of our contributions is in deploying the Cross-Validated EM algorithm implementing an effective, data-driven smoothed Maximum-Likelihood, which can cope with a model working with *all* phrase-pair SCFGs, building upon the work presented in Chapter 4. However, on top of the challenges already discussed there in the context of the application of CV-EM on PBSMTs, learning SCFGs poses significant novel challenges, the core of which lies in the hierarchical nature of a stochastic SCFG translation model and the relevant additional layer of latent structure. We address these issues in this chapter. Another important contribution is in defining a lexicalised reordering component within ITG that captures order divergences orthogonal to those tracked by the Hiero SCFG, but somewhat akin to PBSMT 'monotone-swap-discontinuous' reordering models (Tillman, 2004). Our best system exhibits Hiero-level performance on French-English Europarl data using an SCFG-based decoder. Our findings should be insightful for others attempting to make the leap from shallow phrase-based systems to hierarchical SCFG-based translation models that use learning methods, as opposed to heuristics.

5.3.1 Fragment Modelling Aspects

The Synchronous Context-Free Grammars which we consider here, both in the case of the Hiero baseline as well as for our own grammar designs presented in the next section, are phrase-based SCFGs. Unlike the Inversion Transduction Grammar as it was originally introduced as a word-based model in (Wu, 1997), these grammars allow synchronous rules with right-hand sides which include a lexicalised part of arbitrary length. Such rules can describe contiguous or discontiguous aligned sentence-pair *fragments* of the training word-aligned parallel corpus. Under the assumption that we do not impose any arbitrary constraints on the length of these lexicalised rule segments, such phrase-based SCFGs can then be categorised under the Fragment Model family. In this context, the abstract part of the SCFG together with the rule probabilities provide the necessary stochastic generative machinery combining the lexical (dis)contiguous fragments to form sentence-pairs.

As in the case of PBSMT, this powerful modelling feature exposes the learning of these grammars under a Maximum Likelihood objective to the same overfitting issues as other *all-fragment models* such as Phrase-Based SMT (Marcu and Wong, 2002; DeNero et al., 2006) and Data-Oriented Parsing (Bod et al., 2003; Zollmann and Sima'an, 2006). Maximum Likelihood Estimation (MLE) returns degenerate grammar estimates that memorise well the parallel training corpus but generalise poorly to unseen data. As for the other fragment models, also in the case of SCFGs, this overfitting tendency leads towards an MLE estimate which effectively memorises whole sentence-pairs, using merely a trivial abstract structure leading directly towards the emission of whole training sentence-pairs, like for example $S \rightarrow X \rightarrow \langle \mathbf{e}, \mathbf{f} \rangle$.

Such degenerate MLE estimates essentially memorise the empirical frequency of sentence-pairs in the parallel corpus but generalise extremely poorly as they also predict nothing more past what is included in the training data, as explained in section 3.1.5. The failure of straightforward applications of MLE to arrive at estimates which generalise well can be also attributed to a trade-off effect on the bias-variance decomposition of the expected Generalisation Error. The zero GE due to estimator bias is counter-balanced by a very high GE due to estimate variance, as we discuss for Fragment Models in general in section 3.1.6.

Independently of the aspect that it is being considered, the overfitting tendency of MLE estimators is encumbering the learning of SCFGs in all the aspects of the translation process that they are modelling, posing further challenges than those encountered while learning PBSMT phrase-table parameters in Chapter 4. On one hand, similarly to the estimation of PBSMT models, it does not allow us to identify and shift probability mass towards reusable lexical fragments. In the case of SCFGs however, this is complemented by the inability to learn any non-trivial translation structure, as the MLE solution overfits towards the minimal syntactic elements necessary to construct sentences from the largest memorised bilingual fragments. In order for the learning of SCFGs under a likelihood optimisation objective to arrive at any meaningful results, both in terms of reusable lexical components as well as abstract syntactic constructions, this strong tendency of the MLE estimator to memorise the training parallel corpus must first be addressed.

```
INPUT: Word-aligned parallel training data $\mathcal{X}$
       Grammar extractor $G$
       The number of parts $J$ to partition $\mathcal{X}$
OUTPUT: SCFG $\mathbf{G}$ with estimates $\hat{\theta}^{CV} = \{p(r)\}$ for all grammar rules $r$

Partition training data in $J$ equal parts $\mathcal{X}^1, \ldots, \mathcal{X}^J$
For $1 \leq j \leq J$ do
    Extract grammar rules set $\mathbf{G}_j = G(\mathcal{X}^j)$
Initialise $\mathbf{G} = \cup_j \mathbf{G}_j$, $\hat{\theta}_0^{CV} = \{p_0(r) : r \in \mathbf{G}\}$ uniform per rule LHS
Let $r = 0$      // EM iteration counter
Repeat
   Let $r = r + 1$
   E-step:
      For $1 \leq j \leq J$ do
           Calculate expected counts given $\mathbf{G}$, $\hat{\theta}_{r-1}^{CV}$,
              for derivations $D^{-j}$ of $\mathcal{X}^j$
              using rules from $\cup_{k \neq j} \mathbf{G}_k$
   M-step: set $\hat{\theta}_r^{CV}$ to ML estimate given expected counts
Until $\hat{\theta}_r^{CV}$ has converged
```

Figure 5.8: The CV Expectation-Maximization algorithm for SCFG learning.

5.3.2 CV-EM SCFG Estimation

In order to avoid the overfitting solution of plain MLE, we opt instead for a Cross-Validated MLE learning objective, which we implement using the Cross-Validated EM algorithm presented in section 3.2. Here we use Cross-Validation to leverage the bias-variance trade-off for learning stochastic all-phrase SCFGs. Given an input all-phrase SCFG grammar with phrase-pairs extracted from the training data, we maximise training data likelihood subject to CV smoothing. Splitting the word-aligned parallel training data $\mathcal{X}$ in J roughly equally-sized parts $\mathcal{X}^1, \ldots, \mathcal{X}^J$, for each data part $\mathcal{X}^j$ we consider only derivations $\mathbf{D}^{-j}$ which employ grammar rules extracted from the rest of the data $\mathcal{X}^{-j}$. An essential part then of the learning process involves choosing the grammar *extractor* $G(\mathcal{X})$, a function from data to an all-phrase SCFG under a particular grammar design, which we discuss in section 5.4 below.

A summary of the CV-EM algorithm for the learning of SCFG joint translation models such as those of equations (5.1) and (5.2) can be seen in Figure 5.8. As in all applications of CV-EM, being an EM instance guarantees convergence and a non-decreasing CV-smoothed training data likelihood after each iteration. Our practical implementation is based on a synchronous version of the Inside-Outside algorithm. This takes care during the E-step of the efficient computation of expected counts of rule applications in derivations according to the current parameter set and is a straightforward adaptation of the monolingual version, considering bilingual instead of monolingual spans. The running time is $O(n^6)$, where n is the input's length, but by considering only derivation spans which do not cross word-alignment points, our implementation runs in reasonable times for relatively large corpora.

Beside being an estimator of the SCFG probability parameter set $\hat{\theta}$, the CV-MLE learning algorithm has the added value of being a grammar learner focusing on reducing generalisation error, in the sense that probabilities of grammar productions should reflect the frequency with which these productions are expected to be used for translating future data. Since the CV criterion prohibits for every data point derivations that use rules that can only be extracted from the same data part, such rules are assigned zero probabilities in the final estimate and are effectively excluded from the grammar. In this way, the algorithm 'shapes' the input grammar, concentrating probability mass on productions that are likely to be used with future data.

In this chapter we do not pursue the use of grammar extractors outputting complex abstract structures, even though we do move further than a plain Hiero-like grammar totally lacking this aspect. For this reason, the effect of restricting the grammar mentioned above relates more to the lexical part of the grammar designs that we experiment with. Nevertheless, concentrating on lexical units which are expected to generalise by applying CV-smoothing on the lexical level is crucial in allowing us to estimate the parameters related to the higher-level syntactical components, as well as to learn how to combine these reusable lexical building blocks together.

The number of abstract syntactic rules used in the grammars that we present below is limited and their design is a generic one without any reference to the training data. This allows us to consider these as included in every extracted rule-set $G(\mathcal{X}^j)$ and allow them to survive the CV-smoothing in their entirety. However, as we increase the complexity of the higher-level syntax in the synchronous grammars that we consider and *especially* if this part of the grammar is constructed in reference to the training data, we believe it is important to also address the possible overfitting of the abstract part of the grammar. We consider this issue in Chapter 6.

5.3.3 Smoothing the Model

The practical application of CV-EM for SCFGs also demands the treatment of boundary cases. There will often be sentence-pairs in $\mathcal{X}^j$, that cannot be fully derived by the grammar extracted from the rest of the data $\mathcal{X}^{-j}$. The reason might be: (a) 'unknown' words (i.e. not appearing in other parts of the CV partition) or (b) complicated combinations of adjacent word-alignments. To address this, we employ external smoothing of the grammar, *prior* to learning.

Our solution is to extend the SCFG extracted from $\mathcal{X}^{-j}$ with new emission productions deriving the 'unknown' phrase-pairs (i.e., found in $\mathcal{X}^{j}$ but not in $\mathcal{X}^{-j}$). Crucially, the probabilities of these productions are drawn from a *fixed* smoothing distribution, i.e. they remain constant throughout estimation. Our smoothing distribution of phrase-pairs $\langle\tilde{e},\tilde{f}\rangle$ for all pre-terminals considers source-target phrase lengths drawn from a Poisson distribution with unit mean, drawing subsequently the words of each of the phrases uniformly from the vocabulary of each language, similar to (Blunsom et al., 2009).

$$p_{smooth}(\langle\tilde{e},\tilde{f}\rangle) = \frac{p_{poisson}(|\tilde{f}|;1)\ p_{poisson}(|\tilde{e}|;1)}{V_f^{|\tilde{f}|}\ V_e^{|\tilde{e}|}} \tag{5.4}$$

Since the smoothing distribution puts stronger preference on shorter phrase-pairs and avoids competing with the 'known' phrase-pairs, it leads the learner to prefer using as little as possible such smoothing rules, covering only the phrase-pairs required to complete full derivations.

5.4 Parameter Spaces and Grammar Extractors

As we discussed in 5.2.1, the translation structure is a latent modelling component and an MT practitioner is free to consider it from different perspectives, which may be based on machine learning, linguistic or cognitive grounds. However in the end the synchronous structure is primarily judged empirically, based on the ability to more closely capture the translation process and lead us towards better translations. In the context of the learning framework presented in the previous section, a crucial modelling choice is then establishing the space of latent synchronous grammatical constructions that our learner will consider against the empirical observations in the training data.

In our SCFG learning pipeline, the decisions related to the synchronous grammar design are encoded in the *Grammar Extractor* (GE). A GE is a function from a word-aligned parallel corpus to a set of Synchronous Context-Free Grammar rules. Together with the constraints that render a proper joint probabilistic SCFG, i.e. the sum of probabilities for productions that have the same left-hand side must be one, the GE also serves to define the parameter space of the stochastic model that we establish by extending every rule in the output of the GE with a probability.

The Grammar Extractors used in this chapter create SCFGs productions of two different kinds:

1. Abstract hierarchical synchronous productions that define the space of possible derivations up to the level of SCFG *pre-terminals*

2. The phrase-pair emission rules that expand the pre-terminals to phrase-pairs of varying lengths.

Start $S \rightarrow X \ / \ X$

Monotone Expansion $X \rightarrow X_{\boxed{1}} \, X_{\boxed{2}} \, / X_{\boxed{1}} \, X_{\boxed{2}}$

Switching Expansion $X \rightarrow X_{\boxed{1}} \, X_{\boxed{2}} \, / X_{\boxed{2}} \, X_{\boxed{1}}$

Phrase-Pair Emission $X \rightarrow \tilde{e} \ / \ \tilde{f}$

Figure 5.9: Single Phrase-Pair NT Grammar.

Computing the GE's output begins by extracting phrase-pair emitting rules for the set of *all* translational equivalents (without length upper-bound) abiding to the word-alignment, according to the rules of (Och and Ney, 2004; Koehn et al., 2003). These phrase-pair emitting rules are complemented by the abstract translation structure rules that cover the distance between the start symbol and the phrase-pairs. However, while the phrase-pairs that will be the right-hand sides of the phrase-pair emission rules depend on the parallel corpus, we cannot extract translation structure rules from it as the latter is not labelled with a synchronous parse. For this, the translation structure part of the grammar output of the GEs that we examine in this chapter, does not depend on their input. Since these rules will be present in the SCFGs extracted from all cross-validation parts, the CV-EM learning algorithm implementation of Figure 5.8 cannot protect against overfitting caused by this part of the grammars. For this reason, for this first examination of SCFG learning with CV-EM discussed in this chapter, we have elected to employ relatively simple translation structures, to mitigate the risk of overfitting due to over-specialised abstract structure rules.

Below we present the two grammar extractors employed in our experiments. We start out from the most generic, ITG-like formulation, and aim at incremental refinement of the hierarchical productions in order to capture relevant, content-based phrase-pair reordering preferences in the training data.

5.4.1 Single Phrase-Pair NT SCFG

This is a phrase-based binary SCFG grammar employing a single non-terminal X covering each extracted phrase-pair. The other two productions consist of monotone and switching expansions of phrase-pair spans covered by X. Finally, the whole sentence-pair is considered to be covered by X. We will call this the 'plain SCFG' extractor and the simple abstract translation structure that it produces serves as a baseline against which more elaborate grammar designs can be empirically compared. The SCFG produced by the plain SCFG extractor, given an input corpus $\mathcal{X}$ from which phrase-pairs $\langle \tilde{e}, \tilde{f} \rangle$ can be extracted, is listed in Figure 5.9.

Start $S \rightarrow X_{\boxed{1}} / X_{\boxed{1}}$

Monotone Expansion

$$X \rightarrow X_{\boxed{1}} X_{\boxed{2}} / X_{\boxed{1}} X_{\boxed{2}}$$
$$X^{\mathbf{L}} \rightarrow X_{\boxed{1}} X_{\boxed{2}} / X_{\boxed{1}} X_{\boxed{2}}$$
$$X^{\mathbf{R}} \rightarrow X_{\boxed{1}} X_{\boxed{2}} / X_{\boxed{1}} X_{\boxed{2}}$$

Switching Expansion

$$X \rightarrow X^{\mathbf{L}}_{\boxed{1}} X^{\mathbf{R}}_{\boxed{2}} / X^{\mathbf{R}}_{\boxed{2}} X^{\mathbf{L}}_{\boxed{1}}$$
$$X^{\mathbf{L}} \rightarrow X^{\mathbf{L}}_{\boxed{1}} X^{\mathbf{R}}_{\boxed{2}} / X^{\mathbf{R}}_{\boxed{2}} X^{\mathbf{L}}_{\boxed{1}}$$
$$X^{\mathbf{R}} \rightarrow X^{\mathbf{L}}_{\boxed{1}} X^{\mathbf{R}}_{\boxed{2}} / X^{\mathbf{R}}_{\boxed{2}} X^{\mathbf{L}}_{\boxed{1}}$$

Phrase-Pair Emission

$$X \rightarrow \tilde{e} \,/\, \tilde{f}$$
$$X^{\mathbf{L}} \rightarrow \tilde{e} \,/\, \tilde{f}$$
$$X^{\mathbf{R}} \rightarrow \tilde{e} \,/\, \tilde{f}$$

Figure 5.10: Lexicalised-Reordering SCFG

5.4.2 Lexicalised Reordering SCFG

One weakness of the ‘plain SCFG’ is that the reordering decisions in the derivations are made without reference to lexical content of the phrases; this is because all phrase-pairs are covered by the same non-terminal. As a refinement, we propose a grammar extractor that aims at modelling the reordering behaviour of phrase-pairs by taking their content into account. This time, the X non-terminal is reserved for phrase-pairs and spans which will take part in monotonic productions only. Two fresh non-terminals, $X^{\mathbf{L}}$ and $X^{\mathbf{R}}$, are used for covering phrase-pairs that participate in order switching reordering operations with other, adjacent phrase-pairs. The non-terminal $X^{\mathbf{L}}$ covers phrase-pairs which appear first in the source language order, and the latter those which follow them. The grammar rules produced by this GE, dubbed ‘switch grammar’, are listed in Figure 5.10.

The reordering information captured by the switch grammar is in a sense orthogonal to that of Hiero-like systems utilising rules such as those listed in Figure 5.6. Hiero rules encode hierarchical reordering patterns based on surrounding context. In contrast, the switch grammar models the reordering preferences of the phrase-pairs themselves, similarly to the monotone-swap-discontinuous reordering models of Phrase-based SMT models (Tillman, 2004). On top of that, it strives to match pairs of such preferences, combining together phrase-pairs with compatible reordering preferences, as well as conditioning the production of every

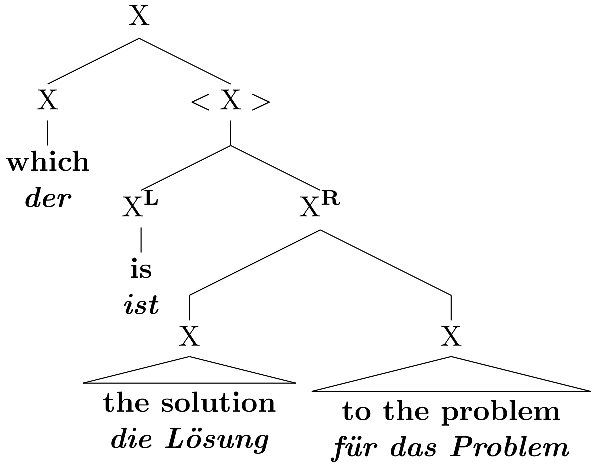


Figure 5.11: A sub-tree covering a secondary clause between English and German using the switch SCFG. $< >$ indicates a switch reordering operation between the two children of the non-terminal. The application of the rule $X \rightarrow X^{\mathbf{L}}\ X^{\mathbf{R}}$ indicates that the two children (verb and noun phrase) must switch as we translate between the two languages, while the resulting verb phrase combines monotonically with the context on its left.

non-terminal to the reordering behaviour of the span covered by it. In this way, it addresses the modelling pitfalls described in section 5.2.2. Now the reordering choices of every synchronous derivation expansion are affected both by the preferences of the children as well as the parents of every node in the derivation tree, as encoded by the three specialised non-terminals present in the left and right-hand side of every production rule.

An example of a derivation subtree for a secondary clause between English and German can be seen in Figure 5.11. For the switch SCFG that we employ in this chapter, while the form of the abstract structure in the example can be explained in linguistic terms, identifying it past the pre-terminals makes use of a small set of generic rules and their probabilities, which together represent the *overall* reordering behaviour of synchronous spans across the training corpus. In Chapter 6 we will enrich the synchronous grammatical constructions to explicitly condition such reordering operations on linguistic cues.

5.5 Experiments

Pairing each Grammar Extractor with the CV-EM implementation of section 5.3.2 allows us to learn probabilistic Synchronous Context-Free Grammars and estimate their parameters from training word-aligned parallel corpora. In this section we proceed to integrate these synchronous grammars within an SCFG-based decoder. We subsequently evaluate our performance in relation to the state-of-the-art Hiero baseline of section 5.2.3 on a French to English translation task.

5.5.1 Decoding

The joint model of bilingual string derivations provided by the learnt SCFG grammar can be used for translation given a input source sentence, since:

$$\arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} p(\mathbf{e}, \mathbf{f})$$

We use our learnt stochastic SCFG grammar with the decoding component of the Joshua SCFG toolkit (Li et al., 2009). The full translation model interpolates log-linearly the probability of a grammar derivation together with the language model probability of the target string. The model is further smoothed, similarly to phrase-based models and the Hiero system, with smoothing features ϕ such as the lexical translation scores of the phrase-pairs involved and rule usage penalties in the same way as our baseline. As usual with statistical translation, we aim for retrieving the target sentence $\mathbf{e}$ corresponding to the most probable derivation $D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle$ for the source side $\mathbf{f}$ making use of rules r, with:

$$p(D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle) \propto \phi_{\mathrm{LM}}^{\lambda_{\mathrm{LM}}}(\mathbf{e})\, p_{\mathrm{SCFG}}(\mathbf{e}, \mathbf{f})^{\lambda_{\mathrm{SCFG}}} \times \prod_{r \in D} \prod_{i \neq \mathrm{LM,SCFG}} \phi_i^{\lambda_i}(r) \tag{5.5}$$

The interpolation weights are tuned using Minimum Error Rate Training (Och, 2003).

5.5.2 Results

We test empirically the learner's output grammars for translating from French to English, using $J = 5$ for the Cross-Validation data partitioning. The training material is a GIZA++ word-aligned corpus of 200K sentence-pairs from the Europarl corpus (Koehn, 2005), with our development and test parallel corpora of 2K sentence-pairs stemming from the same source. Training the grammar parameters until convergence demands around 6 hours on an 8-core 2.26 GHz Intel Xeon system. Decoding employs a 4-gram language model, trained on English Europarl data of 19.5M words smoothed using modified Kneser-Ney discounting (Chen and Goodman, 1998), and lexical translation smoothing features based on the GIZA++ alignments.

In a sense, from a learning perspective the real baseline that we might compare against should be a system employing the plain MLE estimate for the grammar extracted from the whole training corpus. However, as we have already discussed, this assigns zero probability to all sentence-pairs outside of the training data and is subsequently bound to perform extremely poorly, as decoding would then completely rely on the smoothing features. In addition, we cannot directly compare the CV-EM estimates for the plain and switch SCFGs against estimates that are heuristically trained similarly to those employed in Hiero or PBSMT, as it is not clear how the reordering rule probabilities of a grammar similar to the ones we use could be trained heuristically based on extraction counts, given that the relevant structure is unobserved.

Instead, we opt to compare against a hierarchical translation baseline provided by the Joshua toolkit, trained and tuned on the same data as our learning algorithm. The grammar used by the baseline is much richer than the ones learnt by our algorithm, also employing rules which translate with context, as discussed in section 5.2.3. However, the baseline does not make use of abstract translation rules without a lexical part, relying on the glue grammar of Figure 5.7 to monotonically combine the discontiguous phrase-pairs together after they have been recursively expanded. Relating the performance of our learnt stochastic SCFG grammars to a hierarchical translation baseline such as this, has the added advantage of comparing against a system which remains in the state-of-the-art of SCFG-based translation, evaluating the potential of our approach to deliver real-world competitive translation performance.

System	Lexical Smoothing	BLEU
joshua-baseline	No	27.79
plain scfg	No	28.04
switch scfg	No	28.48
joshua-baseline	Yes	29.96
plain scfg	Yes	29.75
switch scfg	Yes	29.88

Table 5.1: Empirical results, with and without additional lexical translation smoothing features during decoding

Table 5.1 presents the translation performance results of our systems and the baseline. On first observation, it is evident that our learning algorithm outputs stochastic SCFGs which manage to generalise, avoiding the degenerate behaviour of plain MLE training for these models. Given the notoriety of the estimation process, this is noteworthy on its own. Having a learning algorithm at hand which realises to a reasonable extent the potential of each stochastic grammar design (as implemented in the relevant grammar extractors), we can now compare between the two grammar extractors used in our experiments. The results table highlights the importance of conditioning the reordering process on lexical grounds. The plain grammar with the single phrase-pair non-terminal cannot accomplish this and achieves a lower BLEU score. On the other hand, the switch SCFG allows such conditioning. The learner takes advantage of this feature to output a grammar which performs better in taking reordering decisions, something that is reflected in both the actual translations as well as the BLEU score achieved.

Furthermore, our results highlight the importance of the additional smoothing decoding features of equation (5.5). The unsmoothed baseline system itself scores considerably less when employing solely the heuristic translation score. Our unsmoothed switch grammar decoding setup improves on the baseline by a considerable difference of 0.7 BLEU, highlighting the reliance of the heuristic estimates on these additional smoothing features to provide reasonable translations. Subsequently, when adding the smoothing lexical translation features, both systems record a significant increase in performance, reaching comparable levels of performance.

The degenerate behaviour of MLE for SCFGs can be greatly limited by constraining ourselves to grammars employing *minimal* phrase-pairs: phrase-pairs which cannot be further broken down into smaller ones according to the word-alignment. One could argue that it is enough to perform plain MLE with such minimal phrase-pair SCFGs, instead of using our more elaborate learning algorithm with phrase-pairs of all lengths. To investigate this, for our final experiment we used a plain MLE estimate of the switch grammar to translate, limiting the grammar's phrase-pair emission rules to only those which involve minimal phrase-pairs. The very low score of 17.82 BLEU (without lexical smoothing) not only highlights the performance gains of using longer phrase-pairs in hierarchical translation models, but most importantly provides a strong incentive to address the overfitting behaviour of MLE estimators for such models, instead of avoiding it.

5.6 Related Work

Most learning of phrase-based models, e.g. (Marcu and Wong, 2002; DeNero et al., 2006) and the work presented in Chapter 4, works without hierarchical components such as those employed by ITG/SCFG grammars. These learning problems pose other kinds of learning challenges than the ones presented by the explicit learning of SCFGs. While Chiang's original work (Chiang, 2005a; Chiang, 2007) introduces a particular flavour of phrase-based binary synchronous grammars, his learning approach keeps almost intact the heuristic estimation of PBSMT. The learning problem is not expressed in terms of an explicit objective function and surface heuristic counts are used instead. Nevertheless, it has been very difficult to match the performance of Hiero-like models without use of these heuristic counts.

A somewhat related work, (Blunsom et al., 2008b), attempts learning new non-terminal labels for synchronous productions in order to improve translation. This work differs substantially from our work because it employs a heuristic estimate for the phrase pair probabilities, thereby concentrating on a different learning problem: that of refining the grammar symbols. Our approach might also benefit from such a refinement but we do not attempt this problem here. In contrast, (Blunsom et al., 2008a) works with the expanded phrase pair set of (Chiang, 2005a), formulating an exponential model and concentrating on marginalising out the latent segmentation variables. Again, the learning problem is rather different from ours. Similarly, the work in (Zhang et al., 2008a) reports on a multi-stage model, *without* a latent segmentation variable, but with a strong prior preferring sparse estimates embedded in a Variational Bayes (VB) estimator. This work concentrates the efforts on pruning both the space of phrase pairs and the space of (ITG) analyses.

To the best of our knowledge, the work presented in this chapter based on the results of (Mylonakis and Sima'an, 2010) was the first to attempt learning probabilistic phrase-based binary SCFGs as translation models, in a setting where both a phrase segmentation component and a hierarchical reordering component are assumed as latent variables. Like our approach, (DeNero et al., 2008) also employ an all-phrases model, however the work presented here complements the results of Chapter 4 in showing that it is possible to train such large-scale grammars under iterative algorithms like CV-EM, without need for sampling or pruning.

5.7 Discussion

Phrase-based stochastic SCFGs provide a rich formalism to express translation phenomena, which has been shown to offer competitive performance in practice. Since learning SCFGs for machine translation has proven notoriously difficult, most successful SCFG models for SMT rely on rules extracted from word-alignment patterns and heuristically computed rule scores, with the impact and the limitations imposed by these choices yet unknown.

Some of the reasons behind the challenges of SCFG learning can be traced back to the introduction of latent variables at different, competing levels: word and phrase-alignment used side by side with hierarchical reordering structure, with larger phrase-pairs reducing the need for extensive reordering structure and vice versa. While imposing priors such as the often used Dirichlet distribution or the Dirichlet Process provides a method to overcome these pitfalls, we believe that the data-driven CV-MLE learning objective and the CV-EM algorithm employed in this chapter provide an effective alternative to them, focusing more on the data instead of importing generic external human knowledge. Our use of CV-EM to learn Synchronous CFGs adds additional evidence to the effectiveness of our algorithm to train models assuming increasingly complex latent variables, moving from the flat segmentation variables of Chapter 4 to the recursive structures of SCFGs.

We believe that the work in this chapter makes a significant step towards learning synchronous grammars for SMT. This is an objective not only worthy because of promises of increased performance, but, most importantly, also because it increases the depth of our understanding of SCFGs as vehicles of latent translation structures. Our usage of the induced grammars directly for translation, instead of an intermediate task such as phrase-alignment, aims exactly at this.

While the latent structures that we explored here were relatively simple in comparison with Hiero-like SCFGs, they take a different, content-driven approach to learning reordering preferences, rather than the context-driven approach of Hiero. We believe that overall these approaches are not merely orthogonal, but could also prove complementary. Taking advantage of the possible synergies between content and context-driven reordering learning is an appealing direction of future research stemming from this thesis. This is particularly promising for other language pairs, such as Chinese to English, where Hiero-like grammars have been shown to perform particularly well.

In the following chapter, we build on the intuitions gained and the results presented above to learn a translation model employing a rich, linguistically motivated latent structure. This moves further than synchronous grammars which use a handful of abstract categories to describe the translation process, like those we employed here. Even though we proceed towards using grammars taking advantage of hundreds of thousands of abstract categories, we retain the design principles behind the ‘switch SCFG’ presented here. We use its ability to facilitate learning how to combine together the reordering preferences of phrase-pairs and those of abstract categories, within a translation structure robust enough to cover whole sentence-pairs. In this way, the successful deployment of CV-EM as a learning algorithm for the somewhat simpler SCFGs presented in this chapter, as well as our experimentation with different synchronous grammar design principles, pave the way for the work that follows.

Chapter 6

Learning Linguistically Motivated Latent Translation Structure

Research efforts towards making use of the syntactic aspects of translation have been intensified during the past decade. We have already witnessed in this thesis the build-up from earlier work on translation formalisms driven by formal syntax such as (Aho and Ullman, 1969; Lewis and Stearns, 1968), to Wu's introduction of the Inversion-Transduction Grammar (ITG) (Wu, 1997). The latter, a subset of the Synchronous Context-Free Grammars (SCFGs), seems to combine the merits of simplicity and computational efficiency with the ability to describe a multitude of frequently occurring translation phenomena. Chiang (2005a) moved further by combining in the Hiero system the hierarchical nature of the ITG with the modelling potential of Phrase-Based Statistical Machine Translation (PB-SMT), to model translation as a hierarchical process which recursively expands discontiguous phrase-pairs.

Chiang in the Hiero system itself did not explore the potential of SCFGs to describe translation in terms of an *abstract* hierarchical process, choosing as a first step to focus on the ability of the SCFG formalism to provide the mechanics for a model employing discontiguous phrase-pairs extracted from a parallel corpus. Nevertheless, its empirical success in translating between languages with significant syntactic differences such as English and Chinese, triggered a barrage of work on syntax-aware translation such as (Zollmann and Venugopal, 2006; Liu et al., 2006; Chiang, 2010), a substantial part of which focuses on SCFG-based approaches. Many of these approaches aimed to relate the structural aspects of the translation process to the linguistic syntax of the source and/or target language, as we discuss in more detail in the next section.

The positive results of these approaches exemplified the gains to be had from incorporating linguistic syntax elements in a translation system. Overall, it is widely recognised that many translation phenomena correlate with linguistic structures, and the relative success of work such as that mentioned in the previous paragraph provides further empirical evidence on this issue. However, as showcased by the disappointing early results from seemingly straightforward approaches to do so (see e.g. (Koehn et al., 2003)), taking advantage of linguistic annotations is a non-trivial problem that remains open and still strongly attracts the interest of MT researchers today.

As we discussed in Chapter 5, the transition from phrase-based to hierarchical SMT already marked a significant increase in the complexity of the latent variables included in the relevant models. In the general case, training a hierarchical model involves learning a complex structural hidden variable with a recursive nature, which the latent variables of the phrase-based models miss. This significantly increases the stress on the learning components of hierarchical SMT systems, as the difficulty of inducing translation structure increases as we move further away from the observed lexical surface.

The initial introduction of phrase-based SCFGs for SMT in the form of the Hiero system by (Chiang, 2005a) did not address this Machine Learning challenge, choosing instead to employ heavily lexicalised synchronous productions, all but completely avoiding abstract translation structure. This allowed Hiero to be trained under the same extraction heuristics as PBSMT, based on counting the number of extractions of discontiguous bilingual patterns with ‘gaps’ in the same way as PBSMT estimates are based on counts of contiguous phrase-pairs. This estimator, together with the related PBSMT estimator, is a heuristic one, given that it is not known to optimise any objective function of the training data, as well as because the extraction counts that it uses are not related to any observable events in the data: we know that these discontiguous bilingual patterns appear in the data but we neither know nor make an effort to find out how they participate in data constructions. Still, making sure that synchronous productions are grounded by lexical context, avoiding abstract productions and employing no more than a pair of synchronous non-terminals, together with the support provided by additional smoothing features, allows Hiero-like systems to provide state-of-the-art performance for some language pairs.

Later systems such as (Zollmann and Venugopal, 2006; Liu et al., 2006; Liu et al., 2009) moved further on the path laid by Hiero. These systems made use of linguistically motivated abstract categories and a hierarchical translation structure which reaches higher-up from the lexical surface, by means of recursive translation patterns extracted from the monolingual syntactic parse trees of the source and/or target training sentences. Crucially, the trend to employ heuristics based on bilingual pattern extraction counts to estimate the parameters of SMT models was also extended to translation models assuming a richer hierarchical structure, something that applies to all of the three aforementioned systems. This choice was possibly made in connection with the difficulties of matching state-of-the-art performance by training syntactic SMT models using better-founded estimation approaches, such as the Expectation-Maximization algorithm (Galley et al., 2006) or Bayesian inference (DeNero et al., 2008).

The considerable progress in the direction of employing linguistic syntax in the context of hierarchical SMT models has led to systems providing state-of-the-art performance for many language pairs, especially those with significant syntactic divergence. Still, there are significant remaining challenges for syntax-based SMT. Although many of these are related to the additional technical and computational challenges related to training and decoding that must be tackled, we believe that a considerable bottleneck preventing a breakthrough is the way these methods approach learning hierarchical models from data. Learning based on heuristics becomes increasingly arbitrary as we move further up from the observed lexical surface, especially as the recursive building blocks employed become less lexicalised and more abstract in nature. This is not an issue of a purely theoretical nature, as it prevents hierarchical SMT systems from realising the full descriptive and modelling potential of synchronous grammars and the rest of the bilingual syntax formalisms. Practical implementations are forced to commit to compromises regarding the grammar families and rules they exploit, so that the overall model still delivers reasonable performance when trained by the heuristic rules.

There is another problem with the use of linguistic syntax in a rigid, heuristic scheme without a clear *translation-centric* learning objective: we run the risk of imposing unnecessary linguistic constraints, which might have little to do with translation and lead to sub-optimal system output. While some of the bilingual, linguistic patterns that these systems extract do constitute useful translation building blocks, avoiding to ascertain which of these are actually relevant for translation and how they can be combined together means that one may end up with a model which fails to generalise. Instead, we believe we should be aiming at models which are not linguistically constrained but linguistically *motivated*, by learning to take advantage of only those linguistic cues which help translate better and through determining how to combine together the syntax-based translation building blocks in an effective ensemble.

In this chapter, we build upon the theoretical discussion and the algorithms presented in Chapter 3, as well as the empirical findings on applying these methods to learn translation models provided in Chapters 4 and 5. We use these to formulate a method to learn linguistically motivated hierarchical translation models, based on our results first published in (Mylonakis and Sima'an, 2011).

Our efforts are concentrated on learning to take advantage of the interplay between monolingual structure, which can be considered observed when employing a syntactic parser, and the hidden, bilingual translation structure we must induce. The result is a learning approach that aims at discovering effective SCFG-based models making use of a linguistically-aware abstract hierarchical translation structure, which focuses on only those syntactic cues which are found to benefit translation.

We do this by optimising a clear, bilingual learning objective based on Cross-Validated MLE (CV-MLE), promoting the generalisation capacity of the models. This objective allows us to induce a probabilistic abstract translation structure which is empirically shown to robustly and effectively capture the recursive nature of translation across whole sentence-pairs. We show that this provides significantly improved translation output on a range of language pairs in comparison with a state-of-the-art hierarchical translation baseline.

6.1 Linguistically Aware Hierarchical SMT

The efforts to utilise linguistic syntax for Statistical Machine Translation created a new link between current MT research and its historical roots. In the early days of MT, it was thought that the translation from one language to another could be described in terms of a set of rules employing abstract categories, which were often related to syntactic elements. However, the endeavour to manually compile and orchestrate together such sets of rules reached its limits well before producing reasonable translation quality on source language domains which were not heavily restricted. It was gradually recognised that the level of complexity of cross-language communication rendered this effort extremely difficult. The breakthrough on MT initiated by the work on the IBM models (Brown et al., 1990), brought a surge of research activity on overcoming the limitations of rule-based systems through learning *lexical* bilingual correspondences between the source and target languages, by using corpora of already translated text to build probabilistic translation models.

Either to keep the learning and engineering challenges manageable, or perhaps so as to distance itself from their rule-based predecessors, SMT approaches originally mostly stayed close to the lexical surface, employing word or phrase-based models directly translating lexical units from source to target. Lexical SMT succeeded in making a clear step forward both in terms of translation performance and language-pair coverage, as well as by bringing Machine Translation research back to the spotlight.

Still, SMT systems employing shallow, mostly lexical translation models are affected by the sparse nature of natural language. These concerns led towards a resurgence of *syntactic* MT approaches. Syntax-driven SMT employs bilingual rules making use of linguistic or other abstract categories to explain translation as a recursive process, aiming to generalise better past the training data. The crucial new ingredients which made feasible what in the past was considered highly challenging were the advancements in probabilistic models and statistical learning for Machine Translation.

These efforts have been directed towards SMT models employing linguistic syntax on either the source or target side of the translation process, or even across both languages. Each approach enjoys its own strengths and weaknesses.

- Using *source*-side syntax (Quirk et al., 2005; Liu et al., 2006) allows a model to condition translation operations, such as lexical choice and reordering, on the linguistic structure of the source sentence. Since the source sentence is the fixed input of a translation system, such a 'tree to string' translation system may condition translation decisions upon a single source parse tree generated with a natural language parser. In this way, we avoid the additional task of disambiguating over the translation hypothesis subspace related to monolingual linguistic syntax.

- Approaches utilising *target*-side syntax (Yamada and Knight, 2001; Yamada and Knight, 2002; Galley et al., 2004; Galley et al., 2006; Zollmann and Venugopal, 2006; Hassan et al., 2009) put the focus on increasing the target output grammaticality, by combining together on the target side the linguistically augmented rule counterparts of unlabelled lexical source elements. However, as the target sentence's linguistic structure is not known, these 'string to tree' systems need to address a significant increase in the translation hypothesis space.

- A further family of methods, sometimes referred to as 'tree to tree' systems, aims to relate the translation process to linguistic information from both the source as well as the target side (Poutsma, 2000; Way, 1999; Hearne and Way, 2003; Eisner, 2003; Zhang et al., 2008b; Liu et al., 2009; Chiang, 2010). While this line of work aspires to reap the 'best of both worlds', it also faces increased sparsity issues as it strives to match together linguistic structures across both languages. It involves higher computational costs and is exposed to parsing errors from both source and target sides.

- Some systems belonging to all three cases above extend towards employing a source and/or target *forest* instead of a single-best parse tree (Mi et al., 2008; Liu et al., 2009). Such forest is a packed representation of a subset of all parse trees of the respective underlying sentence together with their probabilities. This data structure allows these systems to take into account more input from the parsing model employed than that offered by the Viterbi parse.

Interestingly, despite the intuitive advantages of such approaches, early on (Koehn et al., 2003) exemplified the difficulties of integrating linguistic information in translation systems. Syntax-based MT often suffers from inadequate constraints in the translation rules extracted, or from striving to combine these rules together towards a full derivation. Recent research tries to address these issues, e.g. by re-structuring training parse trees to better suit syntax-based SMT training (Wang et al., 2010). Other work moves from linguistically motivated synchronous grammars to systems where linguistic plausibility of the translation process is assessed through additional features in a phrase-based system (Venugopal et al., 2009; Chiang et al., 2009), obscuring the impact of higher level syntactic processes.

6.2 Our approach

While it is assumed that linguistic structure does correlate with some translation phenomena, in this work we do not employ it as the backbone of translation. In place of linguistically *constrained* translation imposing syntactic parse structure as the backbone of the pivoting mechanism between the source and target languages, we opt for linguistically *motivated* translation. We learn latent hierarchical structure, taking advantage of linguistic annotations but *shaped and trained* for translation.

We start by labelling each phrase-pair span in the word-aligned training data with multiple linguistically motivated categories. These labels are extracted from single-best syntactic parses of the training source sentences and offer multi-grained abstractions from the lexical surface of each bilingual span. The label charts listing the linguistic categories that cover each bilingual span, together with the training sentence-pairs, are the input of our learning algorithm. Our algorithm extracts the linguistically motivated rules and estimates the probabilities for a stochastic SCFG, without arbitrary constraints such as phrase or span sizes.

Estimating such grammars under a Maximum Likelihood criterion is known to be plagued by strong overfitting leading to degenerate estimates (DeNero et al., 2006). In contrast, our learning objective not only avoids overfitting the training data but, most importantly, learns joint stochastic synchronous grammars which directly aim at generalisation towards yet unseen instances. By advancing from structures which mimic linguistic syntax, to learning linguistically aware latent recursive structures targeting translation, we achieve significant improvements in translation quality for 4 different language pairs in comparison with a strong hierarchical translation baseline.

Our key contributions are presented in the rest of the chapter. We first introduce a joint translation model which separates hierarchical translation structure from phrase-pair emission. This model is based on a synchronous grammar design which takes advantage of our work on conditioning synchronous rules on reordering operations presented in Chapter 5. We then consider a *chart* over phrase-pair spans filled with source-language linguistically motivated labels. We show how we can employ this crucial input to extract and train a hierarchical translation structure model with millions of rules. We do this by efficiently examining all hierarchical structures that can be built taking advantage of these labellings. Subsequently, we establish which of these describe the data best, according to a smoothed learning objective based on Cross-Validated MLE and implemented in terms of the Cross-Validated EM algorithm presented in section 3.2. We continue by demonstrating how to decode with our model by constraining derivations to linguistic hints of the source sentence. We present our empirical results and close the chapter with a discussion of related work and our conclusions.

$$
\begin{aligned}
\mathrm{SBAR} &\rightarrow [\mathrm{WHNP}\ \ \mathrm{SBAR\backslash WHNP}] && \text{(a)}\\
\mathrm{SBAR\backslash WHNP} &\rightarrow \left\langle \mathrm{VP/NP}^{\mathbf{L}}\ \ \mathrm{NP}^{\mathbf{R}} \right\rangle && \text{(b)}\\
\mathrm{NP}^{\mathbf{R}} &\rightarrow [\mathrm{NP}\ \ \mathrm{PP}] && \text{(c)}\\
\mathrm{WHNP} &\rightarrow \mathrm{WHNP}_{\mathbf{P}} && \text{(d)}\\
\mathrm{WHNP}_{\mathbf{P}} &\rightarrow \text{which / der} && \text{(e)}\\
\mathrm{VP/NP}^{\mathbf{L}} &\rightarrow \mathrm{VP/NP}^{\mathbf{L}}_{\mathbf{P}} && \text{(f)}\\
\mathrm{VP/NP}^{\mathbf{L}}_{\mathbf{P}} &\rightarrow \text{is / ist} && \text{(g)}\\
\mathrm{NP}^{\mathbf{R}} &\rightarrow \mathrm{NP}^{\mathbf{R}}_{\mathbf{P}} && \text{(h)}\\
\mathrm{NP}^{\mathbf{R}}_{\mathbf{P}} &\rightarrow \text{the solution / die Lösung} && \text{(i)}\\
\mathrm{NP} &\rightarrow \mathrm{NP}_{\mathbf{P}} && \text{(j)}\\
\mathrm{NP}_{\mathbf{P}} &\rightarrow \text{the solution / die Lösung} && \text{(k)}\\
\mathrm{PP} &\rightarrow \mathrm{PP}_{\mathbf{P}} && \text{(l)}\\
\mathrm{PP}_{\mathbf{P}} &\rightarrow \text{to the problem / für das Problem} && \text{(m)}
\end{aligned}
$$

Figure 6.1: English-German SCFG rules for the relative clause(s) ‘which is the solution (to the problem) / der die Lösung (für das Problem) ist’. [ ] signify monotone translation, ⟨ ⟩ a swap reordering.

6.3 Joint Translation Model

Our model is based on a probabilistic Synchronous Context-Free Grammar (Wu, 1997; Chiang, 2005a). We employ binary SCFGs, i.e. grammars with a maximum of two non-terminals on the right-hand side. Also, for this work we only used grammars with either purely lexical or purely abstract rules involving one or two non-terminal pairs. An example can be seen in Figure 6.1, using the ITG-style notation and assuming the same non-terminal labels for both sides.

We utilise *probabilistic* SCFGs, where each rule is assigned a conditional probability of expanding the left-hand side symbol with the rule’s right-hand side. Phrase-pairs are emitted jointly and the overall probabilistic SCFG is a *joint* model over parallel strings.

6.3.1 Hierarchical Reordering SCFG

In section 5.2.2, we discussed the possible pitfalls of modelling translation under the SCFG formalism when directly extending monolingual non-terminals to their bilingual counterparts. The key issue revolves around the need to account for the fact that synchronous abstract categories must not only capture how the bilingual spans they cover combine with their source or target language context, but also their reordering preferences across the language pair. Effective propagation of these reordering preferences across the derivations of a linguistically motivated grammar can be beneficial when a linguistic category alone proves too coarse as a bilingual class to also capture reordering behaviour.

We address these issues by relying on an SCFG grammar design that is similar to the 'Lexicalised Reordering' grammar of section 5.4.2. As in the rules of Figure 6.1, we separate non-terminals according to the reordering patterns in which they participate. Non-terminals such as $\mathrm{B}^{\mathbf{L}}$, $\mathrm{C}^{\mathbf{R}}$ take part only in swapping right-hand sides $\langle \mathrm{B}^{\mathbf{L}}\ \ \mathrm{C}^{\mathbf{R}} \rangle$ (with $\mathrm{B}^{\mathbf{L}}$ swapping from the source side's left to the target side's right, $\mathrm{C}^{\mathbf{R}}$ swapping in the opposite direction), while non-terminals such as B, C take part solely in monotone right-hand side expansions [B C]. These non-terminal categories can appear also on the left-hand side of a rule, as in rule (c) of Figure 6.1.

However, in contrast with the Lexicalised Reordering grammar of Chapter 5, monotone and swapping non-terminals in this case do not emit phrase-pairs themselves. Rather, each non-terminal NT is expanded to a dedicated phrase-pair emitting non-terminal $\mathrm{NT}_{\mathbf{P}}$, which generates all phrase-pairs for it and nothing more. In this way, we explicitly model the preference of non-terminals to either expand towards a (long) phrase-pair or be further analysed recursively. This is done through the competition of expansions $\mathrm{NT} \rightarrow \mathrm{NT}_{\mathbf{P}}$ preparing to emit a phrase-pair, against the rest of the rules with NT as left-hand side that further analyse the non-terminal in abstract terms. Furthermore, this set of *pre-terminals* allows us to separate the higher order translation structure from the process that emits phrase-pairs, a feature we employ next to apply a different learning strategy on each part of our synchronous grammar.

In Chapter 5 this grammar design mainly contributed to model lexical reordering preferences. While we retain this function, for the rich linguistically-motivated grammars used in this chapter, this design effectively propagates reordering preferences above and below the current rule application, as in rules (a)-(c) of Figure 6.1, allowing to learn and apply complex reordering patterns. Using these non-terminals in the right-hand side of synchronous rules allows us to *synchronise* together the reordering preferences of adjacent spans, by combining together two bilingual non-terminals with a preference to reorder monotonically (rules (a) and (c)) or swap with each other between source and target sentences (rule (b)). In addition, employing these non-terminals on the left-hand side of synchronous productions allows us to *propagate* reordering preferences higher-up the synchronous tree derivation from the bottom-up perspective of a decoding algorithm, or to condition productions on reordering behaviour when we look at it as a top-down generative process.

The different types of grammar rules are summarised in abstract form in Figure 6.2, while the reader can find later in this chapter, in section 6.4.2 and

$$
\begin{array}{rlrl}
\mathrm{A} \rightarrow & [\mathrm{B}\ \ \mathrm{C}] & \mathrm{A} \rightarrow & \langle \mathrm{B}^{\mathbf{L}}\ \ \mathrm{C}^{\mathbf{R}} \rangle \\
\mathrm{A}^{\mathbf{L}} \rightarrow & [\mathrm{B}\ \ \mathrm{C}] & \mathrm{A}^{\mathbf{L}} \rightarrow & \langle \mathrm{B}^{\mathbf{L}}\ \ \mathrm{C}^{\mathbf{R}} \rangle \\
\mathrm{A}^{\mathbf{R}} \rightarrow & [\mathrm{B}\ \ \mathrm{C}] & \mathrm{A}^{\mathbf{R}} \rightarrow & \langle \mathrm{B}^{\mathbf{L}}\ \ \mathrm{C}^{\mathbf{R}} \rangle \\
\mathrm{A} \rightarrow & \mathrm{A}_{\mathbf{P}} & \mathrm{A}_{\mathbf{P}} \rightarrow & \alpha \ / \ \beta \\
\mathrm{A}^{\mathbf{L}} \rightarrow & \mathrm{A}^{\mathbf{L}}_{\mathbf{P}} & \mathrm{A}^{\mathbf{L}}_{\mathbf{P}} \rightarrow & \alpha \ / \ \beta \\
\mathrm{A}^{\mathbf{R}} \rightarrow & \mathrm{A}^{\mathbf{R}}_{\mathbf{P}} & \mathrm{A}^{\mathbf{R}}_{\mathbf{P}} \rightarrow & \alpha \ / \ \beta
\end{array}
$$

Figure 6.2: Hierarchical Reordering Grammar rule categories; A, B, C non-terminals; α, β source and target strings respectively.

Figure 6.4, a more concrete example on how this grammar design is applied in practice. We will subsequently refer to this grammar structure as Hierarchical Reordering SCFG (HR-SCFG).

6.3.2 Generative Model

We arrive at a probabilistic SCFG model which jointly generates source $\mathbf{e}$ and target $\mathbf{f}$ strings, by augmenting each grammar rule with a probability, summing up to one for every left-hand side. The probability of a derivation D of tuple $\langle \mathbf{e}, \mathbf{f} \rangle$ beginning from start symbol S is equal to the product of the probabilities of the rules used to recursively generate it.

We separate the structural part of the derivation D, down to the pre-terminals $\mathrm{NT}_{\mathbf{P}}$, from the phrase-emission part. The grammar rules pertaining to the structural part and their associated probabilities define a model $p(\sigma)$ over the latent variable σ. This determines the recursive, reordering and phrase-pair segmenting structure of translation, as in Figure 6.4. Given σ, the phrase-pair emission part merely generates the phrase-pairs utilising distributions from every $\mathrm{NT}_{\mathbf{P}}$ to the phrase-pairs that it covers, thereby defining a model over all sentence-pairs generated given each translation structure. The probabilities of a derivation and of a sentence-pair are then as follows:

$$p(D) = p(\sigma)\ p(\mathbf{e}, \mathbf{f} | \sigma) \tag{6.1}$$

$$p(\mathbf{e}, \mathbf{f}) = \sum_{D : D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle} p(D) \tag{6.2}$$

By splitting the joint model in a hierarchical structure model and a lexical emission one we facilitate estimating the two models separately. The following section discusses this.

<table>
<tr><td colspan="4">X, SBAR, WHNP+VP, WHNP+VBZ+NP</td></tr>
<tr><td></td><td colspan="3">X, VBZ+NP, VP, SBAR\WHNP</td></tr>
<tr><td colspan="3">X, SBAR/NN, WHNP+VBZ+DT</td><td></td></tr>
<tr><td></td><td colspan="2">X, VBZ+DT, VP/NN</td><td></td></tr>
<tr><td colspan="2">X, WHNP+VBZ,
SBAR/NP</td><td colspan="2">X, NP,
VP\VBZ</td></tr>
<tr><td>X, WHNP,
SBAR/VP</td><td>X, VBZ,
VP/NP</td><td>X, DT,
NP/NN</td><td>X, NN,
NP\DT</td></tr>
<tr><td><b>which</b></td><td><b>is</b></td><td><b>the</b></td><td><b>problem</b></td></tr>
</table>

Figure 6.3: The label chart for the source fragment 'which is the problem'. Only a sample of the entries is listed.

6.4 Learning Translation Structure

In the previous section we established the SCFG grammar design that we will be employing as summarised in Figure 6.2, as well as the probabilistic foundations of the joint translation model we will be learning. We will now describe our approach towards incorporating linguistic information in our model, so that we cover the distance from the generic framework of Figure 6.2, to a linguistically motivated model employing rules such as those of Figure 6.1.

Our first step will be to describe how to encode the linguistic information contained in automatically generated source sentence parse trees, into a data structure: a *chart* covering bilingual spans. Our aim is to label each such span with linguistically motivated categories which could prove helpful to describe the translation process and the bilingual correspondences between the source and target sentences. The next step is to build a linguistically motivated probabilistic *synchronous grammar* belonging to the HR-SCFG family, which is able to describe all SCFG structures that we can build by taking advantage of these labellings. Finally, we *estimate* the parameters of this grammar by disambiguating between *all* such alternative linguistically motivated explanations of the parallel training data using the Cross-Validated EM algorithm.

6.4.1 Phrase-Pair Label Chart

The input to our learning algorithm is a word-aligned parallel corpus. We consider as phrase-pair spans those that obey the word-alignment constraints of (Koehn et al., 2003). For every training sentence-pair, we also input a chart containing one or more labels for every synchronous span, such as that of Figure 6.3. Each label describes different properties of the phrase-pair (syntactic, semantic etc.), possibly in relation to its context, or supplying varying levels of abstraction (phrase-pair, determiner with noun, noun-phrase, sentence etc.). We aim to induce a recursive translation structure that explains the joint generation of the source and target sentence taking advantage of these phrase-pair span labels.

For this work, we employ the algorithm for assigning labels to word-aligned bilingual spans from (Zollmann and Venugopal, 2006). Their algorithm outputs linguistically motivated labels using a syntactic parse tree covering the target sentence of a sentence-pair. Crucially, we use their algorithm on parses of *source* sentences instead.

An important point is that, contrary to (Zollmann and Venugopal, 2006), we assign all applicable labels to every span. In this way, each label set captures the features of the source side's parse-tree without being bounded by the actual parse structure, as well as provides a coarse (X, NP) to fine-grained (DT+JJ+NN, VP\VBZ) view of the source phrase. Furthermore, including all labels for each span allows us to evaluate their usefulness in taking part in synchronous structures according to our learning objective, without artificial biases favouring certain labels over others. In this way, we populate the label charts which, together with the sentence-pairs, form the input of our learning algorithm.

Overall, given a parse of the source sentence, each span is assigned the following kinds of labels:

Phrase-Pair All phrase-pairs are assigned the X label. This conveys no more information apart from the fact that the underlying bilingual span can be considered a valid phrase-pair. In this way, it functions as a back-off label, both when no further specialised linguistic label is applicable, as well as by competing for probability mass with labels which do cover a span but do not seem to contribute towards explaining the translation process according to the learning objective function.

Constituent The source phrase is a constituent A.

Concatenation of Constituents The source phrase is labelled A+B as a concatenation of constituents A and B and similarly for 3 constituents. These labels are oftentimes used for bilingual spans with a source side which violates the bracketing structure of the source parse tree. Already from work on Phrase-Based SMT, these non-constituents are known to be useful for translation, even though they do not correspond to a monolingual constituent span in the linguistic structure of the source sentence (Koehn et al., 2003).

Partial Constituents Categorial grammar (Bar-Hillel, 1953) inspired labels A/B, A\B, indicating a partial constituent A missing constituent B right or left respectively. These context-aware linguistic labels have the potential to help orchestrate translation operations on a bilingual span with the bilingual context surrounding it, a function they frequently fulfilled in our empirical work presented later in the chapter.

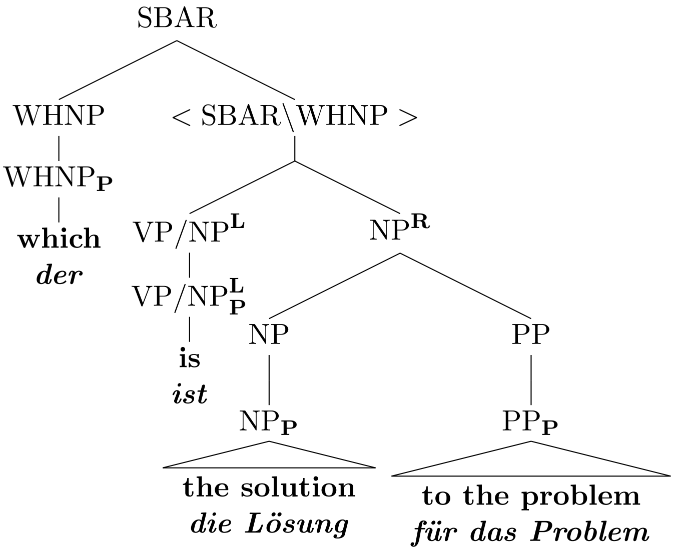


Figure 6.4: A derivation of a sentence fragment with the grammar of Figure 6.1.

6.4.2 Grammar Extraction

From every word-aligned sentence-pair and its label chart, we extract SCFG rules as those of Figure 6.2. There are three types of rules extracted:

- *Binary* rules are extracted from adjoining bilingual spans up to the whole sentence-pair level. The non-terminals of both left and right-hand side are derived from the label names plus their reordering function (monotone, left/right swapping) in the span examined.

- A single *unary* rule per non-terminal NT generates the phrase-pair emitting non-terminal $\mathrm{NT}_{\mathbf{P}}$. A transition $\mathrm{NT} \rightarrow \mathrm{NT}_{\mathbf{P}}$ signifies that the bilingual span will be covered by a single phrase-pair and will not be further analysed in bilingual sub-constituents.

- Unary rules $\mathrm{NT}_{\mathbf{P}} \rightarrow \alpha \;/\; \beta$ generating a *phrase-pair* are created for all the labels covering it.

The result is a grammar which can capture a rich array of translation phenomena based on linguistic and lexical grounds. It can also explicitly model the balance between memorising long phrase-pairs and generalising over yet unseen ones, as shown in the next example.

The derivation in Figure 6.4 illustrates some of the formalism's features. A preference to reorder based on lexical *content* is applied for is / ist. Noun phrase

$\text{NP}^{\mathbf{R}}$ is recursively constructed with a preference to constitute the right branch of an order swapping non-terminal expansion. This is matched with $\text{VP/NP}^{\mathbf{L}}$ which reorders in the opposite direction. The labels VP/NP and SBAR\WHNP allow linguistic syntax *context* to influence the lexical and reordering translation choices. Crucially, *all* these lexical, attachment and reordering preferences (as encoded in the model's rules and probabilities) must be *stochastically* matched together to arrive at the analysis in Figure 6.4. This matching takes place according to the preferences encoded in the rule's probabilities and is not enforced by applying linguistic or other constraints.

While we label the phrase-pairs similarly to (Zollmann and Venugopal, 2006), the extracted grammar is rather different. We *do not* employ rules that are grounded to lexical context ('gap' rules). Using such rules would allow the derivation of the sentence-pair fragment in Figure 6.4 to employ lexically grounded productions such as:

$$\text{SBAR} \rightarrow \text{ which is NT}_{\boxed{1}}\ \text{PP}_{\boxed{2}}\ /\ \text{der NT}_{\boxed{1}}\ \text{PP}_{\boxed{2}}\ \text{ist}$$

If their right-hand side lexical context is matched in a test sentence, these productions allow for a much shallower synchronous derivation tree than our grammar, which emphasises abstract recursive structure and employs synchronous trees of a larger depth, as can be seen in Figure 6.4. Using such 'gap' rules would probably offer our model the chance to score even higher in terms of translation performance. However, in this work, the choice not to employ lexicalised abstract productions allows us to clearly separate the lexical and abstract parts of the synchronous grammar we are learning. This facilitates the separate estimation of their parameters and allows us to focus on the highly interesting challenge of inducing an effective abstract hierarchical translation structure.

6.4.3 Parameter Estimation

Our joint translation model of equations (6.1) and (6.2) consists of two clearly separated probabilistic components:

1. A hierarchical translation structure model $p(\sigma)$ generating the recursive SCFG structure σ beginning at the start symbol S and down to the HR-SCFG pre-terminals $\text{NT}_{\mathbf{P}}$.

2. The phrase-pair emission model $p(\mathbf{e}, \mathbf{f}|\sigma)$ generating the lexical surface $\langle \mathbf{e}, \mathbf{f} \rangle$ given the pre-terminals of structure σ.

We take advantage of the form of our model to apply a different estimation strategy for the parameters of each of the two components. We draw our motivation for this choice from the observation that the stochastic variables modelled by each of the two components can be considered to be related in a different manner to the training data, which are made of sentence-pairs coupled with their label charts. The phrase-pair emitting model is constituted of a set of conditional distributions $p(\tilde{e}, \tilde{f}|\mathrm{NT}_{\mathbf{P}})$ of phrase-pairs $\langle \tilde{e}, \tilde{f} \rangle$ given their covering label $\mathrm{NT}_{\mathbf{P}}$, with each entry corresponding to a phrase-pair emitting rule $\mathrm{NT}_{\mathbf{P}} \rightarrow \ \alpha \ / \ \beta$. We make the assumption that these distributions correspond to the *observed* distribution of phrase-pairs being covered by each label in the label charts of Figure 6.3, rendering the estimation of their parameters a case of learning from *complete* data.

However, the model $p(\sigma)$ over the recursive translation structure is related to the *unobserved* variable σ, which describes all aspects of translation apart from phrase-pair emission. Estimating the parameters of this model necessitates learning from *incomplete* data, as our training data does not include information on the structures that the labels of the bilingual charts participate in.

Phrase-Pair Emission Model We estimate the parameters for the phrase-emission model $p(\mathbf{e}, \mathbf{f}|\sigma)$ of equation (6.1) using Relative Frequency Estimation (RFE) on the *label charts* induced for the training sentence-pairs, after the labels have been augmented by the reordering indications. In this RFE estimate, every rule $\mathrm{NT}_{\mathbf{P}} \rightarrow \ \alpha \ / \ \beta$ receives a probability in proportion with the number of times that $\alpha \ / \ \beta$ was covered by the NT label.

This is based on the aforementioned assumption that the phrase-pair emission distributions $p(\tilde{e}, \tilde{f}|\mathrm{NT}_{\mathbf{P}})$ correspond to the observed distribution of phrase-pairs being covered by each label in the label charts. We employ this simplifying assumption to crucially reduce the set of free parameters during estimation to that which is most interesting for the work in this chapter: the probability distributions relating to the unlexicalised, abstract recursive part of the translation process.

Translation Structure Model Estimating the parameters under Maximum-Likelihood Estimation (MLE) for the latent translation structure model $p(\sigma)$ is bound to overfit towards memorising whole sentence-pairs as discussed in section 3.1.5, with the resulting grammar estimate not being able to generalise past the training data. However, apart from overfitting towards long phrase-pairs, a grammar with millions of structural rules is also liable to overfit towards degenerate latent structures which, while fitting the training data well, have limited applicability to unseen sentences.

We avoid both pitfalls by estimating the grammar probabilities with the Cross-Validating Expectation-Maximization algorithm (CV-EM) of section 3.2, a cross-validating instance of the EM algorithm (Dempster et al., 1977). It works iteratively on a partition of the training data, climbing the likelihood of the training data while cross-validating the latent variable values. It does so by considering for every training data point only those latent variable values which can be produced by models built from the rest of the data, excluding the current part. As a result, the estimation process simulates maximising future data likelihood, using the training data to directly aim towards strong generalisation of the estimate.

For our probabilistic SCFG-based translation structure variable σ, implementing CV-EM boils down to a synchronous version of the Inside-Outside algorithm, modified to enforce the CV criterion. In this way we arrive at cross-validated ML estimate of the σ parameters, while keeping the phrase-emission parameters of $p(\mathbf{e}, \mathbf{f}|\sigma)$ fixed. After the label charts are constructed for each sentence-pair and source parse tree and the linguistically motivated HR-SCFG is extracted as described in section 6.4.2, the implementation follows the lines drawn in section 5.3.2 and the workflow of Figure 5.8. The only difference is that in this case, during the estimation of the parameters of the hierarchical translation structure model $p(\sigma)$, we exclude the parameters of the phrase-pair emission model from estimation, employing in their place the values estimated with RFE from the label charts. The CV-criterion, apart from avoiding overfitting also results in discarding the structural rules which are only found in a single part of the training corpus, leading to a more compact grammar while still retaining millions of structural rules that are more hopeful to generalise.

Overall, unravelling the joint generative process, by modelling latent hierarchical structure separately from phrase-pair emission, allows us to concentrate our inference efforts towards the hidden, higher-level translation mechanism.

6.5 Decoding Aspects

Up to this point, we have presented how to extract a probabilistic Hierarchical Reordering SCFG grammar using the word-aligned parallel corpus and the source parse trees, as well as how to estimate its parameters employing RFE for the phrase-pair emission part and CV-EM for the part of the grammar generating the abstract synchronous structure. In the end, what we have is a CV-MLE estimate of a joint translation model and what we still miss is how to actually translate with it. In this section we build a translation system around our learnt model, the Latent Translation System (LTS), and we evaluate empirically its performance in comparison with a Hiero state-of-the-art baseline.

We begin by integrating the joint model estimate as part of a log-linear, feature based translation model. While the form of this model is similar to those frequently employed in Phrase-Based SMT or Hiero-like systems, translating with it necessitates novel pruning and decoding[1] strategies. This is necessary in order to handle the significant increase in both the grammar size as well as the decoding hypothesis space, a subject we treat next. With all the decoding details sorted out, we move on to evaluate our system against a heuristically estimated

[1] Decoding refers to the foundational SMT problem of recovering the target language translation with the highest probability, given a source sentence and a model estimate; see sections 2.1 and 2.2.

hierarchical translation baseline across four diverse language pairs.

6.5.1 Decoding Model

The induced joint translation model can in general be used to recover $\arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f})$, as this is equal to $\arg\max_{\mathbf{e}} p(\mathbf{e},\mathbf{f})$. Nevertheless, instead of using our model estimate on its own, we employ the induced probabilistic HR-SCFG $\mathbf{G}$ as the backbone of a log-linear, feature based translation model, with the derivation probability $p(D)$ under the grammar estimate being one of the features. This is augmented with a small number n of additional smoothing features ϕ_i for derivation rules r. These are:

1. Conditional phrase translation probabilities

2. Lexical phrase translation probabilities

3. A word generation penalty

4. A count of swapping reordering operations

Feature categories (1), (2) and (3) are applicable to phrase-pair emission rules and we use values for both translation directions, while (4) is only triggered by structural rules. These extra features assess translation quality past the synchronous grammar derivation and encode general reordering or word emission preferences for the language pair. As an example, while our probabilistic HR-SCFG maintains a separate joint phrase-pair emission distribution for every non-terminal, the smoothing features (1) and (2) above assess the conditional translation of surface phrases irrespective of any notion of recursive translation structure. Overall, it is important to note that in this decoding scheme, while the phrase-pair emission part of the grammar is supported by the usual smoothing features found in a typical Phrase-Based SMT system, the hierarchical part of the model's derivations is completely reliant on the estimates of the previous section, apart from a mere count of swapping reorderings.

The final feature is the language model score for the target sentence, mounting up to the following model used at decoding time, with the feature weights λ trained by Minimum Error Rate Training (MERT) (Och, 2003) on a development corpus. We use this model to output the translation corresponding to the most probable derivation.

$$p(D \overset{*}{\Rightarrow} \langle \mathbf{e}, \mathbf{f} \rangle) \propto p(\mathbf{e})^{\lambda_{lm}} p_{\mathbf{G}}(D)^{\lambda_{\mathbf{G}}} \prod_{i=1}^{n} \prod_{r \in D} \phi_i(r)^{\lambda_i} \tag{6.3}$$

6.5.2 Pruning Strategies

The HR-SCFG we extract and train from the parsed, word-aligned parallel corpus contains millions of abstract bilingual rules on top of the millions of phrase-pair emission productions. In addition, instead of the two non-terminals (S, X) of Hiero synchronous grammars, our grammar makes use of hundreds of thousands of non-terminals, capturing linguistic as well as reordering correspondences. Using such a grammar efficiently during decoding demands the introduction of effective pruning strategies: (a) to reduce the size of the grammar prior to decoding, cutting down on the decoder's memory footprint and the time needed to search through the grammar and (b) to reduce the number of hypotheses about the translation structure by taking advantage of automatically generated input parse trees, while making sure that the remaining hypothesis space is diverse enough to complete derivations. For these reasons, we apply the following modifications to the Joshua SCFG decoder (Li et al., 2009), which we use to translate with our model of equation (6.3):

Expected Counts Rule Pruning To compact the hierarchical structure part of the grammar prior to decoding, we prune rules that fail to accumulate more than a number α of expected counts during the last CV-EM iteration. For English to German and for the value $\alpha = 10^{-8}$ that we use throughout our experiments, this brings the structural rules from 15M down to 1.2M.

The phrase-pair emitting rules are not pruned at this stage, allowing all phrase-pairs extracted from the training data to take part in test data derivations. If we only need to translate a fixed test set, this lexical part of the synchronous grammar can be reduced in size by discarding entries corresponding to source phrases that do not appear in the test set. The same however does not apply to the abstract part of the grammar, for which it is difficult to discard prior to decoding rules which cannot be applied for a particular test set. This would require an expensive parsing run through the test set to find out which rules do not take part in any derivation and is unlikely to reduce considerably the size of the abstract part of the grammar. This is because while the lexical part of the grammar can be filtered in respect to the *unambiguous* source input, the abstract part relates to the *ambiguous* hidden hierarchical structure, over which our model considers a multitude of hypotheses employing thousands of rules per input sentence.

Overall, we consider pruning based on the expected counts of each rule in the training set according to our estimate as a much more informed pruning criterion than those based on probability values or right-hand side counts. Pruning the grammar based on the estimated probability values is sub-optimal as these are not comparable across left-hand sides. A low probability right-hand side expansion of a frequent left-hand side has more chances to actually be employed during decoding, than a high probability expansion of a very rare left-hand side. In addition, pruning based on keeping a fixed number of productions per left-hand side symbol makes also little sense, as frequent symbols such as X or NP need many more expansions than highly specialised ones.

Instead of these largely arbitrary pruning approaches, having access to the *cross-validated* expected counts of synchronous productions allows us to reduce the size of the grammar according to the expectation that these will be employed in derivations under our model, discarding rules that are expected to appear extremely infrequently. This makes sure that the ensemble of rules which together form most of the high probability derivations remains in the grammar and we avoid inadvertently removing a crucial component of frequent derivations.

Source Labels Constraints As for this work the phrase-pair labels used to extract the grammar are based on the linguistic analysis of the source side, we can construct the label chart for every input sentence from its single-best syntactic parse[2]. We subsequently use it to consider only derivations with synchronous spans that are covered by non-terminals matching one of the input sentence labels for those spans. This applies both for the non-terminals covering phrase-pairs as well as the higher level parts of the derivation.

In this manner we not only constrain the translation hypotheses resulting in considerably faster decoding time, but, more importantly, we may ground the hypotheses more closely to the available linguistic information of the source sentence. This is of particular interest as we move up the derivation tree, where an initial wrong choice below could propagate towards hypotheses wildly diverging from the input sentence's linguistic parse.

Even though for the work presented in this chapter we only employ labels extracted solely from the source sentence linguistic analysis, these hypotheses constraints can also be applied to grammars employing non-terminals related to both the source as well as the target syntactic structure. In this case, the constraints would apply to the part of the rules related to the source sentence structure, still allowing a considerable part of the hypotheses to be pruned away. This would make sure that the remaining hypothesis space at least satisfies the source side linguistic structure, which may be assumed observed given the single-best automatically generated parse tree of the test sentence.

Per Non-Terminal Pruning The Joshua decoder uses a combination of beam and cube-pruning (Huang and Chiang, 2007). As our grammar uses non-terminals in the hundreds of thousands, it is important not to prune away prematurely non-terminals covering smaller spans and to leave more options to be considered as we move up the derivation tree. Apart from producing sub-optimal translations, pruning according to the *locally* optimal inside probability and without respect to

[2] Section 6.4.1 offers more details on how we construct the label charts and the kinds of labels that we populate them with.

the surrounding context might even lead to an inability to complete full derivations for a test sentence.

For this, for every cell in the decoder's chart, we keep a separate bin per non-terminal and prune together hypotheses leading to the same non-terminal covering a cell. This allows full derivations to be found for all input sentences, as well as avoids aggressive pruning at an early stage. Given the source label constraint discussed above, this does not increase running times or memory demands considerably as we allow only up to a few tens of non-terminals per span.

6.6 Experiments

We evaluate our method on four different language pairs with English as the source language and French, German, Dutch and Chinese as target. The data for the first three language pairs are derived from parliament proceedings sourced from the Europarl corpus (Koehn, 2005), with WMT-07 development and test data for French and German. The data for the English to Chinese task is composed of parliament proceedings and news articles. For all language pairs we employ 200K and 400K sentence pairs for training, 2K for development and 2K for testing (single reference per source sentence). Both the baseline and our method decode with a 3-gram language model smoothed with modified Knesser-Ney discounting (Chen and Goodman, 1998), trained on around 1M sentences per target language. The parses of the source sentences employed by our system during training and decoding are created with the Charniak parser (Charniak, 2000).

We compare against the state-of-the-art hierarchical translation (Chiang, 2005a) baseline of section 5.2.3, based on the Joshua translation system under the default training and decoding settings (`josh-base`). Apart from evaluating against a state-of-the-art system, especially for the English-Chinese language pair, the comparison has an additional interesting aspect. The heuristically trained baseline takes advantage of 'gap rules' to reorder based on lexical context cues, but makes very limited use of the hierarchical structure above the lexical surface. In contrast, our method induces a grammar with no such rules, relying on lexical content and the strength of a higher level translation structure instead. For this, comparing the two approaches together also evaluates if a grammar with an emphasis on unlexicalised hierarchical structure as ours is able to provide state-of-the-art results, something which prior work failed to establish.

6.6.1 Training & Decoding Details

To train our Latent Translation Structure (LTS) system, we used the following settings. CV-EM cross-validated on a 10-part partition of the training data and performed 10 iterations. The structural rule probabilities were initialised to uni-

	English-Chinese		English-German	
	`josh-base`	`lts`	`josh-base`	`lts`
Non-terminals	2	203,650	2	198,204
Abstract rules (training)	-	15,848,032	-	24,316,629
Abstract rules (decoding)	2	1,117,974	2	2,192,694
Lexicalised rules (decoding)	1,759,709	3,154,467	6,160,252	3,729,977
Total rules (decoding)	1,759,711	4,272,441	6,160,254	5,922,671

Table 6.1: Ruleset sizes for the grammars used in our `lts` system and the `josh-base` baseline extracted from 400K word-aligned English-Chinese and English-German sentence-pairs.

form per left-hand side, the phrase-emission distributions were kept fixed to their RFE estimate as discussed in section 6.4.3.

The decoder does not employ any 'glue grammar', as is usual with hierarchical translation systems to limit reordering up to a certain cut-off length. Instead, we rely on our LTS grammar to reorder and construct the translation output up to the full sentence length.

In summary, our system's experimental pipeline is as follows:

1. All source sentences of the training corpus are parsed using Charniak's parser (Charniak, 2000), and label charts are created from these parses.

2. The Hierarchical Reordering SCFG is extracted and its parameters are estimated employing CV-EM.

3. The structural rules of the estimate are pruned according to their expected counts and smoothing features are added to all rules.

4. We train the feature weights under MERT.

5. We decode with the resulting log-linear model.

The overall training and decoding setup is appealing also regarding computational demands. On an 8-core 2.3GHz system, training on 200K sentence-pairs demands 4.5 hours, while decoding runs on 25 sentences per minute.

Table 6.1 compares the sizes of the abstract and lexicalised parts of the HR-SCFG used in our LTS system with those of the baseline for two of the language pairs: English-Chinese and English-German. The two grammars take a different view on modelling hierarchical translation structure. Our system considers hundreds of thousands of bilingual categories while the baseline employs a single non-terminal X past the start symbol. Furthermore, the only purely abstract rules in the baseline are the two glue rules of Figure 5.7 which monotonically concatenate the translations produced using the non-contiguous phrase-pair productions of Figure 5.6. In contrast, our system makes use of million of linguistically motivated abstract structure rules extracted from the training data label charts, complemented by purely lexical phrase-pair emission productions.

The lexicalised rules used by the two systems also differ in nature. All but the two glue rules that the baseline employs are rules with a lexicalised right-hand side, emitting non-contiguous phrase-pairs from the single X non-terminal. These rules are extracted from bilingual spans with a length of at most 10 on each side. In contrast, the lexicalised rules that our LTS system employs, are expanding the multiple bilingual categories of the HR-SCFG grammar to generate contiguous phrase-pairs. Such rules are extracted for every label covering a bilingual span in the label chart we generate for each training sentence. Consistent with the work presented in Chapters 4 and 5, we extract these rules without any constraint on the length of the bilingual spans we consider. As can be noted in Table 6.1, the exact proportion of lexicalised rules used by the baseline in comparison with those used by LTS differs per language-pair, depending on factors such as the distribution of lengths of sentences and the number of labels covering the bilingual spans. Still, the number of lexicalised rules used by the two systems remains in the same order of magnitude.

The number of hierarchical structure rules is reduced prior to decoding: while we explore derivations using all rules extracted from the training data label charts during training, to speed up decoding we prune rules which have gathered an extremely low number of expected counts during training, as explained in section 6.5.2. Overall, the total number of rules employed by our system and the baseline to decode the test set, are also in the same order of magnitude

6.6.2 Results

LTS vs. Baseline Table 6.2 presents the results for the baseline and our method for the 4 language pairs, for training sets of both 200K and 400K sentence pairs. Our system (`lts`) outperforms the baseline for all 4 language pairs for both BLEU and NIST scores, by a margin which scales up to +1.92 BLEU points for English to Chinese translation when training on the 400K set. In addition, increasing the size of the training data from 200K to 400K sentence pairs widens the performance margin between the baseline and our system, in some cases considerably. All but one of the performance improvements are found to be statistically significant (Koehn, 2004) at the 95% confidence level, most of them also at the 99% level.

Training set size	**English to**	French		German	
		BLEU	NIST	BLEU	NIST
200K	`josh-base`	29.20	7.2123	18.65	5.8047
	`lts`	**29.43**	**7.2611****	**19.10****	**5.8714****
400K	`josh-base`	29.58	7.3033	18.86	5.8818
	`lts`	**29.83**	**7.4000****	**19.49****	**5.9374****

Training set size	**English to**	Dutch		Chinese	
		BLEU	NIST	BLEU	NIST
200K	`josh-base`	21.97	6.2469	22.34	6.5540
	`lts`	**22.31***	**6.2903***	**23.67****	**6.6595****
400K	`josh-base`	22.25	6.2949	23.24	6.7402
	`lts`	**22.92****	**6.3727****	**25.16****	**6.9005****

Table 6.2: Experimental results for training sets of 200K and 400K sentence pairs. Statistically significant score improvements from the baseline at the 95% confidence level are labelled with a single star, at the 99% level with two.

English to	**French**	**German**	**Dutch**	**Chinese**
`joshua-base`	29.20	18.65	21.97	22.34
`lts-heuristic`	29.03	18.56	22.00	21.46
`lts`	**29.43****	**19.10****	**22.31****	**23.67****

Table 6.3: Comparison of our LTS system (`lts`) against a system employing the same HR-SCFG grammar design and decoding options, albeit with *heuristic* probability estimates (`lts-heuristic`), instead of the CV-EM estimates the LTS system uses. All numbers refer to BLEU scores and two stars reflect a significant result at the 99% confidence level. The scores of the baseline system are provided as reference points.

	System	**200K**	**400K**
(a)	`lts-nolabels`	22.50	24.24
	`lts`	**23.67****	**25.16****
(b)	`josh-base-lm4`	23.81	24.77
	`lts-lm4`	**24.48****	**26.35****

Table 6.4: Additional experiments for English to Chinese translation examining (a) the impact of the linguistic annotations in the LTS system (`lts`), when compared with an instance not employing such annotations (`lts-nolabels`) and (b) decoding with a 4th-order language model (`-lm4`). BLEU scores for 200K and 400K training sentence pairs.

We selected an array of target languages of increasing reordering complexity with English as source. Translating to French involves mainly local reordering phenomena, while German and Dutch call for longer range reordering, especially for subordinate clauses. English to Chinese translation poses further reordering challenges with complex, often long-range reordering patterns between the two languages. Examining the results across the target languages, LTS performance gains increase the more challenging the sentence structure of the target language is in relation to the source's, as highlighted when translating to Chinese. Even for Dutch and German, which pose additional challenges such as compound words and morphology which we do not explicitly treat in the current system, LTS still delivers significant improvements in performance. Additionally, the robustness of our system is exemplified by delivering significant performance increases for all language pairs.

CV-EM vs. Heuristic Estimation In our LTS system, we brought together the HR-SCFG design of section 6.3.1 with the linguistically motivated label chart of section 6.4.1, to extract an SCFG trained with the CV-EM algorithm of section 3.2 and decode with the pruning methodology of section 6.5.2. In a further line of experiments, we wish to isolate the contribution of the learning approach in the system's performance. We aim to examine the impact of using CV-EM to train the probabilistic SCFG used by the LTS system, against a heuristic estimator. For this, we assemble a system that is identical to LTS in both extracting the HR-SCFG grammar as well as decoding with it, apart from the fact that it employs *heuristic* estimates for the production probabilities (`lts-heuristic`). These are set similarly to the approach used in the baseline: extraction counts are registered each time an abstract rule can be extracted from the label chart of a sentence-pair, or when a phrase-pair can be extracted from the word-aligned sentence-pair.

The heuristic learning of an HR-SCFG extracts a massive number of rules, with the grammar extracted from 200K of English-German sentence-pairs counting more than 87M rules. Under CV-EM, cross-validating the abstract rules and pruning them according to their expected counts reduces the size of the SCFG used during decoding considerably. We cannot use the exact same pruning method for the baseline and for this reason we are forced to resort to a different pruning strategy that is relatively comparable. The heuristic estimator does not use expected counts but extraction counts instead, and we use the latter to prune the heuristically estimated grammar of `lts-heuristic` prior to decoding. Removing all abstract rules which are extracted less than 95 times and all phrase-pair emission rules extracted only once, results in a synchronous grammar of a comparable size like that employed by `lts`.

Table 6.3 presents a comparison of the translation performance of our LTS system against its heuristically estimated variation, across all four language pairs when training on 200K sentence-pairs. On one hand, the `lts-heuristic` implementation performs reasonably well, scoring in all but one of the translation tasks within -0.2 BLEU in comparison to the hierarchical baseline, with English to Chinese being the exception. We attribute this result to the robustness of our HR-SCFG grammar design as well as the pruning constraints we employ during decoding, which make sure that the translation hypotheses do not deviate from the syntactic structure of the test sentence. Still, the HR-SCFG grammar estimated with CV-EM and pruned according to the expected counts amassed during estimation, significantly outperforms the heuristically estimated one. CV-EM estimates the parameters of the synchronous productions according to how useful they are in explaining the bilingual correspondences between source and target training sentences, instead of merely examining how often they can be extracted from the training data. The results of Table 6.3 indicate that this is not solely a feature with a strictly theoretical value, but on the contrary also manifests itself in terms of translation performance.

Effect of Linguistically-Motivated Labels & LM For the English to Chinese translation task, we performed further experiments along two axes. We first investigate the contribution of the linguistic annotations, by comparing our complete system (`lts`) with an otherwise identical implementation (`lts-nolabels`) which does not employ any linguistically motivated labels. The latter system then uses a labels chart as that of Figure 6.3, which however labels all phrase-pair spans solely with the generic X label. The results in Table 6.4(a) indicate that a large part of the performance improvement can be attributed to the use of the linguistic annotations extracted from the source parse trees, indicating the potential of the LTS system to take advantage of such additional annotations to deliver better translations.

The second additional experiment relates to the impact of employing a stronger language model during decoding, which may increase performance but slows down decoding speed. Notably, as can be seen in Table 6.4(b), switching to a 4-gram LM results in performance gains for both the baseline and our system. While the margin between the two systems decreases, our system continues to deliver a considerable and significant improvement in translation BLEU scores.

6.7 Related Work

In this chapter, we focus on the combination of learning latent structure with syntax and linguistic annotations, exploring the crossroads of machine learning, linguistic syntax and machine translation. A first point of comparison with the existing literature stems from our use of a *joint* translation model. Training a joint phrase-based probability model was first discussed in (Marcu and Wong, 2002), and even though it was trained by maximising the joint likelihood of the training data, the estimates were still converted to conditional translation distributions prior to decoding. Even though the work by (Marcu and Wong, 2002) was highly influential for their efforts to train a phrase-based model under a well-understood learning criterion as we discuss in section 2.3.2, the vast majority of the state-of-the-art performing translation models are centred on conditional translation distributions. In this chapter, we go against the conventional wisdom in the SMT field to show that a translation system based on such a joint model can perform competitively in comparison with conditional probability models, when it is augmented with a rich latent hierarchical structure trained adequately to avoid overfitting.

Earlier approaches for linguistic syntax-based translation such as (Yamada and Knight, 2001; Galley et al., 2006; Huang et al., 2006; Liu et al., 2006) focus on memorising and reusing parts of the structure of the source and/or target parse trees and constraining decoding by the input parse tree. In contrast to this approach, we choose to employ linguistic annotations in the form of unambiguous synchronous span labels, while discovering ambiguous translation structure taking advantage of them. Our approach avoids assuming that translation can be *fully* explained through linguistic structure tree correspondences and transformations, an inflexible strong assumption which we consider unnecessary: it imposes monolingual linguistic structure as the sole pivoting mechanism between the two languages even though it only partially correlates with translation decisions. Instead, here we focus on making external information based on linguistic analyses of source sentences available to a learner that optimises model estimation according to a *translation* learning objective. Then, our learning algorithm disambiguates *which* of the linguistic syntax patterns are indeed informative to explain the bilingual correspondences.

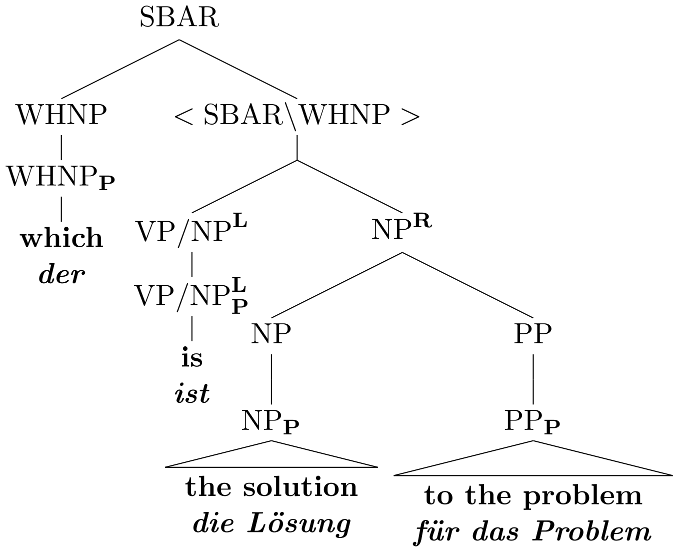
Later work (Marton and Resnik, 2008; Venugopal et al., 2009; Chiang et al., 2009) takes a more flexible approach which is more similar to our own efforts. They opt to influence translation output using linguistically motivated features, or features based on source-side linguistically-guided latent syntactic categories (Huang et al., 2010). However, the features employed by these methods are *local* in nature, considering the linguistic plausibility of applying individual synchronous rules. As a result, these efforts totally lack the concept of a linguistically motivated hierarchical abstract *structure* reaching across the whole sentence-pair, which is exactly the focus of our own methodology. Putting these crucial differences aside, a feature-based approach and ours are not mutually exclusive, as we also employ a limited set of features next to our trained model during decoding. We find augmenting our system with a more extensive feature set an interesting research direction for the future.

An array of recent work (Chiang, 2010; Zhang et al., 2008b; Liu et al., 2009) sets off to utilise source *and* target syntax for translation. While for this work we constrain ourselves to source language syntax annotations, our method can be directly applied to employ labels taking advantage of linguistic annotations from both sides of translation. The decoding constraints of section 6.5.2 can then still be applied on the source part of hybrid source-target labels.

For the experiments in this chapter we employ a label set similar to the nonterminals set of (Zollmann and Venugopal, 2006). However, the synchronous grammars we learn share few similarities with those that they heuristically extract, with Figure 6.5 comparing example structures based on the two grammar designs. The HR-SCFG we adopt allows capturing more complex reordering phenomena and, in contrast to both (Chiang, 2005a; Zollmann and Venugopal, 2006), is not exposed to the issues highlighted in section 5.2.2. Nevertheless, our results underline the potential of linguistic annotations similar to those of (Zollmann and Venugopal, 2006) as part of latent translation variables.

The majority of the aforementioned work does not concentrate on *learning* hierarchical, linguistically motivated translation models. Yamada and Knight (2001) employ the EM algorithm to train a syntax-driven, word-based translation model, making use of fixed size syntactic and lexical units. However, when the same model is extended to allow translating with phrase-pairs, a heuristic estimator is used to train the phrase-emission probabilities (Yamada and Knight, 2002). Galley et al. (2006) employ EM to train a translation model which generates translations by combining together syntactic units of variable sizes. However, they mitigate EM's overfitting behaviour by constraining the size of these units, so that each encompasses at most four elementary syntactic elements, in contrast with our approach which does not impose such arbitrary constraints.

Cohn and Blunsom (2009) sample rules of the form proposed in (Galley et al., 2004) from a Bayesian model, employing Dirichlet Process priors favouring smaller rules to avoid overfitting. Their grammar is however also based on the target parse-tree structure, with their system surpassing a weak baseline by a small margin. In contrast to the Bayesian approach, which imposes external priors to lead estimation away from degenerate solutions, we take a *data-driven* approach to arrive to estimates which generalise well. The rich linguistically motivated latent variable learnt by our method delivers translation performance that compares favourably to a state-of-the-art system.



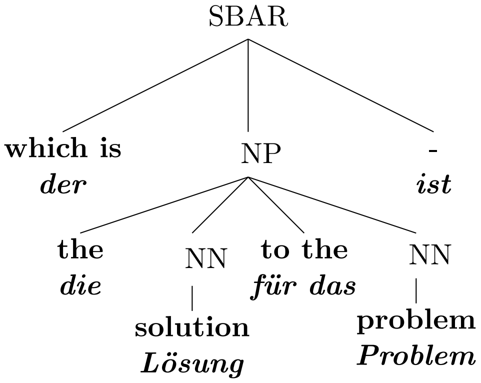


Figure 6.5: The HR-SCFG structure above is compared against a possible derivation below of the same synchronous subtree under a grammar extracted by the SAMT system. Our grammar emphasises abstract hierarchical translation structure conditioned on reordering behaviour, while the SAMT grammar relies on lexicalised, linguistically influenced productions.

Finally, in the previous chapter we also employ the CV-EM algorithm to estimate the parameters of an SCFG, albeit for a much simpler one based on a handful of non-terminals. Here we take advantage of some of the grammar design principles empirically shown in Chapter 5 to aid in inducing robust hierarchical translation structures covering whole sentence-pairs. However, we do this for an immensely more complex grammar with millions of hierarchical latent structure rules and show how such grammar can be learnt and applied taking advantage of source language linguistic annotations.

6.8 Discussion

This chapter wraps up the progression of work starting from Chapter 4 on learning phrase-based translation models. All of these learning methods for Statistical MT were founded upon our contributions on learning models falling under the Fragment Model family where Phrase-Based SMT and phrase-based hierarchical SMT models belong, as well as those on Cross-Validated MLE estimation and the Cross-Validated CV-EM algorithm. Here, building upon the further findings and empirical observations of Chapters 4 and 5, we contribute a method to learn and apply a latent, linguistically motivated hierarchical translation structure. To this end, we take advantage of source-language linguistic annotations to motivate instead of constrain the translation process. An input chart over phrase-pair spans, with each cell filled with multiple linguistically motivated labels, is coupled with the HR-SCFG design to arrive at a rich synchronous grammar with millions of structural rules and the capacity to capture complex linguistically conditioned translation phenomena. We address overfitting issues by cross-validating climbing the likelihood of the training data and propose solutions to increase the efficiency and accuracy of decoding.

Our existing approach could be additionally fine-tuned to further improve performance, with an interesting direction being smoothing the HR-SCFG grammar estimates. Learning translation and reordering behaviour with respect to linguistic cues is facilitated in our approach by keeping separate phrase-pair emission distributions per emitting non-terminal and reordering pattern, while the employment of the generic X non-terminals already allows backing off to more coarse-grained rules. Nevertheless, we still believe that further smoothing of these sparse distributions, e.g. by interpolating them with less sparse ones, could in the future lead to an additional increase in translation quality.

An interesting aspect of our work is delivering competitive performance for difficult language pairs such as English-Chinese with a joint probability generative model and an SCFG without ‘gap rules’. Instead of employing hierarchical phrase-pairs, we invested in learning the higher-order hierarchical synchronous structure behind translation, up to the full sentence length. While these choices and the related results challenge current MT research trends, they are *not* mutually exclusive with them. Future work directions include investigating the impact of hierarchical phrases for our models as well as any gains from additional features in the log-linear decoding model.

In Chapter 5, we discussed how our CV-MLE learning approach could be successfully deployed to estimate the parameters of a hierarchical translation model. This model, despite its focus on unlexicalised abstract translation structure, still shared some common features, like the small number of bilingual categories, with the heuristically estimated Hiero baseline. While achieving competitive performance on real-world translation tasks is highly noteworthy on its own, the work presented in this chapter goes further than merely providing an alternative, better-founded method to estimate the parameters for grammars similar to those used by the already established, heuristically trained hierarchical SMT models. Instead, we believe that the true potential of our approach lies in going *past* current synchronous grammar designs, allowing SMT practitioners to *learn* hierarchical models for which the heuristics seem to provide weaker estimates: the system we present here is, to our knowledge, the first which provides strong translation performance using a synchronous grammar emphasising the use of abstract, un-lexicalised translation structures. We hope that the open-ended character of our method, which allows the incorporation of further external linguistic, semantic or other cues as well as alternative grammar designs, will provide a potent framework to progress further in deciphering the recursive nature of translation.

Chapter 7

Conclusions

In this thesis, we contributed a learning framework to uncover the latent structure that underlies multilingual data. Our methodology was founded on combining two well-understood learning approaches: Maximum Likelihood Estimation and the Expectation-Maximization algorithm on the one hand and Cross-Validation on the other. In this way we formulated the Cross-Validated EM algorithm (CV-EM), a principled method to cross-validate Maximum Likelihood Estimation (MLE) on incomplete data, which crucially retains the desirable algorithmic and estimation properties of EM. We used CV-EM as the theoretical foundation upon which we developed learning frameworks for three distinct translation models, each approaching the hidden structure of translation from a different perspective. Our implementations were empirically shown to perform at least on a par with state-of-the-art, heuristically trained baselines, significantly outperforming the latter as the translation models became more complex and the structural divergence between source and target languages increased.

Maximum Likelihood Estimation and the Expectation-Maximization algorithm are mainstays in the fields of Machine Learning and Natural Language Processing. Still, their application for modern, phrase-based translation models has proven highly challenging. In order to understand the reasons behind this, we showed that phrase-based models belong to a wider family of models for complex data: Fragment Models. This allowed us to delineate the source of both their modelling strength, as well as the pitfalls of estimating their parameters. We observed that, on one hand, the parameter space of a Fragment Model explores the full spectrum of different levels of abstracting away from the training data, ranging from deriving them from the smallest units to completely memorising them. At the same time however, this powerful feature is also their weakest point, as it exposes Fragment Models to a strong tendency to overfit training sets.

The Cross-Validated MLE (CV-MLE) learning objective and the CV-EM algorithm as the implementation of CV-MLE optimisation for incomplete data, directly address these learning challenges. They both cross-validate the space of our hypotheses on the values missing from the training data. For the translation models we examine, these values relate to the hidden structure of translation. Applying cross-validation aims to avoid overfitting and learn instead structures which generalise past the data included in the parallel corpus we are training on. Estimation within our framework stands out from competing approaches by combining a familiar and well-understood estimation objective, with a clear optimisation algorithm offering theoretical guarantees of operation.

Apart from this learning framework, this work contributed a methodology to learn models taking advantage of the linguistic structure of sentences to better translate. A recurring problem in prior work on translation models driven by linguistic syntax, was imposing unnecessary constraints on the translation process. The grammatical structure of sentences can explain only a subset of the translation phenomena, and assuming that all bilingual correspondences and transformations can be explained in grammatical terms can have a negative effect on translation performance.

In this thesis, we show how a translation-centric learning objective can be used to identify those linguistic cues that are useful for translation. We contribute a method based on the CV-EM algorithm to learn hierarchical translation models that offer linguistically motivated explanations of the recursive nature of translation across whole sentence-pairs. This is hardly a theoretical exercise, as our learnt models are found to significantly outperform a strong hierarchical translation baseline.

There is a number of interesting directions for future research directly emerging from this work. Here, following a stepwise approach, we chose to focus on learning translation models which separate lexical emission from abstract structure. However, we consider the induction of synchronous grammars which do not follow this separation, such as those making use of lexically grounded abstract rules (phrase-pairs with 'gaps'), as a highly interesting problem, involving a further examination of the interplay between lexical context and hierarchical structure. Another promissing direction is investigating further grammar designs and families of bilingual categories that can be effective in describing translation. As an example, one can explore different syntactic annotation schemes, such as head-lexicalisation or dependency structures, or move even further to facilitate synchronous structures based on semantic analyses of the source and target sentences. Finally, in this work we focused on taking advantage of unambiguous, single-best analyses of the parallel data to learn the latent structure of translation. It would be interesting to consider how we could exploit in our learning algorithms the ambiguous output that many natural language analysis systems provide, such as linguistic parse forests, as has already been done for heuristically extracted synchronous grammars.

While the empirical part of this work was exclusively focused on modelling translation, our theoretical contributions are applicable to any problem of modelling complex, structured data using incomplete training sets. The heart of our approach lies on addressing foundational problems in Machine Learning: examining how to disambiguate the composition of data from smaller modelling components of arbitrary sizes, and how to balance the ability of a model to memorise with its capacity to generalise. For this reason, we believe that our approach and the algorithms we propose can be applied to further tasks from the Natural Language Processing and Machine Learning domains. Promising examples are natural language parsing and the analysis of structured and semi-structured text, but also image parsing and financial fraud pattern detection.

Before closing this thesis, we turn back to our central theme: learning the structure of language. Each language is characterised by its own syntactic and semantic structure. Sampling sentences belonging to it and examining them side-by-side, can help to identify these latent structural patterns, whether this is work performed manually by linguists or automatically by Natural Language Processing models and algorithms. Interestingly, we could also consider natural languages themselves as data points. In this case, comparing how meaning is structured and encoded across different languages can point towards uncovering the patterns underlying natural language as a phenomenon of its own.

Efforts to manually identify such patterns and how they relate back to the various languages, such as some past interlingual translation approaches, have made little progress. However, as highlighted by the application of statistical methods in Machine Translation instead of manually compiled rule-sets, perhaps if it is difficult for humans to identify these patterns, we can instead try to learn how to automatically discover them in multilingual data. In this work, we contributed algorithms which search for this binding material that stands between human languages. We made small, careful steps, reaching up to examining how the linguistic structure of source sentences corresponds to the structure of target strings. Still, the open-ended character of our contributions on learning the latent structure of multilingual data, allows us to consider this work as a step in the direction of shedding some light on the patterns behind human language.

Bibliography

Pieter Adriaans and Ceriel Jacobs. 2006. Using MDL for Grammar Induction. In *Proceedings of the 8th International Colloquium on Grammatical Inference (ICGI)*, pages 293–306, Tokyo, Japan. Springer.

Pieter Adriaans and Paul Vitanyi. 2007. The Power and Perils of MDL. In *Proceedings of the 2007 IEEE International Symposium on Information Theory*, pages 2216–2220, Nice, France, June.

Alfred V. Aho and Jeffrey D. Ullman. 1969. Syntax Directed Translations and the Pushdown Assembler. *Journal of Computer and System Sciences*, 3:37–56, February.

Hirotugu Akaike. 1974. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19:716–723.

James K. Baker. 1979. Trainable grammars for speech recognition. In *Proceedings of the Spring Conference of the Acoustical Society of America*, pages 547–550.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Yehoshua Bar-Hillel. 1953. A Quasi-Arithmetical Notation for Syntactic Description. *Language*, 29(1):47–58.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, 41(1):164–171.

Matthew J. Beal. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Journal of Computational Linguistics*, 22(1):39–71, March.

Jeff Bilmes. 1997. A Gentle Tutorial of the EM algorithm and its application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report TR-97-021, ICSI.

Alexandra Birch, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 154–157. Association for Computational Linguistics.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008a. A Discriminative Latent Variable Model for Statistical Machine Translation. In *Proceedings of ACL-08: HLT*, pages 200–208. Association for Computational Linguistics.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008b. Bayesian Synchronous Grammar Induction. In *Advances in Neural Information Processing Systems 21*, Vancouver, Canada, December.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs Sampler for Phrasal Synchronous Grammar Induction. In *Proceedings of the 47th Annual Meeting of the Association of Computational Linguistics*, Singapore, August. Association for Computational Linguistics.

Rens Bod and Remko Scha. 1996. *Data-Oriented Language Processing: An Overview.* Research report nr. LP-96-13, ILLC Research reports, University of Amsterdam, Amsterdam, The Netherlands.

Rens Bod, Remko Scha, and Khalil Sima'an, editors. 2003. *Data Oriented Parsing.* CSLI Publications, Stanford University, Stanford, California, USA.

Rens Bod. 1992. A Computational Model of Language Performance: Data Oriented Parsing. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, Nantes.

Rens Bod. 1995. *Enriching Linguistics with Statistics: Performance models of Natural Language.* PhD dissertation. ILLC dissertation series 1995-14, University of Amsterdam.

Rens Bod. 2000. Parsing with the Shortest Derivation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000)*, pages 69–75, Saarbrücken, Germany.

Rens Bod. 2006. An All-Subtrees Approach to Unsupervised Parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING)*, Sydney, Australia, July.

Rens Bod. 2007. Is the End of Supervised Parsing in Sight? In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 400–407, Prague, Czech Republic, June. Association for Computational Linguistics.

Remko Bonnema, Paul Buying, and Remko Scha. 1999. A New Probability Model for Data Oriented Parsing. In Paul Dekker, editor, *Proceedings of the Twelfth Amsterdam Colloquium*, pages 85–90. ILLC/Department of Philosophy, University of Amsterdam, Amsterdam.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16:79–85, June.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19:263–311, June.

Jean-Cédric Chappelier and Martin Rajman. 1998. A Generalized CYK Algorithm for Parsing Stochastic CFG. In *Proceedings of 1st Workshop on Tabulation in Parsing and Deduction (TAPD'98)*, pages 133–137, Paris, France, April.

Eugene Charniak. 2000. A Maximum Entropy Inspired Parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*, pages 132–139, Seattle, Washington, USA, April.

Stanley Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Harvard University, August.

Yihua Chen and Maya R. Gupta. 2010. EM Demystified: An Expectation-Maximization Tutorial. Technical Report UWEETR-2010-0002, University of Washington.

Colin Cherry and Dekang Lin. 2007. Inversion Transduction Grammar for Joint Phrasal Translation Modeling. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 17–24, Rochester, New York, April. Association for Computational Linguistics.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 New Features for Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

David Chiang. 2005a. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June.

David Chiang. 2005b. An Introduction to Synchronous Grammars. Technical report, University of Maryland.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, June.

David Chiang. 2010. Learning to Translate with Source and Target Syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

Alexander Clark. 2001. Unsupervised Induction of Stochastic Context-Free Grammars Using Distributional Clustering. In *Proceedings of the Fifth Conference on Natural Language Learning*, CoNLL '01, Toulouse, France.

John Cocke. 1969. Programming Languages and their Compilers: Preliminary Notes. Technical report, Courant Institute of Mathematical Sciences, New York University.

Trevor Cohn and Phil Blunsom. 2009. A Bayesian Model of Syntax-Directed Tree to String Grammar Induction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 352–361, Singapore, August. Association for Computational Linguistics.

Michael Collins. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 175–182, Stanford University, CA, USA.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585, December.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City. Association for Computational Linguistics.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling Alignment Structure under a Bayesian Translation Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii, October. Association for Computational Linguistics.

Bonnie J. Dorr. 1994. Machine Translation Divergences: A Formal Description and Proposed Solution. *Computational Linguistics*, 20(4):597–633, December.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2001. *Pattern Classification*. John Wiley & Sons, NY, USA.

Jay Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13:94–102, February.

Jason Eisner. 2003. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan, July. Association for Computational Linguistics.

Heidi Fox. 2002. Phrasal Cohesion and Statistical Machine Translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 304–3111, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Stuart Geman and Donald Geman. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start). In *Proceedings of ACL-08: HLT*, pages 746–754, Columbus, Ohio, June. Association for Computational Linguistics.

John Goldsmith. 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27:153–198, June.

Joshua T. Goodman. 1998. *Parsing Inside-Out*. PhD thesis, Department of Computer Science, Harvard University, Cambridge, Massachusetts.

Peter Grünwald. 1996. A Minimum Description Length Approach to Grammar Inference. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 203–216, London, UK. Springer-Verlag.

Peter D. Grünwald. 2007. *The Minimum Description Length Principle*. The MIT Press.

Hany Hassan, Khalil Sima'an, and Andy Way. 2009. A Syntactified Direct Translation Model with Linear-time Decoding. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1182–1191, Singapore, August. Association for Computational Linguistics.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. 2001. *The Elements of Statistical Learning.* Springer.

Mary Hearne and Andy Way. 2003. Seeing the Wood for the Trees: Data-Oriented Translation. In *Proceedings of the Machine Translation Summit IX*, pages 165–172, New Orleans, Louisiana, USA, September.

Tom Heskes. 1998. Bias/Variance Decompositions for Likelihood-Based Estimators. *Neural Computation*, 10:1425–1433.

Hieu Hoang and Philipp Koehn. 2008. Design of the Moses Decoder for Statistical Machine Translation. In *ACL Workshop on Software Engineering, Testing, and Quality Assurance for NLP 2008.*

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical Syntax-Directed Translation with Extended Domain of Locality. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, Boston, MA, USA.

Liang Huang, Hao Zhang, Daniel Gildea, and Kevin Knight. 2009. Binarization of Synchronous Context-Free Grammars. *Computational Linguistics*, 35:559–595, December.

Zhongqiang Huang, Martin Cmejrek, and Bowen Zhou. 2010. Soft Syntactic Constraints for Hierarchical Phrase-Based Translation Using Latent Syntactic Distributions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 138–147, Cambridge, MA, October. Association for Computational Linguistics.

Abraham Ittycheriah and Salim Roukos. 2007. Direct Translation Model 2. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 57–64, Rochester, New York, April. Association for Computational Linguistics.

Fredrick Jelinek and Robert Mercer. 1980. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop on Pattern Recognition in Practice.*

Fredrick Jelinek and Robert Mercer. 1985. Probability Distribution Estimation from Sparse Data. *IBM Technical Disclosure Bulletin*, 28:2591–2594.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York, April. Association for Computational Linguistics.

Mark Johnson. 2002. The DOP Estimation Method is Biased and Inconsistent. *Computational Linguistics*, 28(1):71–76.

Mark Johnson. 2007. Why Doesn't EM Find Good HMM POS-Taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.

Tadao Kasami. 1965. An Efficient Recognition and Syntax-Analysis Algorithm for Context-Free Languages. Technical report, Air Force Cambridge Research Lab, Bedford, MA, USA.

Slava M. Katz. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP)*, 35(3):400–401.

Dan Klein and Christopher D. Manning. 2004. Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 478–485, Barcelona, Spain, July.

Reinhard Kneser and Hermann Ney. 1995. Improved Backing-Off for M-gram Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, volume 1, pages 181–184, May.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, May.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation 2005*, Pittsburgh, PA, USA, October.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation . In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit 2005*.

Ron Kohavi. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1143.

Karim Lari and Steve J. Young. 1990. The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm. *Computer Speech and Language*, 4(1):35–56.

Philip M. Lewis and Richard E. Stearns. 1968. Syntax-Directed Transduction. *Journal of the ACM*, 15:465–488, July.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An Open Source Toolkit for Parsing-Based Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March. Association for Computational Linguistics.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The Infinite PCFG Using Hierarchical Dirichlet Processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697, Prague, Czech Republic, June. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving Tree-to-Tree Translation with Packed Forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 558–566, Suntec, Singapore, August. Association for Computational Linguistics.

Adam Lopez. 2007. Hierarchical Phrase-Based Translation with Suffix Arrays. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 976–985, Prague, Czech Republic, June. Association for Computational Linguistics.

Daniel Marcu and William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Philadelphia, PA, USA, July. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Yuval Marton and Philip Resnik. 2008. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011, Columbus, Ohio, June. Association for Computational Linguistics.

I. Dan Melamed. 2003. Multitext Grammars and Synchronous Parsers. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 79–86, Edmonton, Canada. Association for Computational Linguistics.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-Based Translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June. Association for Computational Linguistics.

Robert C. Moore and Chris Quirk. 2007. An Iteratively-Trained Segmentation-Free Phrase Translation Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 112–119, Prague, Czech Republic. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2008. Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 630–639, Honolulu, USA, October. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2010. Learning Probabilistic Synchronous CFGs for Phrase-Based Translation. In *Fourteenth Conference on Computational Natural Language Learning*, pages 117–125, Uppsala, Sweden, July. Association for Computational Linguistics.

Markos Mylonakis and Khalil Sima'an. 2011. Learning Hierarchical Translation Structure with Linguistic Annotations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–652, Portland, Oregon, USA, June. Association for Computational Linguistics.

Makoto Nagao. 1984. A Framework of a Mechanical Translation Between Japanese and English by Analogy Principle. In *Proceedings of the International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, Lyon, France. Elsevier North-Holland.

Franz J. Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

Franz J. Och and Hans Weber. 1998. Improving Statistical Natural Language Translation with Categories and Rules. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 985–989, Montreal, Quebec, Canada. Association for Computational Linguistics.

Franz J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, June. Association for Computational Linguistics.

Franz J. Och. 1999. An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 71–76, Bergen, Norway. Association for Computational Linguistics.

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

John R. Pierce, John B. Carroll, Eric P. Hamp, David G. Hays, Charles F. Hockett, Anthony G. Oettinger, and Alan Perlis. 1966. Language and Machines: Computers in Translation and Linguistics. Technical report, Automatic Language Translation Advisory Committee, Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, D.C., USA.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised Morphological Segmentation with Log-Linear Models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 209–217, Boulder, Colorado. Association for Computational Linguistics.

Arjen Poutsma. 2000. Data-Oriented Translation. In *The 18th International Conference on Computational Linguistics*, pages 635–641, Saarbrücken, Germany.

Detlef Prescher, Remko Scha, Khalil Sima'an, and Andreas Zollmann. 2004. On the Statistical Consistency of DOP Estimators. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands (CLIN)*, Leiden, The Netherlands.

Detlef Prescher. 2004. A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars. *CoRR*, abs/cs/0412015.

Maurice H. Quenouille. 1949. Approximate Tests of Correlation in Time-Series. *Mathematical Proceedings of the Cambridge Philosophical Society*, 45(3):483–484.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, pages 271–279, Ann Arbor, Michigan, USA, June. Association for Computational Linguistics.

Jorma Rissanen. 1978. Modeling by Shortest Data Description. *Automatica*, 14(5):465 – 471.

Jorma Rissanen. 1983. A Universal Prior for Integers and Estimation by Minimum Description Length. *Annals of Statistics*, 11(2):416–431.

Satoshi Sato and Makoto Nagao. 1990. Toward Memory-Based Translation. In *Proceedings of the 13th conference on Computational linguistics - Volume 3*, COLING '90, pages 247–252, Helsinki, Finland. Association for Computational Linguistics.

Giorgio Satta and Enoch Peserico. 2005. Some Computational Complexity Results for Synchronous Context-Free Grammars. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 803–810, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Remko Scha. 1990. Language Theory and Language Technology; Competence and Performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek, LVVN-jaarboek (in Dutch)*, pages 7–22, Almere, The Netherlands.

Cullen Schaffer. 1993. Selecting a Classification Method by Cross-Validation. *Machine Learning*, 13:135–143.

Gideon E. Schwarz. 1978. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On Multiple Context-Free Grammars. *Theoretical Computer Science*, 88:191–229, October.

Claude Shannon. 1948. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:623–656.

Stuart M. Shieber and Yves Schabes. 1990. Synchronous Tree-Adjoining Grammars. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90, pages 253–258, Helsinki, Finland. Association for Computational Linguistics.

Takahiro Shinozaki and Mari Ostendorf. 2008. Cross-Validation and Aggregated EM Training for Robust Parameter Estimation. *Computer Speech & Language*, 22(2):185–195.

Khalil Sima'an and Luciano Buratto. 2003. Backoff Parameter Estimation for the DOP Model. In *Proceedings of the 14th European Conference on Machine Learning (ECML'03), Lecture Notes in Artificial Intelligence (LNAI 2837)*, pages 373–384, Cavtat-Dubrovnik, Croatia. Springer.

Khalil Sima'an and Markos Mylonakis. 2008. Better Statistical Estimation Can Benefit All Phrases in Phrase-Based Statistical Machine Translation. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT) 2008*, Goa, India, December.

Khalil Sima'an. 1996. Computational complexity of probabilistic disambiguation by means of tree-grammars. In *Proceedings of the 16th Conference on Computational Linguistics*, COLING '96, pages 1175–1180.

Khalil Sima'an. 2002. Computational Complexity of Probabilistic Disambiguation. *Grammars*, 5(2):125–151.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. 2005. Translating with Non-contiguous Phrases. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 755–762, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and Ralph Weischedel. 2006. A Study of Translation Error Rate with Targeted Human Annotation. In *In Proceedings of the Association for Machine Transaltion in the Americas (AMTA 2006)*.

Robert Tibshirani. 1996. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

Christoph Tillman. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

John Tukey. 1958. Bias and Confidence in Not Quite Large Samples. *Annals of Mathematical Statistics*, 29:614.

Matthew A. Turk and Alex P. Pentland. 1991. Face Recognition Using Eigenfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Maui, HI, USA. IEEE.

Menno van Zaanen. 2000. ABL: Alignment-Based Learning. In *Proceedings of the 18th Conference on Computational Linguistics*, COLING '00, pages 961–967, Saarbrücken, Germany.

Bernard Vauquois. 1968. A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation. In *IFIF Congress-68*, pages 254–260, Edinburgh, UK, August.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2009. Preference Grammars: Softening Syntactic Constraints to Improve Statistical Machine Translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–244, Boulder, Colorado, June. Association for Computational Linguistics.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *The 16th International Conference on Computational Linguistics (COLING 1996)*, volume 2, pages 836–841.

Ye-Yi Wang and Alex Waibel. 1998. Modeling with Structures in Statistical Machine Translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 1357–1363, Montreal, Quebec, Canada. Association for Computational Linguistics.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Restructuring, Re-labeling, and Re-aligning for Syntax-Based Machine Translation. *Computational Linguistics*, 36(2):247–277.

Andy Way. 1999. A Hybrid Architecture for Robust MT Using LFG-DOP. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(3):441–471.

David H. Wolpert. 1996. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403.

Dekai Wu. 2005. MT Model Space: Statistical Versus Compositional Versus Example-Based Machine Translation. *Machine Translation*, 19:213–227, December.

Kenji Yamada and Kevin Knight. 2001. A Syntax-based Statistical Translation Model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2002. A Decoder for Syntax-Based Statistical MT. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Daniel H. Younger. 1967. Recognition and Parsing of Context-Free Languages in Time n^3. *Information and Control*, 10(2):189–208.

Richard Zens, Franz J. Och, and Hermann Ney. 2002. Phrase-Based Statistical Machine Translation. In *KI 2002: Advances in Artificial Intelligence, 25th Annual German Conference on AI (KI 2002)*, volume 2479 of *Lecture Notes in Computer Science*, pages 18–32. Springer.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous Binarization for Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 256–263, New York City, USA, June. Association for Computational Linguistics.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008a. Bayesian Learning of Non-Compositional Phrases with Synchronous Parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008b. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In

Proceedings of ACL-08: HLT, pages 559–567, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Zollmann and Khalil Sima'an. 2006. A Consistent and Efficient Estimator for Data-Oriented Parsing. *Journal of Automata, Languages and Combinatorics (JALC)*, 10(2/3):367–388.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.

Willem Zuidema. 2007. Parsimonious Data-Oriented Parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 551–560, Prague, Czech Republic, June. Association for Computational Linguistics.

Index

Abstract

This dissertation discusses methods to learn the latent structural patterns that underlie translation data. It explores different approaches to modelling bilingual structure and presents novel frameworks and algorithms, such as Cross-Validated Expectation-Maximization (CV-EM), to learn phrase-based, hierarchical and syntax-driven Statistical Machine Translation (SMT) models from data.

In this thesis, we present methods to automatically learn phrase-based Statistical Machine Translation models that assume a latent bilingual structure as their central modelling variable. Acknowledging that each language is strongly characterised by its individual structural properties, we aim to learn a bilingual structure that augments and supersedes its monolingual counterparts, to bridge the gap between them by explaining the transformations taking place when conveying meaning across languages. The learning frameworks and algorithms we present allow us to discover these structural patterns in bilingual data and automatically learn models that take them into account to better translate. We apply our methodology for a sequence of statistical translation models of increasing complexity. This leads us to the presentation of a well-founded learning framework for hierarchical, syntactically motivated models that explain the translation process by taking advantage of the linguistic structure of language.

Chapter 1 offers an introduction to the context and aims of this work. It introduces the key aspects related to modelling translation structure and discusses the impact of its latent nature, as well as the challenges involved in learning to identify it in bilingual data. In Chapter 2, we start by examining some of the modelling frameworks that have been influential on SMT research, such as word-based, phrase-based and hierarchical SMT. We then discuss the EM algorithm and Cross-Validation, the two theoretical pillars under the novel learning algorithm we introduce in the chapter that follows. Chapter 3 examines the challenges related to learning phrase-based translation models, by considering the wider problem of learning Fragment Models: models which describe how to build new data instances by combining together data fragments extracted from a training dataset. We then introduce the Cross-Validated Expectation-Maximization (CV-EM) algorithm, a novel learning algorithm for Fragment Models which optimises parameters according to a Cross-Validated Maximum Likelihood Estimation (CV-MLE) objective.

The next three chapters describe and empirically evaluate learning frameworks with CV-EM at their core, for three distinct, state-of-the-art SMT models. Chapter 4 contributes a well-founded method to learn the conditional translation probabilities of Phrase-Based SMT models employing contiguous phrase-pairs, centred around disambiguating the latent segmentation of sentence-pairs into phrase-pairs. This method is shown empirically to perform at least as well as the heuristic, ad hoc estimators that are typically used for these models. In Chapter 5, we consider the additional challenges involved in modelling translation with a synchronous grammar, and successfully learn a relatively simple hierarchical translation model which offers comparable performance with a highly competitive baseline. Chapter 6 moves considerably further, to build around CV-EM a learning framework that allows learning complex hierarchical translation models that take advantage of external annotations of source and/or target sentences. We deploy this framework to contribute a method to learn linguistically motivated hierarchical translation models, by identifying the source-language linguistic patterns which are informative for translation. We subsequently show how our approach delivers tangible translation improvements across four distinct language pairs.

The results of Chapter 6 complete those of Chapters 4 and 5, to provide considerable evidence to back the key hypothesis of this thesis: models assuming a latent translation structure *can* be learnt under a clear learning objective, as implemented in terms of a well-understood optimisation framework and learning algorithm. The learnt models are able to provide real-world, competitive translation performance in comparison to heuristic training regimes, rendering the use of the latter unnecessary. Our methodology not only provides a reliable and effective substitute for these heuristic estimators, but most importantly lays a path to the future, by making possible the estimation of powerful translation models that uncover the latent side of translation, and whose estimation under ad hoc algorithms would have been hardly possible.

Samenvatting

Dit proefschrift behelst nieuwe methodes voor het leren van latente structurele patronen in vertaaldata. Het proefschrift bestudeert verschillende benaderingen voor het modelleren van tweetalige structuur, en presenteert een nieuw raamwerk en algoritmes, zoals Cross-Validated Expectation-Maximization (CV-EM), voor het leren van frase-gebaseerde, hiërarchische en syntactisch-gedreven statistische automatische vertaling (SMT) modellen uit data.

In het proefschrift presenteer ik methodes voor het automatisch leren van frase-gebaseerde SMT modellen die uitgaan van een latente tweetalig structuur als centrale variabele. Uitgaand van het feit dat iedere taal sterk gekenmerkt wordt door haar individuele structurele eigenschappen, streven wij ernaar om een tweetalig structuur te leren die in het verlengde ligt van zijn eentalige tegenhanger, met het doel de kloof tussen beiden te overbruggen door de transformaties die plaats vinden in het overbrengen van betekenis tussen talen expliciet te maken. Het leer-raamwerk en -algoritmes die worden gepresenteerd stellen ons in staat om deze structurele patronen te ontdekken in tweetalige data met als doel de gevonden patronen te gebruiken in vertaalmodellen die beter kunnen vertalen. Dit leidt to een wel-gefundeerd leerraamwerk voor hiërarchische, syntactisch-gemotiveerde modellen die het vertaalproces beschrijven middels de linguïstische structuur van taal.

Hoofdstuk 1 geeft een introductie voor de context en doeleinden van dit werk. Het presenteert de hoofdzaken betreffende het modelleren van vertaalstructuur en bespreekt zowel de impact van zijn latente aard als de uitdagingen in het ontdekken daarvan in tweetalige data. Hoofdstuk 2 begint met een uiteenzetting van sommige modellen die invloedrijk zijn geweest in SMT onderzoek, zoals woord-gebaseerde, frase-gebaseerde en hiërarchische SMT. Daarna worden de EM en Cross-Validation algoritmes besproken, de twee theoretische pijlers van het leeralgoritme dat wordt gepresenteerd in het volgende hoofdstuk. Hoofdstuk 3 bestudeert de uitdagingen van het leren van frase-gebaseerde vertaalmodellen, door het bespreken van het algemenere probleem van het leren van Fragment modellen: modellen die nieuwe data instanties bouwen door data fragmenten te combineren geëxtraheerd uit de training dataset. In het vervolg wordt het Cross-Validated Expectation-Maximization (CV-EM) algoritme gepresenteerd, een nieuwe leeralgoritme voor Fragment modellen dat parameters optimaliseert volgens de Cross-Validated Maximum Likelihood (CV-MLE) objectieve functie.

De drie hoofdstukken die hierop volgen presenteren en evalueren op empirische wijze drie state-of-the-art SMT modellen en hun leeralgorithmes die gebaseerd zijn op CV-EM. Hoofdstuk 4 presenteert een wel-gefundeerde methode voor het leren van conditionele vertaal-waarschijnlijkheden voor frase-gebaseerde SMT modellen die werken met onafgebroken frase-paren, met nadruk op het desambigueren van de latente segmentatie van zinsparen in strengen van frase-paren. Deze methode blijkt minstens even goed empirisch te werken als de huidige ad hoc estimatie methodes die doorgaans worden gebruikt met dit soort modellen. Hoofdstuk 5 bestudeert de bijkomende uitdagingen van het modelleren van het vertalen middels synchrone grammatica's, en laat zien hoe een relatief simpele hiërarchisch vertaalmodel met succes geleerd kan worden die vergelijkbare prestaties levert als een zeer concurrerende baseline. Hoofdstuk 6 maakt een significante stap in het bouwen van leeralgorithmes die extensies vormen van CV-EM, voor het leren van complexe hierarchische vertaalmodellen die profiteren van externe annotaties van zinnen in de bron-en/of doel-taal. We zetten deze leeralgorithmes in voor het leren van linguistisch-gemotiveerde hierarchische vertaalmodellen door het identificeren van de taalkundige patronen van de brontaal die informatief zijn voor het vertalen. Vervolgens laten wij zien hoe deze aanpak tastbare verbeteringen levert in vertaal kwaliteit in vier verschillende taalparen.

Hoofdstuk 6 completeert het werk in Hoofdstukken 4 en 5, en levert aanzienlijk bewijs ter ondersteuning van de belangrijkste hypothese van dit proefschrift: modellen die uitgaan van een latente vertaalstructuur *kunnen degelijk* worden geleerd onder een helder leerdoel, en geïmplementeerd middels een goedbegrepen optimalisatie raamwerk en leeralgorithme. De resulterende leermodellen geven competitieve vertaalprestaties in verhouding tot de gangbare heuristische training regimes, en maken het gebruik van deze regimes overbodig. Onze methodologie biedt niet alleen een betrouwbaar en effectief alternatief voor deze heuristische schatters, maar opent ook nieuwe wegen voor de toekomst, door het mogelijk maken van het schatten van krachtige vertaalmodellen die de latente kant van het vertalen blootleggen, en waarvan de schatting middels ad hoc algoritmes zou nauwelijks mogelijk geweest.

Titles in the ILLC Dissertation Series:

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder. . . ": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation