Refining translation grammars through paraphrase clustering

MSc Thesis *(Afstudeerscriptie)*

written by

Ekaterina Garmash
(born June 16th 1988 in Kiev, USSR)

under the supervision of **Dr. Khalil Sima'an** and **Gideon Maillette de Buy Wenniger, MSc**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
December 17th, 2012

Members of the Thesis Committee:
Dr. Alexandru Baltag
Dr. Khalil Sima'an
Dr. Raquel Fernandez
Gideon Maillette de Buy Wenniger, MSc
Prof. Dr. Rens Bod

Institute for Logic, Language and Computation

Abstract

Finding the right model for the structure of translation equivalence between languages is one of the major challenges and lines of research in statistical machine translation. In this thesis we consider a formalization of translation equivalence as synchronous grammars and explore a particular way of modifying a translation grammar – by labeling its nonterminal symbols. The labeling we develop is based on the general notion of semantic equivalence: since it is not know *a priori* what kind of semantic distinctions are relevant to translation equivalence, we define an unsupervised procedure to learn a label set by clustering close paraphrases that somehow characterize strings generated from a given nonterminal symbol. We implement the defined procedure and test a current baseline grammar (Hiero system [Chiang07]) labeled with a generated label set. By trying out a number of labeling algorithms and introducing additional modifications to the grammar, as well as making other changes to the standard translation pipeline, we find that the performance of the labeled grammar is worse than the one of the unlabeled. We discuss possible reasons for that and propose a number of modifications to the labeling procedure we defined and implemented here that could improve the performance.

Contents

Chapter 1

Introduction

The ultimate goal of machine translation (MT) is to automatically translate texts from one language to another at the same qualitative level that professional human translators do. The problem therefore consists in finding or defining *translation equivalence units* between languages and using them to find an optimal translation of an input text. By translation equivalence we understand a relation between linguistic units from two different languages such that they are translations of one another.

Intuitively, a translation is a text that conveys the same information as the original, and therefore it is reasonable to say that the "core" of translation equivalence relation is semantic equivalence. This implies that the task of automatic translation meets a major challenge of natural language processing in general – resolving ambiguity in a text: the original text has to be interpreted correctly in order to render it in another language. On the other hand, the output of translation has to be well-formed from the perspective of the structure of the output language. Thus, MT also subsumes the problem of natural language generation. Further, as with any machine learning problem, an automatic translation system has to be general enough so that it works for (ideally) any kind of text of a given language and is not biased towards a particular subset of cases. This latter point poses a very significant challenge, since natural languages contain a great number of patterns of high complexity.

The task of automatic translation can be approached analytically: one can manually construct rewrite rules from one language to another that would cover the whole language system, as in the *rule-based* approach. However, this approach is hardly capable of alleviating the challenges pointed out above: research in computational linguistics has shown that systems based solely on human expertise fail to embrace the complexity the whole language system. In order to take into account all (or as many as possible) cases, one may define an algorithm to automatically extract patterns directly from data. In empirical linguistics, data is large collections of texts – *corpora* – which one assumes to be a good approximation of a studied linguistic system (i.e. one assumes that the patterns found in a corpus and their distribution is close to the one in the whole actual linguistic system). In MT, one studies not just one linguistic system, but the correspondence between two languages, so the data is bilingual corpora: corpora consisting of texts in two languages being translations of each other.

Example-based machine translation works with a corpus of pairs of sentences translating each other (typically produced by human translations). Given these, translation correspondences between smaller units of text are extracted based on some definition of string correspondence. *Statistical* machine translation (SMT) does not just extract patterns from data collections, but attempts to model its probability distribution. It sees translation as a stochastic process where definitions of random variables are based on translation equivalence. A probabilistic model allows to define a function assigning probability scores to pairs of sentences as an estimation of the likelihood that they are mutual translations. This way we can get a decision procedure of choosing the best translation hypothesis, which is based not on the structural properties of the sentences but on the probabilistic model in which the structural properties and components of the sentences are values of the variables constituting it. In this thesis we work on an extension of an existing SMT model, and the rest of the introduction is dedicated to SMT.

1.1 Translation grammars in SMT

In this section we look at the structure of translation equivalence. There are three major paradigms of SMT with respect to the translation equivalence structure: word-based, phrase-based and hierarchical, each next one can be seen as a generalization of the previous one. We first describe them briefly, and then show that each of them can be formalized as a synchronous[1] *translation grammar* (although it is not typical to do so for the first two paradigms), in order to compare them and to explain why and how we want to refine then most advanced one (hierarchical translation grammar).

A translation hypothesis is constructed out of translation correspondence units – or, from the perspective of a translation grammar, is a result of a derivation produced by the translation grammar. *Word-based* SMT formalizes

[1] *Synchronous* meaning that it generates tuples of strings (in the case of translation – pairs – for both language sides).

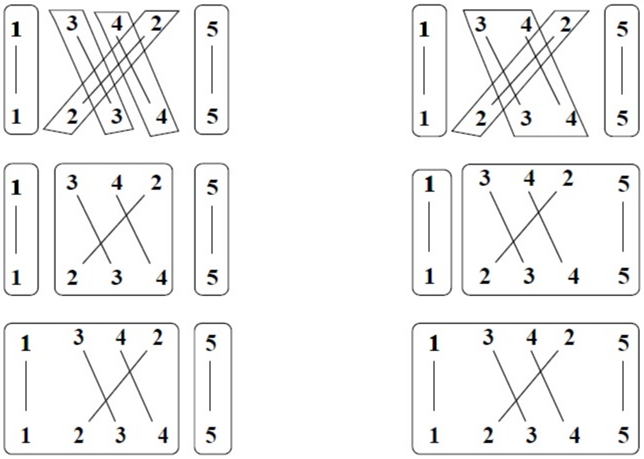

Figure 1.1: A set of possible phrase alignments of constrained by word alignments (represented by lines). (Picture taken from lecture notes to "Element of language processing and learning").

translation equivalence units as pairs of single words. Word pairs constitute *word alignment* of sentences. *Phrase-based* SMT extends the definition by allowing pairs of any contiguous strings of words which respect word alignments (i.e. for any word in the string all of the words aligned to it should be in this phrase pair). Figure 1.1 represents all the possible phrases extractable from aligned phrases, given their word alignments. A translation hypothesis in a a word-based or phrase-based translation grammar is a result of concatenation of translation units. The conceptual difference between word-based and phrase-based SMT is that the former assumes absolute compositionality: that a correct translation of a sentence can be composed of translations of its smallest subunits. With the phrase-based model, on the other hand, it is possible to take contextual information into account by letting the final translation hypothesis be composed of units of any length[2] and letting the probabilistic model decide about the best derivation. *Hierarchical* SMT incorporates the benefits of phrase-based SMT, but allows to generalize further. Construction of a translation hypothesis in hierarchical SMT is better described as a derivation in a grammar which generates strings with both terminal (nonsubstitable) and nonterminal (substitutable) symbols.

As an example, consider a German-English sentence pair *(Sie kam sofort an, She arrived immediately)*, with word alignment pairs (*sie, she*), (*kam, arrived*), (*sofort, immediately*), (*an, arrived*). Some of the possible rules in

[2] Typically length constraints are used in practice.

word-based (1.2), phrase-based (1.3) and hierarchical (1.4) grammars are given below.[3] We see that in order to make a derivation in a word-based and a phrase-based grammar we need additional rules of the form $X \rightarrow XX$ (for concatenation of substrings). For the hierarchical grammar, we can just concatenate strings of rules (a) and (j) and then apply rule (b).

a. $X \rightarrow \langle$sie, *she*$\rangle$

b. $X \rightarrow \langle$sofort, *immediately*$\rangle$

c. $X \rightarrow \langle$kam, *arrived*$\rangle$

d. $X \rightarrow \langle$an, *arrived*$\rangle$

Figure 1.2: Some possible rules in word-based grammar

e. ...everything from the word-based grammar...

f. $X \rightarrow \langle$kam sofort an, *arrived immediately*$\rangle$

Figure 1.3: Some possible rules in a phrase-based grammar

g. ...everything from the phrase-based grammar...

j. $X \rightarrow \langle$kam X an, *arrived X*$\rangle$

Figure 1.4: Some possible rules in an hierarchical grammar

From the above example we can also see that the hierarchical formalism allows to make better generalizations of the data. In a phrase-based framework there is really no way to learn that the German *an* is the so-called separable verb prefix[4]: the best it can do is learn different exemplars of the general rule

[3] These are not the only possible ones and that we do not specify a probability function over rules in order to demonstrate which derivation would be the optimal one. Our goal here is just to demonstrate how derivations are structured in the three different grammars.

[4] A separable verb prefix is a verb particle that is attached to a verb in some subset of syntactic contexts (for example, when the verb is in the infinitive form: *Sie wollte sofort* ***ankommen*** – *She wanted to come immediately*) and is separated (by other words of a sentence) in rest of the contexts.

(*kam sofort an*, *kam langsam an*, *kam plötzlich an*). The hierarchical model learns the pattern with a string *kam X an*, which is a generalization over the different examples.

A hierarchical grammar allows to introduce more abstract representations for the translation equivalence units by distinguishing between different nonterminal symbols. For instance, we may have translation pairs $\langle ADJ\ NP,\ NP\ ADJ\rangle$ for English-French; $\langle get\ X_{state},\ werden\ X_{state}\rangle$, $\langle get\ X_{object},\ bekommen\ X_{object}\rangle$ for English-German, where *werden* is 'become', and *bekommen* is 'receive'. In the next section we will show that refining translation grammar through labeling of nonterminal symbols can be beneficial from the machine translation perspective and motivate the contribution of the present thesis.

1.2 Refining one-nonterminal hierarchical translation grammar: syntactic and semantic perspectives

The current baseline in hierarchical SMT is a system called Hiero [Chiang07]. Its translation grammar has only one nonterminal X (in chapter 2 we give a formal definition)[5] , which implies that for any step in a derivation, if a current string contains X, then any rule can potentially be applied there (provided it fits into the terminal string that is being parsed). One straightforward way to alleviate the problem above is to *label* nonterminal symbols so that the set of rules that can be applied at any step in a derivation reduces. In this thesis we explore precisely this way of refining a hierarchical grammar. In this section section we explain what kind of labeling we want to employ.

In chapter 2 we review some recent literature on how to refine a one-nonterminal hierarchical grammar formalism through labeling. All of the reviewed proposals derive labels based on some syntactic definitions: their underlying hypothesis is that broad syntactic classes are good for specifying a restriction on the class of translation equivalence units that are applicable at some part of a derivation.

At the beginning of the chapter we said that semantic equivalence is the essential part of translation equivalence relation. Syntactic classes capture to a certain extent semantic equivalence. For example, such semantic roles, such as *agent*, *patient*, etc., is to a certain extent approximated with elabo-

[5] There is one more nonterminal, S, for so-called glue-rules (rules for concatenation of subtrees, but they have a meta-role in the grammar, since they do not contain terminal symbols.

rate syntactic labels: categorial grammar tags, in addition to syntactic class, also capture position relative to other phrases in a sentence. In this thesis we attempt to capture semantic equivalence of translation units in a more direct fashion: we group translation equivalence units based on some notion of semantic equivalence. The advantage of "proper" semantic labeling is that it captures more subtle cases of ambiguity. A possible inferiority in comparison to syntactic labeling is that it might overlook the syntactic well-formedness factor (one reason being that semantic classes are typically much narrower).

For an illustration of our motivations, consider an English-Russian language pair and suppose we have unlabeled rules (1.1 - 1.2) extracted from data:

$$X \rightarrow \langle X \text{ is the crown of creation}, X \text{ venec evoljucii} \rangle \tag{1.1}$$

$$X \rightarrow \langle \text{man, chelovek} \rangle \mid \langle \text{man, muzhchina} \rangle \tag{1.2}$$

The English *man* has at least two interpretations: 'human' and 'male'. In Russian there are two separate words for that (and no word that has both of the interpretations): *chelovek* ('human') and *muzhchina* ('male'). For the sake of illustration let us assume that 'male' interpretation is more frequent in the corpus. Then with rules (1.1 - 1.2), a more likely derivation for a sentence *Man is the crown of creation* would be (1.3-1.4) (i.e. 'Male is a the crown of creation') rather than (1.5-1.6).

$$X \rightarrow \langle X \text{ is the crown of creation}, X \text{ venec evoljucii} \rangle \rightarrow \tag{1.3}$$

$$\langle \text{man is the crown of creation, muzhchina venec evoljucii} \rangle \tag{1.4}$$

$$X \rightarrow \langle X \text{ is the crown of creation}, X \text{ venec evoljucii} \rangle \rightarrow \tag{1.5}$$

$$\langle \text{ man is the crown of creation}, \text{chelovek venec evoljucii} \rangle \tag{1.6}$$

If we could label rules in (1.2) as in (1.7) and (1.8) and have two separate rules (1.9) and (1.10) instead of (1.1), it would be very likely that rule (1.9) has a very high probability weight, and (1.10) – a very low one. Then the derivation combining rules (1.9) and (1.7) would be likely to get a higher probability score than the one combining (1.10) and (1.8).[6]

$$X_{L_1} \rightarrow \langle \text{man}, \text{chelovek} \rangle \tag{1.7}$$

[6] Application of separate rules are taken to be independent events, so the probability of their combination is obtained via multiplication of probability weights of individual rules.

$$X_{L_2} \rightarrow \langle \text{man}, \text{muzhchina} \rangle \tag{1.8}$$

$$X \rightarrow \langle X_{L_1} \text{ is the crown of creation}, X_{L_1} \text{ venec evoljucii} \rangle \tag{1.9}$$

$$X \rightarrow \langle X_{L_2} \text{ is the crown of creation}, X_{L_2} \text{ venec evoljucii} \rangle \tag{1.10}$$

This is a typical example of a problem caused by semantic ambiguity, which, as we pointed out at the beginning of the chapter, is very relevant in the field of MT. We showed that using labeling it is possible to distinguish between narrow semantic classes of words. Obviously, syntactic labels would not work here: they are too general to distinguish between such notions as 'human individual' and 'male individual'. One way to solve such a problem is to use some kind of semantic labeling: for instance, based on our example, one could use features as *male/female/undefined* for labeling. The problem is the already mentioned complexity of natural language: it is not known *a priori* what kind of semantic labeling is optimal and it hard to come up with one that would account for all the cases of ambiguity in a language. Therefore, for the same reason that data-based approaches in MT are more favorable than rule-based approaches, we would like to have a procedure that automatically clusters items that are of the same semantic nature, in a way that is relevant to translation equivalence relation.

In the next section we describe the strategy of unsupervised semantic labeling that we explore in this thesis.

1.3 Contributions

In this thesis we explore the following strategy of unsupervised semantic labeling: we do not come up with an ad hoc set of semantic labels, but define a general notion of similarity between translation pairs. The defined semantic similarity allows to derive a quantitative measure which is used to cluster translation pairs that are close enough. The resulting clusters are hypothesized to be semantic equivalence classes.

In the computational linguistics literature semantically close linguistic items are usually called *paraphrases*, and the task of extracting them *paraphrase extraction*. We illustrate with the same example from above how we want employ paraphrases and paraphrase clustering for translation grammar labeling.

Suppose we have a procedure to identify that a word *man* has paraphrases *human*, *human being*, *individual*, *person* (group 1) and *male*, *guy* (group 2). It is reasonable to expect that some of the paraphrases from group 1 will have high co-occurrence scores with the phrase *is a crown of creation*, but not the ones from group 2. On the other hand, it is likely that phrases from group 2 have high co-occurrence scores with a phrase like *are stronger than women*, unlike the phrases from group 1. All of the phrases from group 1 are translated as the Russian *chelovek*, and the ones from group 2 – as *muzhchina*. So if we label the rule (1.1) as (1.9), and devise a set of rules in (1.11), we get high probability scores for all of them and thus bias the decoding system to choose these rules for decoding *Man is the crown of creation*. Analogously, we can devise rules using labeling and grouping paraphrases meaning 'male' under the same label (1.12), in order to bias the system in favor of translating *Men are stronger than women* correctly.

$$X_{L_1} \rightarrow \langle\ \text{human},\ \text{chelovek}\rangle \mid \langle\text{individual},\ \text{chelovek}\rangle \mid \langle\text{man},\ \text{chelovek}\rangle \mid ... \tag{1.11}$$

$$X_{L_1} \rightarrow \langle\ \text{guy},\ \text{muzhchina}\rangle \mid \langle\text{male},\ \text{muzhchina}\rangle \mid \langle\text{man},\ \text{muzhchina}\rangle \mid ... \tag{1.12}$$

The implementation of this idea, which we are going to define in chapter 3, comes with some important limitations that we decided to make for the initial stage of the research for the sake of simplicity. First, ideally we want to cluster translation pairs to form classes of semantically equivalent units. However, the models that we define actually start by clustering phrases of one language and then transforming the cluster into a set of all phrase pairs containing the phrases from the cluster. Second, we do not estimate for each occurrence of a nonterminal the optimal set of translation units that are generated from it. Instead we use a uniform clustering procedure by defining a function that selects n best paraphrases given a list of all paraphrases and their similarity scores.

We summarize the original research presented in this thesis. It consists of two parts: abstract definitions of the clustering model and the empirical testing whether a label set generated according to the definitions can be used to improve performance of a baseline translation system. As an abstract model we define a procedure of generating a label for each occurrence of a nonterminal. Note that we do not define a labeling algorithm. Also note that although our goal is to improve the performance of a particular translation grammar (Hiero), the method in principle can be applied to any translation grammar. Our main contribution in the abstract modeling part is defining several alternative definitions of similarity measures between paraphrases. For the empirical testing part, we run experiments with different labeling algorithms. We also ran some additional experiments to investigate the impact of some of the design choices we made.

1.4 Outline

The rest of the thesis is structured as follows.

Chapter 2 contains the necessary background information about the fields that are involved in this thesis: statistical machine translation, semantic modeling, in particular definitions of semantic similarity, and literature on paraphrases and paraphrase extraction.

In *chapter 3* we define a procedure for generating labels for an hierarchical translation grammar.

In *chapter 4* we describe the experiments that we ran to test the performance of a labeled translations system. We discuss the experimental results and outline possible questions and modeling ideas for further research.

Chapter 5 lists conclusions.

Chapter 2

Background

The purpose of this chapter is to provide the reader with the background information related to and necessary for understanding the original research presented in the subsequent chapters. Our major task in this thesis is to define a procedure for generating labels for nonterminals of a hierarchical translation grammar.

Since the thesis about machine translation, we introduce the most important notions of the field in section 2.1. Paraphrases are informally defined as phrases conveying the same information [BannardCallison-Burch05], therefore more precise instantiations of this general concept has to rely on some notion of semantic similarity. In section 2.2 we make a brief overview of some important state-of-the-art models in computational semantics and the definitions of semantic similarity. In section 2.3 we concentrate on the definitions of paraphrases proposed in the literature and the corresponding methods for extracting them from textual data.

2.1 Statistical Machine Translation (SMT)

The goal of machine translation (MT) is, given a text in one language, to automatically produce a syntactically well-formed text conveying the same information in another language. Currently most MT systems translate a text sentence by sentence. An input sentence (the one which is translated) is called *source* sentence, the output (result of translation) is called *target* sentence. It is a convention in the state-of-the-art SMT, which we will follow, to use f (French) for source text and e (English) for target in formal notation.[1]

[1] This is because first SMT models were tested on French-English parallel texts.

The process of translation in MT is called *decoding*. It relies on a translation grammar – a set of productions specifying how to translate a source text to target text. Decoding consists in constructing a space of derivations, each producing some translation given a grammar and an input sentence, and based on that deciding on the best translation. Thus, a complete translation system must consist of at least two following elements: a translation grammar (supplied with a method to learn it, in case it is not manually created) and a decoding algorithm.

As a field MT originated with a *rule-based* approach, whereby the decoding algorithm was based on explicitly defined syntactic-oriented rules and manually-created dictionaries. A new stage of development of MT came with *example-based* MT [Nagao84]. Its innovation was the source of learning the translation correspondences – a bilingual parallel corpus. A *parallel corpus* is a collection of parallel texts: texts the building units of which are aligned to each other. *Alignment* is a relation between linguistic units (paragraphs, sentences, phrases, words) that are approximately semantically equivalent or, more specifically in our case, translationally equivalent. A *bilingual parallel corpus* consists of texts in two different languages aligned between each other.

Input data in example-based MT (which is usually called translation memory) is sentence-aligned: sub-sentential alignment is achieved by inspecting "minimal pairs" of sentences in one language (sentences that differ only with respect to some proper substring of words) and comparing their translations. As a simplified example, consider sentences *He **loves** summer*, *He **hates** summer* and their respective French translations *Il aime l'ètè*, *Il déteste l'ètè*: it can be inferred that *he* is translated as *il* and *summer* – as *l'ètè*, because these French words repeat in both sentences, just as the English originals; and, correspondingly, *loves* is *aime* and *hates* is *déteste* – because these French words are the substring that make the two sentences different, just as the English originals. The decoding process consists in finding the most similar sentence from the memory for the source sentence and the most similar strings for the parts that do not match, and then using the translations of the matching strings to produce the target sentence (with an algorithm sketched above). Importantly, the choice of matching parts in the memory is based only on string similarity (typically, an optimal translation is the one that requires the least number of string combinations).

Statistical MT is characterized by deeper analysis of bilingual data: it uses probabilistic modeling to learn bilingual correspondence patterns from parallel corpora. A translation grammar comes together with a probabilistic model which assigns a probability value to every derivation built up from the grammar. Besides translation equivalence units a probabilistic model typically includes language model probability, which is usually defined as an n-gram model.

The task of an SMT system is to find the best translation based on set of possible derivations and their scores[2].

Although we have stated that a SMT system consists of several components (translation grammar, probabilistic model, decoding system), we will not review them separately, since throughout the development of SMT models, the evolution of one of the components was always to some extent motivated by the others. That is why we will try to keep the chronological order as the major one in our overview. It is common to single out three major "evolutionary stages" of SMT models: word-based, phrase-based, hierarchical. We will review them one by one and give for each of them a description of their main components. However, the scope of this thesis is the structure of a translation grammar, so we will give more attention to this aspect.

2.1.1 Word-based models

Word-based SMT is represented by five IBM models [Brownetal1993]: they all share the same modeling idea and increase in complexity from first to fifth. We will focus our attention on the first model, since it is enough to illustrate the most important concepts of word-based SMT.

In [Brownetal1993] the authors define a generative probabilistic model in which translation from English to French is seen as a basic process. On an intuitive level, the authors take a semantic perspective: the model they build up can be seen as a process of a native French speaker generating sentences in French so that his "mental representations" are English sentences. They introduce a semantically inspired notion of *cept*: a subset of positions in an English string together with a sense or concept that they reflect. A cept generates French words. On a surface level cepts are realized as *word alignments* – connections between a pair of source and target words which translate (parts) of each other. Word alignment as a relation is not one-to-one: it is often the case that more than two words on one side correspond to a single word on the other side, and even more complex patterns occur. However, the IBM model narrows down its attention to individual words and connections between them and makes an assumption that each English word is aligned to at most one French word.

By transforming $Pr(e|f)$ into $\frac{Pr(e)Pr(f|e)}{Pr(f)}$, we get a decision rule for choosing

[2] The common decision rules are: the most probable translation (i.e. the one that maximizes derivations sum), the one that is obtained from the most probable derivation.

a target e_{best} given source f is defined as in (2.1), which is referred to as the *noisy channel model.* The benefits of reformulating the decision rule problem from $Pr(e|f)$ to $Pr(e) \cdot Pr(f|e)$ is that it allows to separate the problem of finding an English string that is conveys the meaning of, or generates, a source French string ($Pr(f|e)$ – *translation model probability*) and the problem of making sure that the target English string is syntactically well-formed ($Pr(e)$ – *language model probability*). We will focus on the former.

$$e_{best} = \text{argmax}_e Pr(e|f) = \text{argmax}_e Pr(e) \cdot Pr(f|e) \tag{2.1}$$

$Pr(\mathbf{f}|\mathbf{e})$ can be rewritten as $\sum_{\mathbf{f}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$, where $\mathbf{f} = f_1^m = f_1...f_m$, $\mathbf{e} = e_1^l = e_1...e_l$, $\mathbf{a} = a_1^m = a_1...a_m$, and a_i stands for a position in a target string to which f_m is aligned. The authors decompose the latter formula as in (2.2):

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = Pr(m|e) \prod_{j=1}^{m} Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) Pr(f_j|a_1^j, f_1^{j-1}, m, \mathbf{e}) \tag{2.2}$$

In the first IBM model it is assumed that $Pr(m|e)$ is constant (equal to ϵ), and $Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})$ depends only on l (length of the English string), and the attention is concentrated on $t(f_j|e_{a_j}) \equiv Pr(f_j|a_1^j, f_1^{j-1}, m, \mathbf{e})$, which is estimated as the relative frequency of (e, f) connections with respect to all e occurrences. Thus:

$$Pr(f, a|e) = \frac{\epsilon}{(l+1)m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \tag{2.3}$$

Summing over all possible alignments for the given source and target strings, they get a formula of translation probability. The unsupervised learning of the word alignments is done via the expectation-maximization algorithm [Dempster$etal$77], where the maximized function is t.

2.1.2 Phrase-based models

From a conceptual point of view word-based SMT implies the idea of absolute compositionality: translation of bigger strings can be seen as a sum of its smallest parts. This kind of approach has a lot of counterexamples in the linguistic literature: it has been long observed that in language there are word combinations that do not semantically reduce to their building parts (idioms and collocations being examples of different degrees of the general phenomenon of non-compositionality). *Phrase-based* SMT (PBSMT) alleviates this problem by considering probability distributions over bilingual string correspondences of different sizes. From the perspective of PBSMT, source and target strings consist of phrases, and word alignments in a string are taken to be *constraints* on which phrases it consists of:

Definition 1 (Phrase pairs [Zensetal02])**.** *Given a word alignment A between two strings f_1^J and e_1^I, a phrase pair is a pair of substrings f_j^{j+m} and e_i^{i+n} such that:*

- *both have no gaps;*
- *both are consistent with A: for all $f_k \in f_j^{j+m}$ all the e_l such that $(f_k, e_l) \in A$ are in e_i^{i+n}; and analogically for all $a_l \in e_i^{i+n}$.*

A minimal phrase pair (e, f) is such that there is no phrase pair (e', f') where either e' is a substring of e, or f' – a substring of f [SimaanWenniger12].

Definition 2 (Phrases in SMT)**.** *A (minimal) phrase is an element of a (minimal) phrase pair from definition 1.*

Under the noisy-channel model, in order to estimate translation probability of a source phrase given a target phrase it is now marginalized over all possible segmentations (consistent with word alignments) B of the phrases [Zensetal02]:

$$Pr(f_1^J|e_1^I) = \sum_B Pr(f_1^J, B|e_1^I) = \sum_B Pr(f_1^J|B, e_1^I)Pr(B|e_1^I) \tag{2.4}$$

The probability distribution over possible segmentations may be assumed uniform [Zensetal02], and the probability of a source phrase given a segmentation and a target is estimated as ([Koehnetal03]):

$$Pr(\overline{f_1^I}|\overline{e_1^I}) = \prod_{i=1}^{I} \phi(\overline{f_i}|\overline{e_i})d(a_i - b_{i-1}), \tag{2.5}$$

where $\overline{f}_1^I$ designates a particular segmentation of a phrase f_1^J into I phrases (likewise for e_1^J), f_i (analogously e_i) stands for i-th phrase in a segmentation, ϕ is a phrase translation distribution, and $d(a_i - b_{i-1})$ is distortion probability where a_i is the start position of a given source phrase and b_{i-1} is the end position of a source phrase which is translated into $(i-1)$th target phrase. $\phi(\overline{f}|\overline{e})$ is estimated as $\frac{\#(\overline{f},\overline{e})}{\#\overline{e}}$.

2.1.3 Hierarchical models

Translation grammar is further sophisticated by learning correspondences between more abstract patterns than just word strings: *hierarchical* SMT [Chiang05] does that by explicitly formalizing translation correspondence as a synchronous context-free grammar ([AhoUllman]). For example, if we have a pair of Chinese-English sentences in (2.6 - 2.7), where square brackets designate phrase alignments[3], we might, for example, not only extract a phrase pair (*de shaoshu guojia zhiyi*, *one of the few countries*), but also translation pairs (*[1] zhiyi*, *one of [1]*), where *[1]* is a placeholder for a phrase and "1" is an index and designates alignment (translation correspondence) between two units. This way we can define a rule $X \to \langle X_1 \textit{ zhiyi}, \textit{one of } X_1\rangle$.

[Aozhou] [shi] [yu] [Bei Han] [you] [bangjiao] $_1$[de shaoshu guojia zhiyi](2.6)
[Australia] [is] [dipl. rels.]$_1$ [with] [North Korea] [is] [one of the few countries](2.7)

This model can capture translation correspondence between units of different levels of syntactic abstraction. Importantly, the introduction of the nonterminal symbols allows to learn reordering patterns more effectively. Languages typically differ in the relative order of syntactic constituents and not individual words: phrase-based model does not allow to generalize over reordering examples that involve the same type of syntactic constituents, while hierarchical model achieves the generalization by representing the constituents in question with a single non-terminal in question.

The commonly accepted baseline in hierarchical MT is Hiero translation system [Chiang07], we will refer to its translation grammar as the *Hiero grammar*:

Definition 3 (Hiero translation grammar)**.** *Hiero translation grammar is a synchronous context-free grammar (SCFG)* $(\mathcal{V}, \mathcal{T}_S, \mathcal{T}_T, \mathcal{N}, R, S)$*, where* $\mathcal{V}$ *is a set of terminal symbols,* $\mathcal{T}_S = \mathcal{T}_T = \{X_1, X_2, S_1, S_2\}$ *are sets of source and target nonterminals (respectively),* $\mathcal{N} = \{X, S\}$ *is a set of left-hand terminal symbols, such that* S *is a starting symbol.* R *is a set of rules of the form* $A \to \langle \gamma, \alpha, \sim\rangle$*, where* γ *is a source string,* α *is a target string,* $\sim$ *is a one-to-one correspondence between non-terminals in* γ *and* α*. The rules are of two types:*

1. *rules of the form* $X \to \langle \gamma, \alpha\rangle$*, where* γ *and* α *are strings of symbols from*

[3] Example from [Chiang05].

$\mathcal{V}$, $\mathcal{T}_S$, $\mathcal{T}_T$, with the following constraints [4]*:*

- *the length of the source string on the right-hand side is at most five symbols;*
- *each side has at most two nonterminals;*
- *nonterminals cannot be adjacent on the source side;*
- *a rule must have at least one pair of terminal symbols.*

2. *glue rules of the form $S \rightarrow \langle \gamma, \alpha \rangle$, where γ and α are strings of symbols from $\mathcal{T}_S$, $\mathcal{T}_T$, with the constraint that they contain two symbols.*

There are also constraints which are active at the stage of grammar extraction. Hiero grammar rules are extracted in two steps: first, phrase-pairs are extracted according to Def. 1 to form *initial rules* (with X on the left-hand side); second, sub-phrase pairs (if any) are substituted with X symbols. [Chiang07] imposes a constraint that “initial rules” have at most ten words on either side. Another constraint is that unaligned words are not included into initial rules.

In sections 2.1.1-2.1.3 we have seen that how the structural complexity of translation correspondence increases: the result is that there are more structural aspects that we might want to include into the probabilistic model. The noisy-channel model takes care only of the translation probability and of the language model probability, while for the expressive power of the Hiero grammar we could take into account more than that. [OchNey02] propose to interpolate an arbitrary number of properties, or feature functions, of an object in one model. They propose to directly model the probability $Pr(e_1^I|f_1^J)$ within the maximum entropy framework [Berger$etal$96]: in the framework, one specifies a set of M *feature functions* $h_m(e_1^I, f_1^J)$, $m \in \{1, ..., M\}$. Each feature function is accompanied with a weighting parameter λ_m and is interpolated into the whole model as follows:

$$Pr(e|f) = p_{\lambda_1^M}(e|f) = \frac{exp(\sum_{m=1}^{M} \lambda_m h_m(e,f))}{\sum_e exp(\sum_{m=1}^{M} \lambda_m h_m(e,f))}. \tag{2.8}$$

Maximizing the probability with respect to e we get a decision rule:

$$\text{argmax}_e Pr(e|f) = \text{argmax}_e \sum_{m=1}^{M} \lambda_m h_m(e,f). \tag{2.9}$$

[4] The motivation for the introduction of such constraints is *spurious ambiguity* – a situation where at decoding we get many distinct derivations which have the same values for different probability functions that characterize them (cf. log-linear model below). Another motivation is to reduce the search space of a decoder.

This model allows to take into account different weighted functions of source-target pairs (h_i), such as lexical probability (how well individual words are translated), distortion probability (how good the word-order correspondence is), rule penalty (a monotonously increasing function of the number of rules used in the CFG derivation), etc.

2.1.4 Extensions of the state-of-the-art SMT systems: structural perspective

In the previous three sections we have traced down how the complexity and detailedness of translation equivalence patterns increased from model to model. An important part of the current research in SMT is to define a model capturing translation equivalence even better. There are at least two ways to proceed with this goal which are taken in the literature:

1) adding additional feature functions to the log-linear model (sometimes based on additional patterns extractable from the data);

2) modifying translation grammar;

The first option is to directly manipulate the probability of a resulting derivation by taking into account additional structural features of aligned sentences. For example, [Lietal12] add an additional reordering model to the system, [Chiangetal09], [ZollmanVogel11] add distortion features, [GimpelSmith08] discuss different positional features.

In this thesis we are interested in the second option. To be more specific, we are interested in a particular kind of modification – the one that we call labeling: *labeling* of a grammar is assigning labels of some kind to rules which can be defined in grammar. As said in the previous section, the latest baseline system in SMT is usually taken to be Hiero (Def. **??**). As we stated in the introduction, our goal is to provide a labeling for the Hiero grammar, so here we will review the literature on labellings of this translation grammar.

An important reason why there is a general interest in extending, or more specifically, labeling the Hiero grammar is that, as we pointed out in section 1.2, it is able to capture syntactic patterns between linguistic units of different levels of abstraction: between terminal nodes (i.e. actual words) and variables standing for phrases. The drawback of the Hiero grammar is that it is too unrestricted: it has only one nonterminal symbol (X) that can appear in a string with terminals. This implies that in a derivation, when using a rule with X on the right-hand side the next choice of a rule can be any other non-glue rule that fits the pattern of the source string, and only the probabilistic decision rule will decide that the best derivation is. Instead, if there were more than one nonterminal symbol, the set of rules that could be employed at a given step of a derivation would be smaller. This idea underlies the labeling models that we will review. Generally, the algorithm for learning a labeling assumes that the baseline grammar is available and should include the following steps:

1. Take as input a parallel text parsed with the baseline grammar.

2. To each nonterminal symbol in the "parse forest" assign a label according some criterion. Since we are working with parallel texts, there is a choice about the source for labeling information: the labels can be assigned to source side and then copied to the aligned target side (or vice versa). The information for labels can be based on bilingual patterns;

3. Extract the new grammar from the given labeled parse forest.

The labelings proposed in the literature are typically based on the output of some theoretically-motivated tool: phrase-structure parser [MylonakisSimaan11], categorial grammar parser [MylonakisSimaan11], dependency parser [Lietal12], POS-tagger [ZollmanVogel11]. On the other hand, labels can be theory-independent, like, for example, reordering labels [MylonakisSimaan11], [SimaanWenniger12]. The set of labels used to extend the baseline grammar can be predefined (for example, labels can be the direct output of a tagging tool), or they can be coined on the fly. As an example of the latter approach, [ZollmanVogel11] produce labels from sequences of POS tags that cover the phrase span corresponding to the nonterminal. Finally, another important modeling decision is what part of the constituent that a given nonterminal dominates is taken into account for labeling. It can be the whole terminal phrase generated from a given nonterminal, but it can be a part of the phrase: often what is called a dominant head (as defined in dependency grammar) of a constituent is used for labelling [Lietal12].

2.1.5 Evaluation metrics for SMT

The most popular automatic evaluation metrics for SMT is BLEU [Papineni etal02]. It relies on a test data set and a set of its reference translation. The evaluation algorithm compares the translation output by the decoder with the reference translations and outputs a number between 0 and 1.

The algorithm computes modified precision scores for n-grams (1- to 4-grams), denoted by p_n, according to a formula:

$$p_n = \frac{\#\text{n-grams from candidate set in reference translations}}{\#\text{n-grams in reference translations}} \tag{2.10}$$

The final score is:

$$\log \text{BLEU} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \text{log} p_n, \tag{2.11}$$

where r is the length of the reference corpus, c is the length of the candidate set, w_n stands for an n-gram.

2.2 Automatic extraction of semantic information

Mutual paraphrases being sets conveying the same information [BannardCallison-Burch05], we can regard them as semantic equivalence classes and use them as semantic information in applications. The amount of research output on paraphrases and on semantic similarity in theoretical and computational linguistics is enormous and we will not attempt to give the overview of it here. The scope of the present work is very specific: it is how paraphrases are seen from and can be applied in machine translation. But before we go on to present the background on research on paraphrase extraction and application, we would like to spend some time on some relevant highlights in more theoretically-oriented research in computational linguistics. We concentrate on two notable approaches to unsupervised semantic modeling: distributional modeling and the perspective of multilingual learning.

2.2.1 Distributional semantic models

Inspired by [Harris54], there has been a great deal of research in computational modeling of linguistic meaning in terms of distributions in text. By a *distribution* of a linguistic unit we understand a set of linguistic units of a specified kind with which the former co-occurs in a corpus. The basic idea is that the structure of a language can be described in terms of distributions: for example, vowels are surrounded by consonants in a word, articles are elements that precede nouns, a word *dog* due to its semantics often co-occurs with words *bark*. A notorious disadvantage of this approach, which is relevant for paraphrase extraction, is that antonyms usually have close distributions: for example, it is very probable that words *open* and *close* co-occur equally frequently with a word *door*.

Distributional semantic models (DSMs) have as their core assumption that a generalization of the context in which a lexical unit occurs in a corpus can be a good approximation of the semantics of this unit. [Turney06] takes the distributional idea further to model relational similarity between parts of words. DSM defines *semantic space* as a vector space where dimensions (set of vector features) are formalized as co-occurrence counts with the most frequent lexical items from the corpus, and vectors themselves correspond to senses of *target words* – words for which we want to put into the semantic space. By co-occurrence we understand that two words co-occur in a context window (a set of words surrounding a word) of a specified size. n most frequent words (which constitute a *basis*) are chosen for vectors' features because, first, the co-occurrence counts with them provide more precise information and, second, because they make it possible to compare between different vectors.

[PadoLapata07] abstract over the existing DSMs and develop a general model of semantic space, which can be described in terms of what components are needed to construct it from data:

1. *Context* of a target word is formalized as a dependency path π, which is said to be anchored to a target word t if t is the "starting" vertex of the path (notation: π_t). Note that under this definition a context of one occurrence of t may consist of more than one paths (if there are more than one path available). The intuition is that the dependency formalism can capture rather deep semantic relations (if we have appropriate labels for edges of a dependency graph), unlike a simple "bag-of-words" approach, and is less oriented at surface syntactic properties. In addition to that one defines a context selection function *cont:* $W \rightarrow 2^{\Pi_t}$ (where Π_t is a set of all π_t) which specifies what information actually is taken into account.

2. A *basis* B and a basis mapping function $\mu : \Pi \rightarrow B$ (where Π is a set of all dependency paths).

3. A path value function $v : \Pi \rightarrow \mathbb{R}$ (in the simplest case this function assigns 1 uniformly). Further, for each target word type (i.e. a set of all occurrences of some word) a global co-occurrence frequency (generalized over the whole corpus) is determined via $f : B \times T \rightarrow \mathbb{R}$.

[ErkPado10] explore a modification of the above model: they do not include global co-occurrence frequency function into the semantic space. Their idea is to store separately contexts for each token (occurrence) and consider them different senses of a word. This way they perform sense disambiguation: for a given context, they activate only those exemplars that are sufficiently close (greater than some threshold θ) to the current context with respect to a predefined metric. The whole set of exemplars is designated as E and the set of exemplars activated in a current context s – $act(E, s)$:

$$act(E, s) = \{e \in E | sim(e, s) > \theta(E, s)\} \tag{2.12}$$

2.2.2 Semantics in the framework of multilingual learning

In the previous subsection we have outlined a class of models that represent semantics of linguistic units by modeling the context in which they appear in a corpus. Another source of information that can be used for approximating semantics of linguistic units are parallel corpora (defined in 2.1). Aligned elements in a parallel text are assumed to be approximately semantically equivalent, and so if we are able to trace down the regularities in how words (or other units) align to words on the other side of a parallel text, we might also capture semantic regularities. This idea is used in [DiabResnik02], where the authors use aligned French language as a "sense language" for English for the task of word sense disambiguation. The authors point out that the method has its problems: the sketched model basically assumes that the words of a "sense language" are monosemous. Another problem is that their algorithm of semantic disambiguation relies on word alignment, which is generally quite noisy. Interestingly, a similar idea is expressed in [Brownetal1993] (which we have already mentioned), where they also construct a generative model with conditioned e and conditioning f and explain their intuitions as when a native French speaks French he has a corresponding English translation (being some sort of semantic representation) in mind.

The ideas underlying [DiabResnik02] can be characterized as belonging to a more general field of *multilingual learning*, where POS tags, syntactic structure, morphological structure of one language is learned from its correspondence to another language ([SnyderBarzilay10]).

2.3 Paraphrase extraction methods

As a a result of their very broad definition (phrases conveying the same information), one can devise a lot of different methods to extract paraphrases. One way is to use linguistic theory and manually-created linguistic resources, such as dictionaries, thesauri, encyclopedias. But as was said above, we are interested in automatic, semi-supervised and unsupervised ways to extract paraphrases, so only such methods will be reviewed here.

Paraphrase extraction methods are typically based on an idea of semantic similarity, that is why they use essentially the same ways of capturing semantic closeness as the models discussed in section 2. There are two major approaches to paraphrase extraction. In one, expressions are considered mutual paraphrases if their have similar distributions in a corpus. In the second one, the criterion for paraphrase extraction is whether expressions are aligned to the same expression in the parallel text (an approach based on parallel corpora).

2.3.1 Distributional approach

Under the distributional approach phrases are considered mutual paraphrases if they have similar distributions in a corpus.

The idea of distributional similarity is realized in its simplest form in [PascaDienes05] and [Marton etal09]. The set of potential paraphrases is obtained by extracting n-grams from a monolingual corpus. For each occurrence of each n-gram a so-called *lexical anchor* (a context window of a fixed length, which is just a set of words). [PascaDienes05] do not compute the global co-occurrence frequency, but devise a metric to compare the set of lexical anchors for each two n-grams (potential paraphrases). [Marton etal09], on the other hand, use global co-occurrence function to build for each phrase its distributional profile (formalized as a vector). Paraphrase closeness is then estimated as vector similarity.

[LinPantel01] formalize context ,as well as paraphrases themselves as specially transformed dependency paths[5].

2.3.2 Approach using parallel corpora

This family of methods makes use of parallel texts to establish correspondence between two phrases from parallel texts: if there is such correspondence, then these two phrases are mutual paraphrases, since they convey the same information. There are at least two ways to define this correspondence. One can use parallel texts of the same language, as [BarzilayMcKeown01] do, to extract aligned phrases and consider them paraphrase. Or one can use translation to a

[5] The transformation consists in connecting a preposition to a phrase that is its dependent, deleting the edge to the preposition, inserting a new edge from the node that dominated the preposition to the phrase to which the preposition got attached, and, finally, labeling this new edge with the preposition under discussion. This transformation is done in order to capture only semantic relations between words.

different language as a pivot to establish the correspondence between phrases of the same language, as [BannardCallison-Burch05] do.

[BarzilayMcKeown01] use alternative translations of the same foreign text as data. They first perform sentence alignment between them, and then identify parts of the sentence that are the same and parts that are different: for example, two sentences *They* ***came*** *home* and *They* ***arrived*** *home* almost coincide except for the word in the middle – this kind of divergent strings are learned as paraphrases. The approach is sound from the conceptual point of view, but it meets a major problem in application – scarcity of data: multiple translations of the same text are in general not very frequent, and they mostly are alternative translations of fiction literature texts, which are typically available only for a small subset of languages.

A pivot-based approach in [BannardCallison-Burch05] solves the problem of scarcity of data discussed above. It uses phrase-alignments (as defined in phrase-based machine translation) between bilingual corpora to extract monolingual paraphrases. The idea is that if two phrases are aligned to the same foreign phrase, then they are mutual paraphrases. An important benefit of this approach is that one may use different language pairs to extract paraphrases for a given language: for example, if one needs a paraphrase system for Spanish, he may use Spanish-English, Spanish-German, Spanish-French, etc. bitexts. This fact is useful when we consider applications of paraphrasing in machine translation: when a decoder encounters an unknown word in a source sentence (the one that is absent in the translation model), this word may be substituted by its paraphrase, which is present in the translation model. Still considering the same application, the authors define the "best" paraphrase e_{best} which is to be substituted instead of the original one e_1 as:

$$\begin{aligned} e_{best} &= \text{argmax}_{(e_2 \neq e_1)} p(e_2|e_1) && (2.13) \\ &= \text{argmax}_{(e_2 \neq e_1)} \sum_f p(e_2|f, e_1) \cdot p(f|e_1) && (2.14) \\ &\approx \text{argmax}_{(e_2 \neq e_1)} \sum_f p(e_2|f) \cdot p(f|e_1) && (2.15) \end{aligned}$$

From this formula we can derive the similarity measure between paraphrases (note that the measure is not symmetric):

$$p(e_2|e_1) = \sum_f p(e_2|f) \cdot p(f|e_1) \tag{2.16}$$

This baseline is refined by taking syntactic information into account [Callison-Burch08]. First, one can constrain the definition of paraphrases by requiring that they have only those syntactic labels that the original phrase has in the corpus:

$$
\begin{aligned}
p_{SyntConstr_1}(e_2|e_1) &= p(e_2|e_1, S(e_1)) && (2.17)\\
&= \sum_f p(e_2|f, S(e_1)) \cdot p(f|e_1, S(e_1)) && (2.18)
\end{aligned}
$$

where S is a function for a phrase to a set of its syntactic labels in a corpus, and $p(f|e_1, S(e_1))$ is obviously equal to $p(f|e_1)$.

Alternatively, one can maximize the probability with respect to a syntactic label:

$$p_{SyntConstr_2}(e_2|e_1) = \text{argmax}_{s_i \in S(e_1)} p(e_2|e_1, s_i) \tag{2.19}$$

Finally, one can require the paraphrases to have the same syntactic label as the given occurrence of the original phrase. If the given syntactic model allows multiple labels for one occurrence, then the idea can be implemented in two ways: one can maximize the conditional probability with respect to a label ((2.20), where $CurrS(f_1)$ stands for a set of current syntactic labels of f_1), or one can sum over all of the current labels (this will give the same formula as (2.17), with $S(f_1)$ interpreted as the set of current syntactic labels).

$$p_{syntConstr_3}(f_2|f_1) = \text{argmax}_{s_i \in CurrS(f_1)} p(f2|f_1, s_i) \tag{2.20}$$

2.4 Summary and outlook

In this chapter we have reviewed the research in machine translation and computational semantics which sets the necessary background for the original research presented in this thesis.

In 2.1 we have reviewed the state-of-the-art models of SMT, with an emphasis on the translation grammar models, and their evaluation metrics. We also gave a brief overview of the current literature on sophisticating the grammar formalisms for translation – the line of research represented by this thesis.

In section 2.2 we presented a summary of the recent modeling attempts in the field of computational semantics. We drew a line between two kinds of approaches: the ones that approximate linguistic meaning based on the contexts of the units that carry the meanings, and the ones that capture meaning correlation by capturing how linguistic units of one (target) language are related (via translation correspondence) to other language(s).

In section 2.3 we reviewed the literature in computational linguistics on paraphrases and paraphrase extraction. We have shown that the research on paraphrases is a subfield of computational semantics and thus it also can be divided into two classes: some models take paraphrases to be phrases that appear in similar contexts, and other models define paraphrases as phrases aligned to the same elements in a parallel corpus.

With the background set, we can proceed with defining our own model. In the next chapter we specify a procedure for generating labels for an hierarchical translation grammar.

Chapter 3

Paraphrase clustering for label generation

The goal of the present thesis is define a procedure for generating a set of labels for an hierarchical translation grammar (we work with a particular case, the Hiero grammar). We generate a set of labels by considering each occurrence of a nonterminal in a parse forest of an unlabeled grammar and defining a label for this occurrence. The kind of labeling we want to arrive at is semantic-oriented, which in particular implies that the classes denoted by the labels are much narrower than syntactic labels used in the literature (reviewed in chapter 2). We stress that in this chapter we define label generation, and not the actual labeling algorithm. The latter will be considered in the next chapter, where we consider a number of alternative labeling methods: fully labeling the original grammar, labeling a restricted set of occurrences of nonterminals, and others. We start with an informal overview of all the steps needed to generate the labeling we want to arrive at. The sections that follow will provide a detailed and formal specification of the framework.

To informally describe the procedure of label generation, we need to introduce some additional terminology: intuitively, it is reasonable to have a concept of a *representative phrase for a nonterminal.* In general, labeling is about finding a good representative of the class of phrases that are to be generated from some nonterminal symbol. On the other hand, each X in a tree dominates a terminal phrase, therefore, it is reasonable to say that for each occurrence of X there is a phrase that represents it, or characterizes it (it may be the whole dominated phrase, or its subphrase, or some particular representation of the phrase) – we are going to use this second meaning. The idea behind our labeling procedure is then, given a symbol X in a tree, to cluster the closest paraphrases of a phrase that represents it. Therefore, a precise definition of the method to generate our labeling set requires specification of at least the following components:

- what to consider a representative phrase representing for a given constituent;
- how to define a paraphrase of a given phrase;
- what measure function to use to determine cluster membership;
- what clustering algorithm to employ.

The following sections give a detailed specification of the needed components. Section 3.1 specifies the models that we use as an initial step of our clustering: the grammar that we will label – we will refer to it as a *basis for labeling* (the Hiero grammar) and a model for choosing a representative phrase for a non-terminal (CCL model). Section 3.2 defines an algorithm for choosing a cluster of closest paraphrases of a given phrase. First, it gives two general definitions of paraphrase similarity functions and derives from it a paraphrase similarity measure. Second, it defines the clustering algorithm itself. Sections 3.3 gives the simplest implementations of the two general definitions of paraphrases: we call them *basic models.*

3.1 Determining a starting element of a paraphrase cluster

In this section we define formally the models that we will use to produce the first step of the labeling: a grammar that is a basis for labeling and a model which determines for each occurrence of a nonterminal in the basis grammar a representative phrase (in the sense defined above). We conclude this section with a precise specification of our framework.

3.1.1 The baseline model

As a baseline we use a translation grammar Hiero from [Chiang07], the definition for which we repeat here:

Definition 4 (Hiero translation grammar)**.** *The Hiero translation grammar is a synchronous context-free grammar (SCFG)* $(\mathcal{V}, \mathcal{T}_S, \mathcal{T}_T, \mathcal{N}, R, S)$*, where* $\mathcal{V}$ *is a set of terminal symbols,* $\mathcal{T}_S = \mathcal{T}_T = \{X_1, X_2, S_1, S_2\}$ *are sets of of source and target nonterminals (respectively),* $\mathcal{N} = \{X, S\}$ *is a set of left-hand terminal* *symbols, such that S is starting symbol. R is a set of rules of the form $A \rightarrow \langle\gamma, \alpha, \sim\rangle$, where γ is a source string, α is a target string, $\sim$ is a one-to-one correspondence between non-terminals in γ and α. The rules are of two types:*

1. *rules of the form $X \rightarrow \langle\gamma, \alpha\rangle$, where γ and α are strings of symbols from $\mathcal{V}$, $\mathcal{T}_S$, $\mathcal{T}_T$, with the constraints on the form and size of source and target strings specified in Definition 3;*

2. *glue rules of the form $S \rightarrow \langle\gamma, \alpha\rangle$, where γ and α are strings of symbols from $\mathcal{T}_S$, $\mathcal{T}_T$, with the constrain that they contain two symbols.*

3.1.2 Choosing a representative phrase for a nonterminal

A straightforward solution for choosing a representing phrase would be to take the whole terminal phrase that is generated from a given non-terminal in a tree. Such a model might not work for our purposes (clustering), since the likeliness of a phrase to have a paraphrase decreases with its length. As a result, the implemented model will be sparse and might not work well. Also, such a solution neglects one of the major generalizations in linguistics – the "recursiveness" of language, since it treats each phrase as defining its own separate category. Projected onto the setting we are working in (hierarchical translation grammar), the idea of recursiveness of language can be spelled out as: each phrase has a subphrase that captures its essential properties. If this hypothesis is correct, then, given a selection criterion for a dominant element, it can be a first step of generalizing the set of categories. The second step would be (in our case) clustering of phrases, to reduce the set of categories even further.

For the purpose of selecting a dominant element, we might preprocess the data with some existing tool, such as a dependency parser, in order to choose a "dominant" subphrase for each phrase. However, existing tools typically work with syntactic information, while in the introduction chapter we showed that our idea is capture is more narrow semantic regularities. Thus, a syntactic tool might be too crude for our goals. Another problem is that not all tools are available for any language.

In the following subsection we present a model for labeling of the Hiero grammar proposed by Gideon Maillette de Buy Wenniger. The model is fully unsupervised, is based solely on Hiero representations and involves choosing one phrase in each rule (constituent) for labeling a non-terminal.

Core context labels (CCL): an information-theoretic approach to labeling of synchronous hierarchical representations

The CCL model takes an information-theoretic perspective and chooses a phrase with the most unpredictable translation to label a rule with it. The labels that are produced by the model are called *core context labels*, since, supposedly, they capture the core of the phrase for which the labeling is done. In order to single out a phrase with the most unpredictable translation, a probabilistic model is defined.

The extensions of the Hiero grammar that we reviewed in the background chapter are typically based on the structural properties of one of the languages (source or target). This approach is well-motivated, since interlingual differences are often quite systematic. On the other hand, the translation grammars that we reviewed (from word-based to hierarchical) have bilingual structures (word pairs, phrase pairs, etc.) as their basic units. The CCL model continues this latter approach of directly formalizing the interlingual correspondence.

The CCL model formalizes an event space as consisting of events of a particular source phrase having a particular corresponding target unit. A random variable is formalized as a source phrase ranging between different translations:

Definition 5 (Random variable represented by a phrase)**.** *A random variable* $\boldsymbol{var}_f$ *represented by a source phrase* $\boldsymbol{f}$ *ranges between translation pairs in which the first element is* $\boldsymbol{f}$ *and the second element is any* $\boldsymbol{e}$ *phrase. The probability distribution of* $\boldsymbol{var}_f$ *is defined as the conditional probability of a target phrase* e *given* $\boldsymbol{f}$*:*

$$Pr(var_f) = \{Pr(e|f)|e \in \textit{set of target phrases in a corpus}\}$$

As an example, suppose a French word f_1 is aligned to e_1, e_2, e_3 in a bilingual corpus: the values of the corresponding variable are $\{\langle f_1, e_i\rangle | e_i \textit{ is in the corpus}\}$, and their probabilities are $\frac{\#(f_1 \leftrightarrow e_1)}{\# f_1}$, $\frac{\#(f_1 \leftrightarrow e_2)}{\# f_1}$, $\frac{\#(f_1 \leftrightarrow e_3)}{\# f_1}$ and zero for the rest of the values.

The CCL model restricts the set of variables to those which are represented by *minimal phrases*. We repeat the definition from the background chapter: a minimal phrase does not contain a proper subphrase. Non-minimal phrases are taken to be complex objects whose probability distribution is a function of the distributions of the minimal phrases of which it consists: it is defined to be the distribution of the minimal phrase with the highest entropy.

Definition 6 (Random variable in CCL)**.** *A random variable in a CCL model is a random variable represented by a source phrase (Def. 5) which is a minimal phrase.*

Definition 7 (Probability distribution of phrase in CCL)**.** *A probability distribution of a phrase **f** is equal to the one of the minimal phrases constituting it with the highest entropy, defined as:*

$$H(var_f) = -\sum_{x_i \in var_f} Pr(x_i) \log Pr(x_i) = -\sum_{\langle f, e_i \rangle} Pr(e_i|f) \log Pr(e_i|f),$$

where $e_i \in \{$*set of target phrases in a corpus*$\}$ *and the random variable* var_f *is defined in Def.5.*

The basic idea behind the CCL-based labeling is to choose a minimal phrase which has the highest entropy among the minimal subphrases constituting a phrase dominated by a given nonterminal. However, we might want to exclude cases where a very rare phrase pair is chosen, even though formally it has a higher entropy than the rest of the minimal subphrases. For that reason the entropy measure is weighted with a probability of the given minimal phrase pair.

Definition 8 (CCL-based labeling for the Hiero tree representation)**.** *Every non-terminal in a Hiero tree is labeled with a phrase pair* $\langle f, e \rangle$ *which is derived from this non-terminal and has the highest rank among the rest of the minimal subphrases derived from this nonterminal with respect to the following measure:* $\boldsymbol{ambig}_f(\langle f, e \rangle) \times \boldsymbol{prob}(\langle f, e \rangle)$, *where* $\boldsymbol{ambig}_f$ *is the entropy of a variable represented by a source ("French") part of the phrase pair and **prob** is the probability of* $\langle f, e \rangle$*.*

Now we turn to the intuitions and motivations behind the CCL model. The linguistic interpretation of a phrase corresponding to a variable with the highest entropy is the phrase with the most unpredictable translation. If we think of different translations of a phrase as its different senses (cf. discussion of [DiabResnik02] in a background chapter), then we might say that a phrase with the highest entropy is the most ambiguous phrase – we will use this description throughout the text. In the introduction chapter we have discussed one of the major motivations for this thesis project – the need for semantic disambiguation with an example which contained an ambiguous word *man* (*Man is the crown of creation*). We have showed that it is desirable to label rules in such a way that the translations that these rules generate would differentiate between different translations of an ambiguous word. The CCL-based labeling differentiates between different translations of an ambiguous word by labeling the corresponding rule with both the source phrase and its current translation.

On the other hand, such a local approach to labeling has a disadvantage:[1] a core context label derived from a terminal node and its translation may "go up" high in the derivation tree, if the minimal phrase has a high ambiguity score. As a result, we get rules with labels consisting of phrases which are actually generated much later, in a different context. It follows then that at decoding we have to choose a translation of a very ambiguous word just based on the probabilistic model, without any context (words or phrases that are adjacent to the phrase in question in the derived sentence). On the other hand, in the field of statistical parsing lexicalization (labeling of nonterminal with some lexical information from the constituents) has been shown to be useful. However, in parsing lexical labels are additional to syntactic ones, i.e. they they further categorize already existing classes of rules, which is not the case in our setting.

We use the CCL-based labeling as a first step of our own labeling procedure. The hypothesis behind CCL labeling is that a minimal phrase pair chosen to represent a constituent is its good characterization in the sense that it splits the rules in such a way that the resulting best translations increase in quality. In our model we fix the source phrase of the chosen minimal phrase pair and then group together its closest paraphrases. Thus, what we are assuming in addition to the above is: close paraphrases characterize approximately the same sets of constituents.

3.1.3 Summary: a paraphrase-based framework of extension of Hiero translation grammar

Now we can describe in more detail the learning method for our paraphrase-based extension. We specify it in form of a high-level algorithm:

1. The input is a set of aligned sentences parsed with a trained Hiero grammar.

2. Each occurrence of X in the forest receives a CCL label as prescribed in Def.8. We only need the source-side element of a label.

3. For each occurrence of a CCL label, we take its source-side part f_1, and compute its similarity to the rest of the source phrases $sim(f_i|f_1)$. We then choose the subset of all the paraphrases based on their rankings to get a cluster label.

The sections below define models for the third step of the algorithm.

[1] Pointed out by Gideon Wenniger, by personal communication.

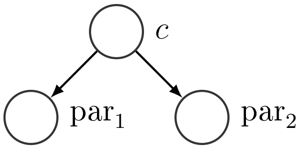

Figure 3.1: A model of type 1: c stands for a conditioning (causing), par_i stand for conditioned paraphrase variables

3.2 Paraphrase clustering: choosing a paraphrase model and a clustering method

Clustering can be described as grouping of elements of the same nature into classes. Clustering typically relies on some measure that characterizes some property of all the elements in the dataset, and based on it the closeness between elements with respect to this measure is determined. Different clustering methods define algorithms that determine for every pair of elements in a set whether they should be included in the same class. Thus, in order to specify our clustering method, we have to fix a measure function of closeness between phrases and also choose a clustering algorithm. The two following subsections do that.

3.2.1 Two types of models for a measure function

In this subsection we describe two different approaches that we take to derive a similarity measure between phrases.

Model of type 1

Under the first approach, which we will call *model of type 1*, a measure function is derived directly from a probabilistic model: it is conditional probability $Pr(\text{par}_2|\text{par}_1)$ (where *par* stands for "paraphrase"). We define a generative model which necessarily has the following elements: two conditioned variables, which represent paraphrases, and a conditioning variable (Figure 3.1). Conditioned variables specify which paraphrase exactly (an actual French phrase, or whatever we take to be a paraphrase in our formalization) occurs: the first variable stands for a paraphrase that is observed, the second one is an alternative phrase that could have occurred instead of the original phrase (that is, conditioned by the same value of the conditioning value). The interpretation of the conditioning variable could in principle be any reasonable information that we can extract from data (i.e. something on which a definition of paraphrases is based).

We can make a choice whether in our model the conditioning variable is observed – this will influence the final formula of conditional probability: (3.1) is a formula for the setting when the conditioning variable is unobserved , (3.2) is for when it is observed. The equality in (3.2) and in the last step in (3.1) is due to the principle of D-separation in a directed graphical model.

$$P(par_2|par_1) = \sum_c P(par_2, c|par_1) = \sum_c P(par_2|c, par_1) \cdot P(c|par_1) = \sum_c P(par_2|c) \cdot P(c|par_1) \tag{3.1}$$

$$P(par_2|par_1, c) = P(par_2|c) \tag{3.2}$$

In section 3 we describe the simplest model of this type in which a conditioned variable is interpreted as a source phrase, and a conditioning – as a target phrase.

Model of type 2

Under the second approach, we build up a vector space model in which paraphrases are represented as vectors. We then use a standard measure of vector similarity to determine paraphrase closeness. In this work we chose the cosine measure, which is a dot product of two vectors divided by their lengths:

$$cos(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|} \tag{3.3}$$

The feature space should somehow characterize the distribution of a given phrase: for example, with respect to translations or contextual information. In section 4 we describe a basic model of type 4, where the features of a vector are taken to be conditional probability between f (a phrase) and all the potential translations e_i.

An important difference from the conditional probability measure of type 1 model is that the current similarity measure is normalized with respect vector lengths. Another difference is that the cosine measure is symmetric, unlike the conditional probability.

3.2.2 Clustering algorithm

In this paper we decided to choose a simple clustering technique, one that allows to get a feeling of the whole framework that we are experimenting with.

Clustering can be characterized as global or local [ManningSchuetze02]. Clustering is global if it estimates classes by taking into account the whole data set. Local clustering methods estimate cluster with respect to a particular data point or data region. It is clear from the informal description of our method that we need local clustering: since we estimate for each data point (phrase) a set of its closest points.

A typical disadvantage of local clustering is that the final result might largely depend on the starting point of the clustering procedure. However in this paper, when estimating a cluster for a particular phrase, we disregard the clusters that have been already computed: each cluster is computed separately for each nonterminal occurrence. This entails that the clusters that we get in the end might intersect to a large extent.

We work with two similar clustering methods: the first one is based on *k nearest neighbors* (kNN), and the second one is a modification of the first one. kNN is an algorithm for estimating density of unknown data distribution [Bishop06]. Its assumption is that there is a (quite small) number k such for each data point x its nearest (according to some similarity measure) k elements are get the same value in some probability distribution. The most popular version of the kNN algorithm is used in supervised classification, in which an unlabeled element gets a label of the majority of elements among its k nearest neighbors. We use the general idea differently: from the spelled out assumption it follows that all the $k + 1$ elements are of essentially the same nature and therefore form a natural class.

Thus, the kNN algorithm for computing a cluster in our model is:

1. given data point (phrase) f_1, compute similarity $sim(f_i, f_1)$ for each f_i in the corpus;

2. order the phrases with respect to similarity score they got;

3. choose k best phrases with the respect to the ordering. In the case where the least score that gets into a cluster is such that if we do not add it, then there will be less than k elements, and if we add all the phrases with this score, then there will be more than k elements, – we choose the second option.

We modify the described method by letting k be function of the distribution of the paraphrase similarity measure. We set a lower bound p on the score that each paraphrase gets, and let only those paraphrases into a clusters that have a score above that bound – that is, we modify the third step of the algorithm.

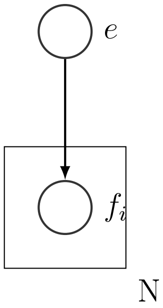

Figure 3.2: Basic model of type 1

The defined clustering algorithm has a variable parameter – k or p, and there is no *a priori* value for it. The optimization measure that we use is the final BLEU score that the whole translation system gets.

3.3 Basic models for type 1 and 2

The models of type 1 and 2 were defined abstractly: we did not specify what kind of data to use for the parameters of the models. In this section we give a simple instantiation of the two models based on the idea proposed in [DiabResnik02] and [BannardCallison-Burch05], and call them *basic models*. The idea behind the basic models is to use the definition of paraphrases from [BannardCallison-Burch05]: two phrases are paraphrases if they have common translations.

3.3.1 Basic model of type 1

Given that we will cluster phrases of a source language, as a basic model of type 1 we will assume the one that has a conditioning e variable and conditioned f variables. An e variable ranges between all the possible English phrases, and f variables – between French phrases (Fig. 3.2). This basic model relies on the already discussed hypothesis that different translations of a word or phrase correspond to a certain extent to its different senses [DiabResnik02].

Based on the abstract template derived in (3.1) and (3.2), the formulas of conditional probability will be (3.4) for a model with unobserved English variable, and (3.5) for when it is observed. We can see that the formula in (3.4) is the exactly the one used in [BannardCallison-Burch05]. Unlike the referred paper, now we have both intuitive and formal motivations for it.[2].

$$p(f_2|f_1) = \sum_e p(f_2|e) \cdot p(e|f_1) \tag{3.4}$$

$$p(f_2|f_1, e) = p(f_2|e) \tag{3.5}$$

3.3.2 Basic model of type 2

As it was said in section 2.1.2, the features of a vector should somehow characterize the corresponding paraphrase. We make the same assumption as for basic model of type 1: we assume that a translation of a phrase provides sufficient characterization of a phrase.

Unlike for the model of type 1, we do not have to make any assumptions about the causal relation between English and French phrases: all we need is to reflect the distribution of French phrases with respect to English phrases. Therefore, we can take a vector to be a list of probabilities $P(e_i|f)$, where f is fixed, and $e_i \in \{\text{all English phrases in a corpus}\}$. We use $v(f)$ to denote such vector for a phrase f. Then for every f_1 and f_2 their cosine score will be:

$$\begin{aligned}
cos(v(f_1), v(f_2)) &= \frac{\sum_e p(e|f_1)p(e|f_2)}{|v(f_1)|\sqrt{\sum_e p(e|f_2)^2}} && (3.6)\\
&= \frac{\sum_e \frac{\#(f_1,e)\#(f_2,e)}{\#f_1\#f_2}}{|v(f_1)|\sqrt{\sum_e p(e|f_2)^2}} && (3.7)\\
&= \frac{1}{\#f_1 \cdot \#f_2 \cdot |v(f_1)|} \frac{\sum_e \#(f_1,e)\#(f_2,e)}{\sqrt{\sum_e \#(e,f_2)^2}} && (3.8)
\end{aligned}$$

For the application that we are considering here – estimating closeness to phrase f_1, the value of the initial factor in (3.8) is constant for all paraphrases. Therefore, for this application, the model under consideration is equivalent to

[2] Actually in the original paper the equality is approximate:

$$\begin{aligned}
p(f_2|f_1) &= \sum_e p(f_2|e, f_1) \cdot p(e|f_1)\\
&\approx \sum_e p(f_2|e) \cdot p(e|f_1)
\end{aligned}$$

This fact is strange, since under assumption of model in Fig.3.2 the equality should be strict (by D-separation). On the other hand, in none of the papers where this paraphrase model is developed available to us is there any explicit discussion about what model is assumed.

the one in which vector features are defined as joint counts of a phrase with all possible translation.

Alternatively, we may define a model where feature values are conditional probabilities of a given phrase given an English translation, for all English translations – $P(f|e_i)$. We designate by $v'(f)$ a vector representing phrase f in this model. The cosine measure will look like:

$$\begin{aligned} cos(v'(f_1), v'(f_2)) &= \frac{\sum_e p(f_1|e)p(f_2|e)}{|v'(f_1)|\sqrt{\sum_e p(f_2|e)^2}} \qquad (3.9) \\ &= \frac{1}{|v'(f_1)|}\frac{\sum_e \frac{\#(f_1,e)\#(f_2,e)}{\#e^2}}{\sqrt{\sum_e \frac{\#(f_2,e)^2}{\#e^2}}} \qquad (3.10) \end{aligned}$$

For the same reason as above we disregard the initial factor. The major difference between the two presented versions of type 2 models is the fact that in the second one the joint counts are normalized with respect to English phrases. So, it is supposed to be more refined than the previous one.

3.4 Assumptions and predictions associated with the defined procedure

The ultimate goal we are pursuing in this thesis is to improve the performance of the Hiero translation system, as evaluated by the BLEU metrics. Therefore, the defined procedure for label generation has a number of implicit assumptions about how the the design choices influence the performance of the resulting grammar (some of which have been mentioned before). These assumptions lead to certain predictions and intuitions about how the defined models will perform. We list and discuss them here.

1. **Close paraphrases are representatives of the same class of phrase pairs**

   We assume that clusters of close paraphrases in general may provide a good restriction of the types of translation equivalence units applicable at a given part of a derivation. More specifically, the assumption is that close paraphrases represent (in the sense described above) approximately the same set of subtrees of a parsed corpus (and thus phrase pairs). Since we base our definition of paraphrases on translation equivalence relation, the hypothesis gets stronger (if not too strong): phrases which have similar distributions of their translations represent the same set of phrase pairs (having the representing phrase pair as its proper part).

2. **All paraphrase classes can be characterized with the same probability distribution**

   Our clustering algorithm is based on an assumption which could be a serious limitation of our whole framework. Our clustering algorithm, k nearest neighbors, assumes that there is a fixed k and a probability distribution such that for *each* data point its k nearest neighbors are assigned the same probability value. Applied to the case, it means that each paraphrase equivalence class is characterized by the same distribution of its elements.

3. **A minimal phrase with the highest ambiguity is a good representative phrase for a constituent**

   We made a choice to use the CCL model for defining a representing phrase. It is a very specific model and it is not clear whether it works well. In the next chapter we test also some alternative definitions.

4. **Differences between alternative design choices**

   There are some design choices that are not final, and we explore a number of alternatives, since we do not know in advance which is the best: language side based on which clustering is done, paraphrase similarity function model, number of members in a cluster (k). However, we do have some intuitions and predictions about how these parameters will work. If the CCL model is a reasonable model for choosing a representing phrase and our intuitions about it are correct, then we expect the model of Type 1 with both source and target states observed to perform better than the other two models, since the former performs sense disambiguation of a highly ambiguous phrase.

3.5 Summary

In this chapter we fully defined a procedure for generating labels for a hierarchical translation grammar. We started by choosing a definition for a phrase representing a constituent, which is a CCL model (one of the motivations being there is no immediately available alternative). We then proposed two types of models for paraphrase similarity measure: one is a generative probabilistic model, the other is a vector space model that uses probabilistic information for its features. We chose a clustering algorithm, which is kNN and its modification. Finally, we have outlined two simple implementations of the two proposed measure model types. The implementations employ solely phrase alignment information, in accordance with the paraphrase definition in [BannardCallison-Burch05]. Finally, we summarized the most important assumptions and predictions underlying the modeling choices that we made.

In the next chapters we test the defined procedure on real data. We stress again that in this chapter we only defined a way to generate labels for a given set of nonterminal occurrences: we did not fix the actual labeling algorithm (which occurrences to label, etc.). In the next chapter we test different algorithms. The strongest hypothesis is that label set generated by our procedure is alone sufficient to label the whole Hiero grammar and improve its performance. However, we will also investigate partial labeling and discuss possible combinations of our label set with others.

Chapter 4

Experiments

In this chapter we test the models for generating a label set defined in chapter 3. We are interested in two main questions. First, if a label set generated according to our procedure can in general be used to improve the performance of the Hiero system, and what are the optimal values of the variable parameters in the models that we employ. Second, if the generated label set is useful, then what is the optimal labeling algorithm: does a fully labeled grammar perform well, or only some partially labeled ones does.

4.1 Experimental setup

We tested our models on a task of translating from French to English. The two subsections below we give information about the training data and the implementation of the translation system that we used.

4.1.1 Data setup

For training of the system we used an English-French Europarl parallel corpus [Koehn05] consisting of 200K sentences. The size of a test set is 1K sentences. We used a parallel corpus consisting of 1K sentences for training the of parameters of the log-linear model using MERT [Och03] (we will refer to this set as the *dev* set). The test and dev set are taken from the WMT2007 data collection.

For grammar extraction, we performed "test-filtering" of the training set: we extracted only those rules the initial rules for which the initial rules have as their source side an n-gram present in the test set [Li$etal$09], [Callison-Burch$etal$05], [Lopez07]. For MERT training, the training set is filtered with the dev set in

Baseline (Hiero)
0.3077

Figure 4.1: BLEU score of the baseline system on the data settings from 4.1.1

the same way.

4.1.2 Implementation details

The translation pipeline used in the experiments consists of several units, provided by different sources. The translation grammar extraction unit is written in Java and is an extension of the implementation made by Gideon Wenniger. It consists of three consecutive subunits:

1. extraction of a mapping from minimal translation pairs to their ambiguity ranks – it is used for choosing the representing phrase according to the CCL model (cf. 3.1.2);

2. extraction of a mapping from "labeling positions" (i.e. X-nonterminals in rules) to paraphrase clusters;

3. extraction of a labeled translation grammar.

For MERT training of the log-linear probabilistic model, decoding and scoring of the hypothesis we used an implementation provided by an opensource SMT toolkit Joshua 4.0 [Ganitkevich$etal$12].

4.2 Experiments

In this section we describe experiments that we ran in order to test the performance of the Hiero grammar labeled with a label set generated according to the definition from chapter 3. The results of the experiments are given as the scores of the BLEU metrics, described in 2.1.5. We compare the performance of our models to the one of the baseline system – Hiero [Chiang07], since it is the one we are trying to improve upon labeling. We tested the pure Hiero system (Joshua 4.0 implementation) with the data settings described in 4.1.1: its resulting BLEU was 0.3077. For convenience of the reader (for better visibility) we put the result in Table 4.1.

4.2.1 Experiments with fully labeled grammars

In this section we report on the experiments with Hiero grammar fully labeled according to the specification in chapter 3: the labeling procedure is applied to every rule. That is, in this section we explore our working hypothesis in its strongest form: that labeling of rules based on a property of its subpart results in a more optimal search space during decoding.

The labeling that we defined contains a number of variable parameters:

1) language side (source or target) on which paraphrase clustering is based: in case of source-based clustering, for each nonterminal we fix a source part of the translation pair chosen according to the CCL model, and cluster its paraphrases; analogically for target-based clustering;

2) a function for k – a number of elements in paraphrase clusters.

For the first parameter, we experimented with both sides the language pair. For the second parameter, we report only on the function that sets a lower bound on the probability score that a paraphrase included in a cluster may have. We have tested some grammars with a function that directly fixes k, and it did not give significantly different results. Since lower bound-based function allows more direct manipulation of the set of resulting clusters, we will use it. We experimented with values of the lower bound: we tried both non-strict lower bound, which excludes only the most unlikely paraphrases (P = 0.2), and a strict one, which allows only very close paraphrases (P = 0.85).

Figures 4.1 and 4.2 provide BLEU scores for different combinations of the two parameters above for the different models defined in chapter 3. The notation should be understood as follows:

- `Type 1` is a basic model of type 1 (3.3.1). `orig.obs.` is a variant where only the causing variable (source, "original") is observed, `transl.obs.` is the one where both source and target variables are observed;

- `Type 2` is a basic model of type 2 (3.3.2): we experimented with the version where feature space is defined as conditional probability of target given source (for source-based clustering; reversed – for target-based).

We see that all of the scores are from 0.05 to 0.1 BLEU point lower than the baseline. This implies that the categorization that the models impose distribute probabilities between rules not in an optimal way. Below we attempt to trace the reasons of that, running some additional experiments. Before reporting on them, we briefly discuss the results from 4.1 and 4.2.

Type of model	P = 0.2	P = 0.85
Type 1, orig.obs.	0.3017	0.2995
Type 1, transl.obs.	0.3020	0.2979
Type 2	0.2983	0.3015

Figure 4.2: BLEU scores for labeling based on source-side (French) clusters

Type of model	P = 0.2	P = 0.85
Type 1, orig.obs.	0.2990	0.2995
Type 1, transl.obs.	0.2937	0.2974
Type 2	0.2975	0.2980

Figure 4.3: BLEU scores for labeling based on target-side (English) clusters

First, some of the best results that we obtained in all the experiments that we ran are in table 4.1 (0.3015, 0.3017, 0.3020). Interestingly, for Type 1 models these good results are for "all but worst" cluster lower bound, and and for Type 2 it is for "the very best" lower bound. But in general, it is hard to say whether the choice of a model (Type 1 or 2, which variables are observed) really makes a difference.

However, we do see a difference between source-based and target-based clustering: the former performs better than the latter. Moreover, the number of rules of source-based labeled grammar varies around 29.5M, and of target-based labeled grammar – around 17M. This possibly implies that the resulting clusters for target-based clustering overlap to a larger extent. Yet, target-based grammars perform worse, which might suggest that the employed clustering results in a non-optimal rule categorization. In the rest of the text, we will assume that source-based clustering captures data properties (for our purposes) better and will experiment only with it.

Rescoring output with labeled grammar

Since the labeled grammars do not perform well enough on their own, we checked whether they can augment the baseline model to rescore translation hypotheses. The decision procedure used in the Hiero system is to choose the translation hypothesis the derivation of which gets the highest score of the probability model function (the score is obtained with Viterbi approximation [Li*etal*10]).

It might turn out that although the labeled grammars perform worse on the whole, for some test sentences they give a better derivation. We checked that by combining n-best lists produced by the baseline grammar and by a labelled grammar. We ran this on a model of Type 1 orig.obs. and transl.obs., P=0.2. However, the resulting BLEU score was equal to the one of the baseline, which implies that none of the n best derivations produced by a labeled grammar beat any of the ones produced by the baseline.

In order to see whether a new labeled grammar can at least be useful to provide additional information for better scoring of hypothetical translations, we implemented a simple rescoring procedure in which the best translation is chosen according to the following procedure. Given a set of n best derivation hypotheses, one constructs a set of translation sentences that the derivations produce, labeled with the number of times it is produced by a derivation from the n-best set. The best translation is the one that is labeled with the greatest counts. In case there are multiple sentences with the (same) greatest count, one chooses a sentence among them that has a derivation with the greatest score. The hypothesis behind this method if there is a translation hypothesis that is produced by a lot of derivations, even suboptimal in terms of their scores, it might be a likely translation. However, it turned out that in "individual" n-best lists each sentence occurs only once, and so in the merged list – at most twice. The described rescoring gave suboptimal results, lower than the results for each of the models separately: for different models, it varied around 0.2700 BLEU (again, for Type 1 orig.obs. and transl.obs., P=0.2).

Sparsity issues

The number of rules that we get in the new grammars is much greater than in the baseline: for baseline it is around 3.3M, and for fully labeled grammars it is 29.5M for source-based labeling and 17M for target-based. Since the source-based grammars perform better than target-based and they have a much greater number of rules, it might be that the inferior performance of our models is due to data sparsity.

To check whether this is the case, we tested some of our models on a 0.5M training set and 1K test and dev sets. With these data settings, the baseline model scores 0.3180 BLEU point. We have tested the source-based models that performed the best in the 200K-corpus experiment. The number of rules now varies around 67M and the BLEU scores for some of the models are:

- Type 1, orig.obs., P = 0.85: 0.3039 BLEU;
- Type 1, orig.obs., P = 0.2: 0.3058 BLEU;
- Type 1, transl.obs., P = 0.2: 0.3042 BLEU.

We see that more data does improve the performance: which could imply that there are deeper reasons why the model does not perform well (the categorization of translation equivalence units is not optimal).

Redefining the choice of the representing phrase

We also experimentally investigated a question already discussed at length in chapter 3: what is a good definition of a representing phrase for a nonterminal. The CCL definition has two important properties: it always chooses a minimal subphrase, and this minimal phrase should be the most ambiguous one among the available. Here we experimented with the second property of the definition: we ran paraphrase-based grammar extraction where a representing phrase was chosen as a minimal phrase with lowest score with respect to the weighted ambiguity measure (as defined in Def.8). The results some of the models and see that changing the selection criterion among minimal makes a difference, so choosing a phrase with greater ambiguity seems to be reasonable:

- Type 1, transl.obs., P = 0.2: 0.3008 BLEU;
- Type 1, transl.obs., P = 0.85: 0.2130 BLEU;
- Type 2, P = 0.85: 0.2989 BLEU.

Summary on the fully labeled grammars

We have seen that fully labeled grammars make the performance of the translation system worse. The grammars are very sparse but increasing the data size does not help, which suggests, that the kind of labels and the labeling method do not refine a grammar in a way that biases the system towards better scoring function. We have seen, however, that the choice of the CCL model to define representing phrases is not unoptimal. Further, we have found that the variation of parameter values of the clustering model makes a lot of difference on the final BLEU result. This could suggest that the for each phrase in a corpus the set of potential paraphrases is very small. However, we did find that source-based clustering is better than target-based.

Since the fully labeled grammars did not perform well, we investigate other algorithms for labeling a Hiero grammar.

4.2.2 Modifying the labeling algorithm

In 4.2.1 we presented experiments with fully labeled grammars. Although their performance did not even reach the baseline, we pointed out some interesting findings, which we will investigate further here.

We found that a grammar fully labeled with paraphrase-based labels is very sparse because the number of labels is very high. We will introduce a modified grammar definition as an attempt to partially alleviate the sparsity problem. Also, we have investigated the impact of the CCL definition of representing phrase on the performance of the translation system: we experimented with an alternative definition in which a minimal phrase is chosen if it has the lowest weighted ambiguity score. In this subsection we modify the grammar extraction algorithm in such a way that we indirectly manipulate the impact of the fact that minimal phrases are selected.

Labeling with modification of the structure of the parse trees

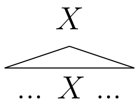

Figure 4.4: Unlabeled subtree

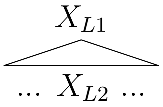

Figure 4.5: Subtree labeled according to an algorithm from chapter 3

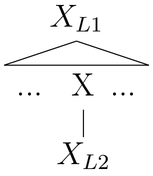

Figure 4.6: Subtree labeled with modified labeling from this subsection

This modification goes as follows. If there is a nonterminal in a tree which dominates another nonterminal, then the latter is not labeled directly: rather, it is substituted with a subtree consisting of an unlabeled nonterminal dominating a labeled nonterminal. For example, consider a subtree in Figure 4.4. Under the labeling algorithm we defined in chapter 3 and employed in subsection 4.2.1, its labeled version would be Figure 4.5, where L_1 and L_2 are

Type of model	P = 0.2	P = 0.85
Type 1, orig.obs.	0.2915	0.2904
Type 1, transl.obs.	0.2902	0.2909
Type 2	0.2921	0.2928

Figure 4.7: BLEU scores for grammar with unary rules

cluster labels computed as defined in chapter 3. Under the modified version of labeling we described here, the subtree would be transformed into 4.6.

Given this modified labeling procedure, the resulting grammar should consist of (apart from glue rules):[1]

- terminal rules, the left-hand side for which is necessarily labeled;
- non-terminal rules with the left-hand side labeled and right-hand side nonterminals unlabeled;
- non-terminal rules with unlabeled left-hand side and right-hand side consisting of a single labeled nonterminal.

With the help of this modification we make the rules of a grammar less specific, since only one side is labeled. This can be a partial solution to the sparsity problem. The results of the experiments with the modified grammar is given in Table 4.7 (we call it grammar with unary rules because it introduces special nonterminal to nonterminal unary rules). The number of rules indeed became smaller (around 10M), but, as we see, the results have got even worse. A possible explanation for this is that, in a derivation, when there is an unlabeled nonterminal which is to be substituted, there is nothing but the statistical model that sets the preference for the choice of rules: in other words, the application of a unary rule generating a labeled nonterminal is not conditioned or restricted by the structure of the string in which it is applied. In sum, we have "split" rules in order to reduce their detailedness, but the splitting resulted in a set of rules which carry no way to structurally identify a preference for them to be used at some point of a derivation. This problem could be solved by labeling the nonterminals which were previously left unlabeled with some wider categories (for instance, syntactic ones).

Type of model	P = 0.2	P = 0.85
Type 1, orig.obs.	0.3034	0.2975
Type 1, transl.obs.	0.3035	0.2975
Type 2	0.2986	0.2966

Figure 4.8: BLEU scores for restricted-3 grammar

Type of model	P = 0.2	P = 0.85
Type 1, orig.obs.	0.2998	0.3014
Type 1, transl.obs.	0.3005	0.3016
Type 2	0.3013	0.3013

Figure 4.9: BLEU scores for restricted-2 grammar

Restricting the set of labeled terminals by length of generated phrases

Another problem with the models we defined is that by choosing the minimal subphrase we capture very local information in a tree. For each representing phrase, we cluster phrases which are its translational equivalents: such equivalence classes are expected to be very narrow. That is why tagging a nonterminal dominating, for instance, the whole sentence with a label defined "locally" might not be a good idea.

Ideally, we would like to apply rules labeled with narrow paraphrase-based semantic information at some point close to the end of the derivation of a sentence: when the translation of a sentence is already known on a high-level, and there are now more subtle ambiguities to resolve. Therefore, we would like a grammar with nonterminal labels representing different linguistic levels: from broad syntactic to narrow semantic.

As an approximation to a desired model, we implemented the following labeling procedure: we set a restriction on the length of the source terminal phrase that is derived from a given non-terminal.[2] Such a restriction is a simple way of controlling how high in a tree a given nonterminal is. By restricting the labeling in such a way, we get labeled rules that specify derivations of short phrases – and thus we do not any more have the problem of local information being used as characterization of subtrees of any length.

[1] When we describe the right-hand side of rues below, it applies to both of elements of the right-hand side pair.

[2] There are perhaps smarter ways to restrict labeling to local parts of a parse tree. For instance, we could set a restriction on the depth of a tree. But due to time limitations we chose a reasonable option that was fastest to implement.

The length of a source phrase in general is restricted to five words in Hiero grammar [Chiang07], so we experimented with the restriction that a terminal source phrase derived from a given nonterminal is at most 3 words and at most 2 words. The results are in Figures 4.8 and 4.9. The results for Type 1 P = 0.2 are the highest results we got with our models in this thesis. We again see that the results of Type 1 models are very similar between each other and different the Type 2 model. Since we also observe that more restrictive clustering (P=0.85) gives worse results, one could suggest that the Type 2 model assigns very low similarity measures to paraphrases which results in very small clusters.

In sum, the fact that we got some very small improvement over the previous results that we got could suggest that paraphrase-based labeling is not hopeless. It would be interesting to see how the model with labels corresponding to different levels of linguistic generalization, sketched above, would perform. Alternatively, one could combine labels of different level in a different way: by "lexicalizing" broad syntactic classes with narrow semantic ones (by analogy to what is done in syntactic parsing).

4.3 Discussion of results and future work

The results of the experiment can generally be characterized as negative, since we did not manage to even reach the baseline results. In this section we summarize the insights and observations that we made throughout the chapter and discuss how they could be employed in future work.

First of all, we have to answer the main question: whether labels obtained based on paraphrase clusters can be used to improve the performance of the Hiero grammar. The answer is not clear, but the results of all the grammars refined with paraphrase clusters that we tested are inferior to the baseline. One of the reasons why paraphrase clusters do not provide good categorization of translation equivalences are the limitations pointed out in chapter 1 and chapter 3: the simplifying heuristics used for clustering and the fact that we actually cluster phrases and not translation units. Further, we found that it is difficult to manually evaluate which of the clustering model variants gives the best results. This might be due to the fact that in general a phrase in a corpus has very few potential paraphrases and so the different similarity metrics give similar scores. This problem can be alleviated if we actually cluster translation pairs, and not monolingual phrases, since the size of the data set would increase. In general, it would be useful if we could define some evaluation metric to make predictions about the resulting set of labels generated by a model.

Second, we tested whether the model for a representative phrase is good. We ran some experiments with an alternative definition which retained an important characteristics of the original one: it always chooses a minimal phrase. The results with the alternative definition were worse. However, it would be interesting to explore other definitions as well: for example, the ones that do not only choose minimal phrases. Notably, it would be interesting to use some existing tool, such as a dependency parser.

Third, we discussed a hypothesis that paraphrase-based clusters could be good for "local labeling". It was somewhat supported by the the fact that a special variant of labeling restricted by length of the source constituent gave the highest results among all the ones we got (but still almost 0.05 BLEU point below the baseline). We also saw that a grammar modification in which labeled nonterminals never appear on the right-hand side with terminal symbols and are generate by separate rules unlabeled on the left-hand side, performs much worse than other labeled grammars. This result could imply that application of left-hand side labeled rules need to be restricted with respect to the context. Therefore we suggested to use paraphrase-based labels in a modified labeling where both wide syntactic and narrow semantic categories are used.

Chapter 5

Conclusion

In this thesis we explore a particular way of labeling of an hierarchical translation grammar, based on automatic extraction of narrow semantic classes, and test the abstract model on the Hiero translation grammar (which has one nonterminal symbol). The general motivation for labeling a grammar is to restrict the set of translation units that can be generated from a given nonterminal symbol at a given stage of the derivation. While most models of translation grammar labeling proposals found in the literature use syntactic-based definitions to generate a set of labels, we developed a procedure for generating semantic-oriented classes for labeling. The motivation for using semantic labels are cases of translational ambiguity (caused by semantic ambiguity in one language and lack thereof in the other language). Our basic idea was to define an unsupervised way of generating labels by defining a similarity measure function between paraphrases and a clustering algorithm.

We realized our intuitive idea in a fully defined procedure for defining paraphrase-based labels given a data set parsed with an original (unlabeled) grammar. We pointed out some simplifications of an original idea: the fact that our models define clustering of monolingual phrases, based on their translations, instead of clustering translation units, and the fact that the clustering algorithm itself is is not optimized to fit the data. The definition of the procedure of generating labels consisted of model for choosing a "representing phrase" for each occurrence of a nonterminal: we chose a CCL model, but pointed out that alternative models could be tried out. Given a way to choose representing phrase, we defined a parametrized procedure for clustering closest paraphrases: we defined a number of alternative similarity measure functions, as well as chose a clustering algorithm for which one can tune the number of elements that the resulting cluster has.

We tested our model empirically by labeling a baseline grammar (Hiero) and running it on a standard translation task. Our first finding was that given the definition paraphrases that we use and the similarity measure between them, the differences between different variants of clustering models vary very insignificantly and it is difficult to manually make any generalizations about the impact of different parameter values. We pointed out that this could be due to the simplifications made at the modeling stage and the properties of the data (paraphrase equivalence classes according to the definition we are using are typically very small). We proposed that using a more complex definition of paraphrases (in particular, taking them to be phrase pairs, not monolingual phrases) and paraphrase similarity measure could be a solution to this problem. We did not spend much time on testing the impact of the definition of the "representative" phrase for a given nonterminal occurrence. Future work might consist in a more thorough research of this question, exploring both supervised and unsupervised definitions. Based on the experimental results with non-fully labeled Hiero grammar, we suggested that paraphrase-based labels could improve the performance of a translation system when combined with a more wide class-based labeling (such as syntactic labeling).

Bibliography

[AhoUllman] A. V. Aho, J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. Journal of Computer and System Sciences, 3:37–56.

[BannardCallison-Burch05] Colin Bannard and Chris Callison-Burch, 2005. Paraphrasing with Bilingual Parallel Corpora. In Proceedings of ACL-2005.

[BarzilayMcKeown01] Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the ACL/EACL*, pp. 50–57.

[SnyderBarzilay10] Snyder, B. and Barzilay, R. Climbing the tower of babel: Unsupervised multilingual learning, 2010, Omnipress.

[Berger*etal*96] Al.L. Berger, S.A. Della Pietra, V.J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39-72.

[Bishop06] Christopher Bishop. Pattern recognition and machine learning. Springer 2006.

[Brownetal1993] Brown, P.F. and Pietra, V.J.D. and Pietra, S.A.D. and Mercer, R.L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*,19(2).

[Callison-Burch08] Chris Callison-Burch. 2008. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In Proceedings of EMNLP 2008.

[Callison-Burch*etal*05] Chris Callison-Burch, Colin Bannard, Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceeding of ACL*.

[Chanetal11] Charley Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking Bilingually Extracted Paraphrases Using Monolingual Distributional Similarity.

[Chiang05] David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 63–270.

[Chiang07] David Chiang. 2007. Hierarchical phrase-based translation. *Computational linguistics*,33(2), pp. 201–228.

[Chiangetal09] David Chiang, Kevin Knight, Wei Wang. 2009. 11,001 New Features for Statistical Machine Translation. In *Proc. NAACL HLT*, 218–226.

[Dempster*etal*77] Dempster, A.P.; Laird, N.M.; Rubin, D.B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B (Methodological) 39 (1): 1–38.

[DiabResnik02] Mona Diab, Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02).

[ErkPado10] Erk, K. and Padó, S. Exemplar-based models for word meaning in context. *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, 2010, Association for Computational Linguistics.

[Ganitkevich*etal*12] Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, PRO, and Paraphrases. In Proceedings of WMT12.

[GimpelSmith08] Kevin Gimpel and Noah A. Smith. 2008. Rich source-side context for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pp.9–17.

[Harris54] Harris, Zelig. 1954. Distributional structure. *Word* 10(23), pp. 146–162.

[Koehnetal03] Koehn, P. and Och, F.J. and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*,pp. 48-54.

[Koehn05] Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit 2005.

[Li*etal*09] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, 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 Workshop on Statistical Machine Translation (WMT09).

[Li*etal*10] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Lane Schwartz, Wren N. G. Thornton, Ziyuan Wang, Jonathan Weese and Omar F. Zaidan. 2010. Joshua 2.0: A Toolkit for Parsing-Based Machine Translationwith Syntax, Semirings, Discriminative Training and Other Goodies. In Proceedings of Workshop on Statistical Machine Translation (WMT10).

[Lietal12] Junhui Li, Zhaopeng Tu, Guodong Zhou, Josef van Genabith.2012. Using Syntactic Head Information in Hierarchical Phrase-Based Translation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pp.232-242.

[LinPantel01] Dekang Lin and Lin Pantel. 2001. DIRT - Discovery of Inference Rules from Text. *Proceedings of ACM SIGKDD Conference on Generating Phrasal and Sentential Paraphrases Knowledge Discovery and Data Mining*, 323–328, San Francisco, CA.

[Lopez07] Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of EMNLP-CoLing.*

[MadnaniDorr10] Nitin Madnani and Bonnie J. Dorr. 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics*, 36(6).

[ManningSchuetze02] Christopher D. Manning and Hinrich Schütze. 2002. *Foundations of statistical natural language processing.* MIT Press.

[Marton etal09] Yuval Marton, Chris Callison-Burch and Philip Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proceedings of EMNLP 2009.*

[MylonakisSimaan11] Markos Mylonakis, Khalil Sima'an. 2011. Learning Hierarchical Translation Structure with Linguistic Annotations. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 642–652.

[Nagao84] Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle.

[OchNey02] Franz Josef Och and Hermann Ney. 2002. Discrimina- tive training and maximum entropy models for statis- tical machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pp. 295–302.

[Och03] Franz Josef Och. 2003.Minimum error rate training in statistical machine translation. In *ACL*, Sapporo, Japan.

[PadoLapata07] Padó, S. and Lapata, M. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2), pages 161–199, 2007, MIT Press.

[PascaDienes05] Marius Pasca and Peter Dienes. 2005. Aligning Needles in a Haystack: Paraphrase Acquisition Across the Web. *Proceedings of IJC-NLP*, 119-130.

[Papineni etal02] Papineni, K., Roukos, S., Ward, T., Zhu, W. J. 2002. BLEU: a method for automatic evaluation of machine translation. ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318

[SimaanWenniger12] Khalil Sima'an, Gideon Maillete de Buy Wenniger. Hierarchical translation equivalence over word alignments. Manuscript.

[Turney06] Turney, Peter. 2006. Similarity of Semantic Relations, *Computational Linguistics*, 32(3), pp.379-416.

[Weese*etal*11] Jonathan Weese, Juri Ganitkevitch, Chris Callison-Burch, Matt Post and Adam Lopez, 2011. Joshua 3.0: Syntax-based Machine Translation with the Thrax Grammar Extractor. In Proceedings of WMT11.

[Zensetal02] Richard Zens, Franz Joseph Och, Hermann Ney. 2002. Phrase-based statistical machine translation.

[ZollmanVogel11] Andreas Zollmann and Stephan Vogel. 2011. A word-class approach to labeling PSCFG rules for machine translation.

[Zollmann11] Andreas Zollmann. 2011. Learning Multiple-Nonterminal Synchronous Grammars for Statistical Machine Translation. Ph.D. Thesis.