SMOOTHING A PBSMT MODEL BY FACTORING OUT ADJUNCTS

MSc Thesis (*Afstudeerscriptie*)

written by

Sophie I. Arnoult
(born April 3rd, 1976 in Suresnes, France)

under the supervision of **Dr Khalil Sima'an**, 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:	**Members of the Thesis Committee:**
August 31, 2011	Dr Khalil Sima'an
	Prof Dr Rens Bod
	Prof Dr Benedikt Löwe

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

Phrase-Based Statistical Machine Translation (PBSMT) became a leading paradigm in Statistical Machine Translation after its introduction in 2003. From the start, one has tried to improve PBSMT by using linguistic knowledge, often by incorporating syntactic information into the model.

This thesis proposes a simple approach to improve PBSMT using a general linguistic notion, that of adjuncts, or modifiers: One expects that in structurally similar languages like French and English, adjuncts in one language are likely to be translated as adjuncts in the other language. After verifying this assumption, this thesis describes how adjunct pairs are deleted from a bilingual corpus to generate new training data for a model, which is then used to smooth a PBSMT baseline.

Experiments on a smoothed French-English model show only a marginal improvement over the baseline. It appears that few of the phrase pairs gained by adjunct-pair deletion are actually used in testing, so that improvement in performance mostly results from successful smoothing. Further research directions would be to find out in how far performance can be improved for this system, but also to apply adjunct-pair deletion to other language pairs and to hierarchical SMT models.

Contents

Chapter 1

Introduction

1.1 Machine Translation

Machine Translation has proved a difficult task since its beginning in the '50s. The simplicity of the task enunciation, translating text from one natural language to another, hides a complex process.

As a matter of fact, natural language translation is difficult even for humans: It requires both advanced if not fluent knowledge of the language to be translated from, or *source* language, and fluent knowledge of the language to be translated into, or *target* language. The domain of translation further requires a bilingual knowledge of the domains in question. Ideally, the target message is *equivalent* to the source message both semantically and pragmatically.

The first Machine Translation systems were rule-based and limited to restricted domains of translation. The growing need for the fast translation of large amounts of data, such as the Canadian Hansards and the European Committee Parliament proceedings, formed an incentive for research while providing it with large parallel corpora. Consequently, Machine Translation was able to move away from the rule-based paradigm, resulting in Example-Based Machine Translation (Nagao, 1984), and Statistical Machine Translation (Brown et al., 1990).

1.2 Phrase-Based Statistical Machine Translation

Statistical Machine Translation (SMT) decomposes the translation problem of a source sentence s into a target sentence t in a *translation task*, modeled by a conditional probability $P(s|t)$, and a *language task* modeled by $P(t)$. The latter ensures fluency of the target, while the former fits in the *noisy-channel approach*, where s is seen as the coded output of a noisy channel, and t as its input. The task of finding t is therefore regarded as a *decoding* task.

The *IBM models* proposed by Brown et al. (1990) regard the source and target sentences as strings of words, with an alignment mapping the target words to those of the source. See section 2.1 for a brief introduction to the models.

An obvious weakness of the IBM models resides in the alignment model, where two target words cannot generate a common source word. Och and Ney (2000, 2003) proposed to solve this issue by symmetrizing word alignments, as explained in section 2.2.

The symmetrization of word alignments paved the way for Phrase-Based Statistical Machine Translation (PBSMT, Koehn et al., 2003), where phrases are taken as translation units, instead of words. PBSMT collects phrase pairs that are consistent with the unified word alignments, and gathers these pairs in a *phrase table*, along with translation probability estimates. The PBSMT model is presented in section 2.3.

At decoding, the system retrieves from the table the phrase pairs whose source phrase constituents exactly match source phrases in the input sentences, then generates the target sentence from left to right: At each step of the decoding process, a new phrase pair is added to the current stack of hypotheses with its translation, distortion and language-model costs. The best hypothesis is the one that minimizes the combined costs of its phrasal hypotheses.

PBSMT builds upon the IBM models, as it uses their word alignments, and surpasses them as it allows to capture local reorderings, estimate translation probabilities of idiomatic expressions, and encode some contextual information[1].

1.3 Problem Statement

PBSMT suffers from a number of weaknesses: (1) It does not fare very well in modeling global reorderings and thus performs less well with language pairs with different word orders; (2) It cannot model discontinuous phrases; (3) Data is sparse, especially for longer phrase pairs, which are then overestimated by the unsmoothed heuristic counting estimator (Koehn et al., 2003); (4) It cannot model context across phrase pairs, as these are assumed to be independent of each other.

The issue of reordering has been tackled through syntactically-informed models: the source and/or the target are parsed to build tree-to-string (Marcu et al., 2006; Huang et al., 2006), string-to-tree (Carreras and Collins, 2009) or tree-to-tree systems (Quirk and Menezes, 2006a,b). Alternatively, in (Hassan et al., 2008) the target is parsed to enrich a PBSMT system with supertags.

While some of these methods allow to model discontinuous phrases (Marcu et al., 2006; Huang et al., 2006), the most successful model for this purpose has been the hierarchical Synchronous Context-Free Grammar (SCFG) model proposed by Chiang (2005). Like standard PBSMT models, this model is purely

[1] see (Quirk and Menezes, 2006b) for a detailed discussion

statistical, and can be enriched with syntactical information. For instance, Zollmann and Venugopal (2006) and Mylonakis and Sima'an (2011) utilize chart parsing to label non-terminals in a SCFG, while Chiang (2010) parses both source and target to build a Synchronous Tree-Substitution Grammar (STSG).

Smoothing has been proposed to improve probability estimations in the phrase table (Kuhn et al., 2006; Foster et al., 2006), and minimal phrase pairs to alleviate data sparsity: see the *tuples* of Schwenk et al. (2007) and the *Minimal Translation Units* of Quirk and Menezes (2006b). In both cases, these new units of translation are utilized in an n-gram translation model, which allows to capture contextual dependencies as with an n-gram language model, and in both cases, their estimates are smoothed.

1.4 Contribution

The goal of this project is to use language hierarchy to enrich a PBSMT system, while remaining in the string-to-string paradigm. We started from the observation that as adjuncts are syntactically optional, they can be deleted from a sentence without loss of grammaticality. Assuming that adjuncts in one language translate into adjuncts in the target language, one can delete adjunct pairs from a sentence pair without loss of grammaticality.

Consequently, one can delete adjunct pairs from the PBSMT baseline's training data to generate new training data, leading to new phrase pairs with which the baseline's phrase table can then be enriched. In this manner, one can access phrase pairs which are latently present in the data but invisible to PBSMT, and increase the size of the phrase table, while remaining in the phrase-based framework.

We tested this idea on the French-English language pair. As these languages have a similar syntax, one can in fact expect that adjuncts in one language often translate into adjuncts in the other language.

Our tests showed very little improvement on the PBSMT baseline. We found that, of the many phrase pairs the baseline is enriched with, only few are actually used in decoding. While our system could be improved in a number of ways, we believe nonetheless that factoring out adjuncts in training data can be interesting for SMT.

1.5 Outline

Chapter 2 provides background information on PBSMT-related topics. The IBM models are briefly presented, as well the GIZA++ word alignments used by PBSMT, and the PBSMT model itself.
Hierarchical models are also presented for comparison, as well as the BLEU metric for the evaluation of SMT systems, and finally, we introduce two grammar

formalisms, Lexicalized Tree-Adjoining Grammars (LTAG) and Head-Driven Phrase-Structure Grammar (HPSG).

In chapter 3, we investigate the feasability of adjunct-pair deletion. We define what we mean by adjuncts in this work, we set criteria to identify English adjuncts in a phrase-structure treebank, and we test in how far English adjuncts are aligned to French adjuncts.

Chapter 4 presents the steps required to implement our model, starting with an overview in section 4.2.
Section 4.3 describes the method followed to generate new training data for our system. From each sentence pair in the training data, we wanted in principle to generate as many sentence pairs as can be obtained by deleting anything between one and all of the adjunct pairs it contains. Generating sentence pairs amounts to creating new training data, which can then be processed by the Moses toolkit to extract new phrase pairs. Our main concern was then to maximize the amount of new phrase pairs while keeping the size of the generated data manageable.
Section 4.4 presents the method used to extract phrase pairs and to enrich the baseline. Training a phrase-based model from the new training data results in a phrase table containing new phrase pairs, with which we smooth the baseline.

Experiments on a number of enriched models are presented and discussed in chapter 5.

We conclude with chapter 6, where we propose ways to improve and extend our model.

Chapter 2

Background

2.1 The IBM models

The *IBM models* proposed by Brown et al. (1990) utilize an alignment variable a (Och and Ney, 2003) mapping word positions in a French sentence s, the source sentence, to word positions in an English sentence t, the target. Optimizing $P(s|t)$ consequently boils down to optimizing the sum on all possible alignments of $P(s, a|t)$:

$$P(s|t) = \sum_{a} P(s, a|t) \tag{2.1}$$

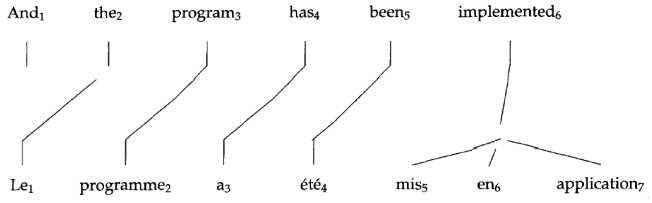

Figure 2.1: Alignment between a French and an English sentence, after Brown et al. (1993)

There are five IBM models, each built on the previous one with an increasingly complex alignment model. Model 1 uses a uniform distribution to model alignment probabilities, and Model 2 uses a zero-order distribution as alignment probabilities are conditioned on the word positions in *f*. Model 3 replaces the alignment parameter by *fertility* and *distortion*: *fertility* models the number of words generated by words in e, while *distortion* models word positions in f given aligned positions in e. Model 4 builds on Model 3 by using a first-order distortion model. Both models are deficient, as they notably allow for the generation of different words at the same position. Model 5 is built on Model 4 by correcting this deficiency.

2.2 Word alignments

Word alignments for PBSMT are established using the GIZA++ toolkit (Och and Ney, 2003), which applies the EM algorithm to the IBM Models 1, 3 and 4 and an HMM model. Word alignments are obtained for both language directions, French-to-English and English-to-French, before being merged into a single word alignment for both language directions.

There are three merging strategies: one can take the intersection or the union of both alignments, or an intermediary alignment. The latter is referred to as 'grow-diag-final' (Koehn et al., 2003) and is the strategy of choice for PBSMT. Starting from the intersection, 'grow-diag-final' adds alignment points from the union which are adjacent (horizontally, vertically or diagonally) to points in the current alignment, as shown in Figure 2.2. This process is repeated until no new points can be added.

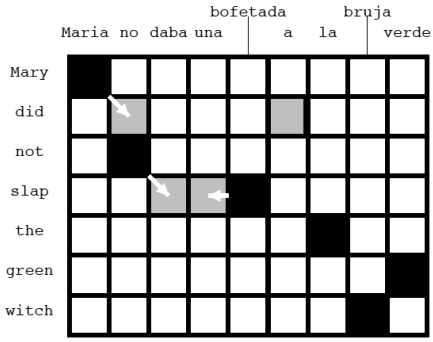

Figure 2.2: Illustration of the grow-diag merging strategy

The resulting alignments serve as basis to extract phrase pairs in PBSMT. A central requirement for phrase-pair extraction is that the phrase pairs must be *consistent* with the word alignments, which is to say that all words within the boundaries of a phrase are aligned to and only to words within the boundaries of the aligned phrase.

2.3 Phrase-Based SMT

The conditional probability $P(s|t)$ in a phrase-based model can be rewritten as the sum of the conditional probabilities of each segmentation $\sigma_{s,t}$ of s and t in a bag of phrase pairs with an ordering $O(s, t, \sigma_{s,t})$.

$$P(s|t) = \sum_{\sigma_{s,t}} P(\phi_s|\phi_t)P(O_s, O_t) \tag{2.2}$$

with (ϕ_s, ϕ_t) the bags of phrase pairs of s and t respectively, and (O_s, O_t) the source and target positions in $O(s, t, \sigma_{s,t})$.

For the purpose of finding the best translation t^*, it is deemed sufficient to find the best derivation $\sigma_{s,t}$ and thus:

$$t^* \simeq \arg\max_{t,\sigma_{s,t},O} P(t|s) \tag{2.3}$$

$$\simeq \arg\max_{t,\sigma_{s,t},O} P(s|t)P(t) \tag{2.4}$$

$$\simeq \arg\max_{t,\sigma_{s,t},O} P(\phi_s|\phi_t)P(O_s,O_t)P(t) \tag{2.5}$$

PBSMT models currently use a log-linear model to interpolate these terms, together with a length penalty $e^{|t|}$:

$$t^* = \arg\max_{t,\sigma_{s,t},O} P_t(\phi_s|\phi_t)^{\lambda_\phi} P_O(O_s,O_t)^{\lambda_O} P_{LM}(t)^{\lambda_{LM}} e^{|t|\lambda_w} \tag{2.6}$$

The translation model $P(\phi_s|\phi_t)$ itself is augmented with a reverse translation model $P(\phi_t|\phi_s)$ as well as lexical weights $P_w(\bar{s_i}|\bar{t_i})$ and reverse lexical weights $P_w(\bar{t_i}|\bar{s_i})$ for every phrase pair $\langle \bar{s_i}, \bar{t_i} \rangle$ of (ϕ_s, ϕ_t). Assuming that phrase-pairs are independent from one another, we can rewrite the translation probability of a sentence as the product of the translation probability of its phrase pairs:

$$P(\phi_s|\phi_t) = \prod_{\langle \bar{s_i}, \bar{t_i} \rangle} P(\bar{s_i}|\bar{t_i}) \tag{2.7}$$

and

$$P(\phi_t|\phi_s) = \prod_{\langle \bar{s_i}, \bar{t_i} \rangle} P(\bar{t_i}|\bar{s_i}) \tag{2.8}$$

The phrase translation probability distribution is estimated by heuristic counting:

$$P(\bar{s_i}|\bar{t_i}) = \frac{count(\bar{s_i}, \bar{t_i})}{\sum_{\bar{s}} count(\bar{s}, \bar{t_i})} \tag{2.9}$$

The reordering $O(s,t,\sigma_{s,t})$ is modeled by a relative distortion probability distribution. Assuming phrase pairs are independent, we have:

$$P(O_s,O_t) = \prod_{\langle \bar{s_i}, \bar{t_i} \rangle} d(a_i - b_{i-1}) \tag{2.10}$$

where a_i is the start position of s_i and b_{i-1} the end position of s_{i-1}.

2.4 Hierarchical models

Hierarchical models (Chiang, 2005) regard sentence pairs as derivations of a Synchronous Context-Free Grammars (SCFG). The grammar consists of rules of the type:

$$X \rightarrow \langle \alpha, \gamma, \sim \rangle \tag{2.11}$$

where α and γ are strings of terminals and / or non-terminals in the source and target language, and $\sim$ is a one-to-one correspondence between non-terminals in α and γ.

Besides, the grammar uses two 'glue' rules, which allow to model a preference for serial combinations of phrase pairs.

$$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle \tag{2.12}$$

$$S \rightarrow \langle X_1, X_1 \rangle \tag{2.13}$$

A rule's probability is modeled log-linearly, using similar features as PB-SMT: translation probabilities $P(\alpha|\gamma)$ and $P(\gamma|\alpha)$, lexical weights $P_w(\alpha|\gamma)$ and $P_w(\gamma|\alpha)$ and phrase penalty e.
The probability of a derivation D is estimated by interpolating the product of the probabilities of the rules r used in the derivation with the language model and the length penalty:

$$p(D) = \prod_{r \in D} p(r) \cdot p_{LM}(t)^{\lambda_{LM}} \cdot e^{-\lambda_{wp}|t|} \tag{2.14}$$

2.5 SMT evaluation with BLEU

The BLEU metric proposed by Papineni et al. (2002) is one of the most popular measures for evaluation of SMT systems. The metric consists essentially of weighted modified precision scores on n-grams. BLEU scores can be counted based on one or more reference translations, but we assume a single reference translation in this presentation.
The BLEU score is based on modified precision scores on n-grams. For unigrams for instance, one counts the words in the decoded output that are present in the reference, and divides this count by the total count of words in the output. The word count is clipped to ensure that words are not counted more often than they appear in the reference translation. In the case of Example 2.15, taken from Papineni et al. (2002), the count of the word 'the' in the output is clipped to its count in the reference, resulting in a modified unigram precision count of 2/7.

(2.15) output: the the the the the the the
reference: the cat is on the mat

The final score is computed by taking the geometric average of the modified n-gram precisions, weighted with positive weights w_n summing to 1, and a brevity penalty over the test corpus:

$$\log \mathrm{BLEU} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log p_n \tag{2.16}$$

where r is the length of the reference corpus[1], and c the length of the decoded output. In standard implementations, the weights w_n are distributed uniformly, and $N = 4$.

2.6 Lexicalised Tree-Adjoining Grammar

A Tree-Adjoining Grammar (TAG, Joshi et al. (1975); Joshi and Schabes (1997)) consists of a quintuple (Σ, NT, S, I, A) such that:
(i) Σ is a finite set if terminal symbols;
(ii) NT is a set of non-terminals;
(iii) S is a distinguished non-terminal;
(iv) I is a finite set of *initial trees*. The internal nodes of initial trees are non-terminals, while nodes on the frontier are either terminals, or non-terminals marked for substitution with a down arrow (↓);
(v) A is a finite set of *auxiliary trees* trees. Auxiliary trees are like initial trees except for one non-terminal, frontier node called the *foot node*. The foot node is annotated with an asterisk (∗) and has the same label as the tree's root node.

A Lexicalised Tree-Adjoining Grammar (LTAG) is a TAG for which all trees have at least one terminal at their frontier.

TAG's allow to derive trees from initial or auxiliary trees using two operations, *adjoining* and *substitution*, which are schematized in Figure 2.3.

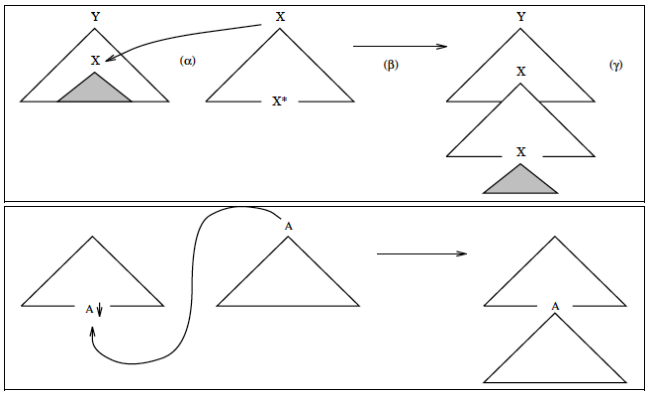

Figure 2.3: Tree-combining operations in TAG, after Joshi and Schabes (1997): adjoining (above) and substitution(under).

Both operations are exclusive, as adjoining can only take place on an internal node, and substitution on a node marked for substitution. Furthermore, only

[1]For several reference translations, r is the *effective reference length*, see Papineni et al. (2002).

auxiliary trees can be adjoined, while only initial trees, or trees derived from substitution can be substituted.

2.7 Head-Driven Phrase Structure Grammar

Head-Driven Phrase Structure Grammar (HPSG, Pollard and Sag (1994); Sag et al. (2003)) belongs to the group of Unification Grammars, and as such it uses a rich lexicon and a limited set of constraints.

Linguistic information is encoded in the lexicon using feature structures. A feature structure consists of a set of feature-value pairs, where values consist of one, or a list of, atomic entities or feature structures. For instance, the words *'a'* and *'cat'* in Figure 2.4 take a syntactic feature SYN and a semantic feature SEM; SYN takes as values the features HEAD and a valence feature VAL, which specifies what elements can be combined with the head of a phrase. VAL takes three values: the specifier feature SPR, the complements feature COMPS and the modifier feature MOD.

Linguistic information is organized using types in a multi-inheritance hierarchy. For example, both *words* and *phrases* are subtypes of the *synsem* type, and as such they inherit the SYN and a SEM features that are associated to this type.

Phrases are projected from lexical items using a limited amount of grammatical rules and principles, which specify constraints on feature values. For instance, the Head-Feature Principle ensures that the HEAD value of the head daughter is identified with that of the mother. This principle is respected by the phrase in Figure 2.4, as its head value is identified with that of the word *'cat'*.

2.8 Outlook

We introduced in this chapter a number of topics relevant to this work. We notably presented the PBSMT model, and the word alignments that form the basis for the extraction of phrase pairs in PBSMT. Besides, we introduced LTAG and an HPSG-based formalism.

In chapter 3, we use the formal grammars introduced here to illustrate what we mean by the term *'adjuncts'* in this work. We also investigate the feasibility of adjunct-pair deletion, which rests not only on the assumption that adjuncts in one language translate into adjuncts in the other language, but also that adjunct pairs are consistent with word alignments.

We will then moving on to the implementation and testing of an enriched model in the following chapters.

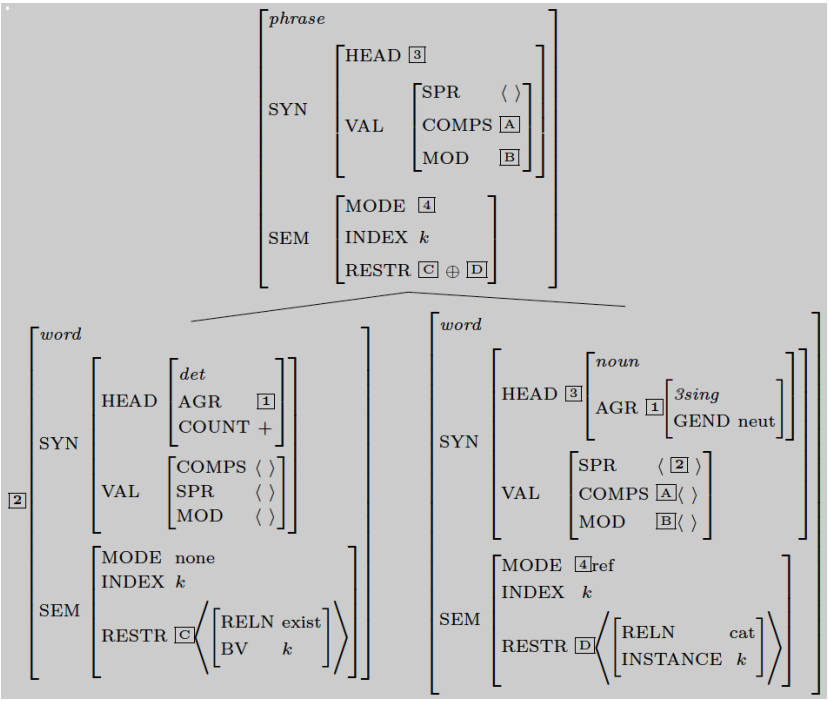

Figure 2.4: Example of HPSG feature specification: the phrase *'a cat'*

Chapter 3

Adjunct alignment between English and French

3.1 Introduction

As adjuncts are syntactically optional elements, deleting them from a grammatical sentence results in another grammatical sentence. For instance, deleting the adjuncts *'always'* and *'science-fiction'* from the English sentence on the left of Figure 3.1 results in a new grammatical sentence, *"John reads novels"*. Similarly, one can delete the adjuncts *'toujours'* and *'de science-fiction'* from the French sentence to obtain a new sentence, *"Jean lit des romans"*.

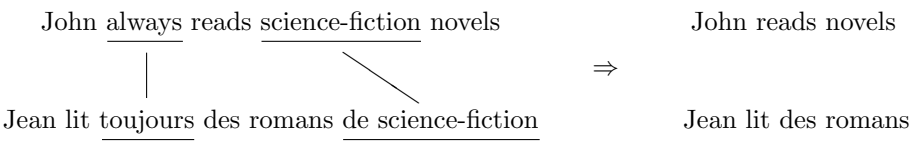

Figure 3.1: Sentence pair with paired adjuncts

The idea investigated here is whether one can extend this observation to deleting adjunct pairs from sentence pairs. If this is feasible, the consequence for a PBSMT model is that one can gain new sentence pairs from the baseline's training corpus and that new phrase pairs can be extracted from this new data to enrich the baseline model. In Figure 3.1 for instance, deleting the adjunct pairs $\langle$*always*, *toujours*$\rangle$ and $\langle$*science-fiction*, *de science-fiction*$\rangle$ would generate the new phrase pairs $\langle$*John reads*, *Jean lit*$\rangle$, $\langle$*reads novels*, *lit des romans*$\rangle$ and $\langle$*John reads novels*, *Jean lit des romans*$\rangle$.

The feasibility of generating data by adjunct-pair deletion rests on a double

assumption. The first is linguistic: We assume that adjuncts on one side of a bilingual corpus translate into adjuncts on the other side; The second is technical: We assume that adjunct pairs are consistent with word alignments.

We will assess these assumptions in this chapter. Before doing so however, we will deal with a point of terminology. In fact, both the terms *'adjunct'* and *'modifier'* can be associated with syntactically optional elements, and they are not used in the same manner by traditional and formal grammars. Section 3.2 sheds some light on the issue and explains what is meant by *'adjunct'* in this work.

We then deal with the issue of adjunct identification. We use a phrase-structure parser, the Charniak parser, to parse the English side of the training corpus. This allows us to identify English adjuncts, which are then paired to French adjuncts using word alignments. In section 3.3, we lay out categorical and distributional criteria to identify English adjuncts in a phrase-structure parse.

Finally, we assess the double assumption mentioned above. Section 3.4 illustrates the problem of cross-linguistic adjunct alignment with a few examples, and proposes a test on an aligned English and French treebank to assess in how far English adjuncts are aligned to French adjuncts.

3.2 Adjuncts, modifiers and complements

3.2.1 Traditional grammar

Traditionally, *adjuncts* are defined as optional constituents that are added to a syntactically complete clause, as opposed to *complements*, which are part of the argument structure of the verb. It is the latter that allows to tell adjuncts from complements: as Examples 3.1 and 3.3 show, the same phrase can serve as adjunct or as complement.

(3.1) Anna called *to make sure you were alright.*

(3.2) Anna called.

(3.3) Anna wanted *to make sure you were alright.*

(3.4) *Anna wanted.

The notion of syntactically optional constituents is also associated with that of *modifiers*. Prototypical modifiers are then adjectives, that modify nouns, and adverbs, that modify verbs.

3.2.2 Formal grammars

Formal grammars make no distinction between adjuncts and modifiers, as the only syntactical difference between them is that adjuncts modify clauses while

modifiers modify heads. Consequently, both terms, *adjuncts* and *modifiers*, are used somewhat interchangeably.

Adjuncts in LTAG

In LTAG, adjuncts are associated with an auxiliary tree. This is the case for *'passionately'* in Figure 3.2, which tree can be adjoined at the VP node of the tree for *'likes'*.

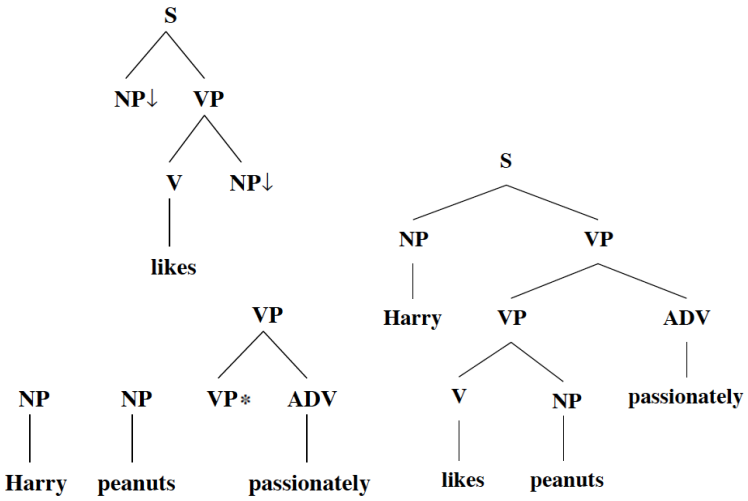

Figure 3.2: Adjuncts in LTAG

However, LTAG's also use auxiliary trees for words that do not qualify as adjuncts. This is notably the case for auxiliary verbs like *'has'* in Figure 3.3.

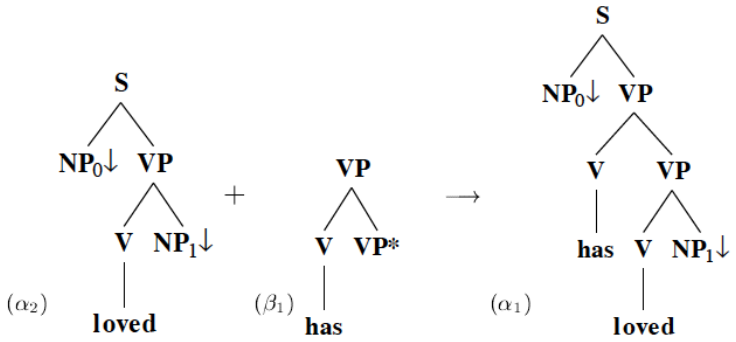

Figure 3.3: Non-adjunct with an auxiliary tree

Adjuncts in HPSG

HPSG describes adjuncts both using the modifier feature MOD. How modifiers combine with a head is specified by the Head-Modifier Rule, given in Figure 3.4.

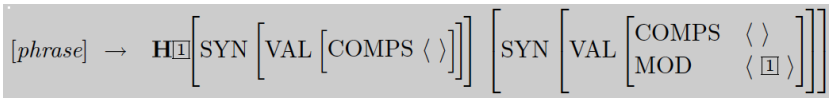

Figure 3.4: Head-Modifier Rule, after Sag et al. (2003)

The Head-Modifier Rule states that a phrase can consist of a head followed by a compatible modifier phrase. In other words, the value of the modifier's MOD feature is of the same category as the head. This allows to ensure for instance that adjectives can only modify nouns, and that adverbs can only modify verbs.

Adjuncts versus complements

"... complements refer to the essential participants in the situation that the sentence describes, whereas modifiers serve to further refine the description of that situation. This is not a precisely defined distinction, and there are problems with trying to make it into a formal criterion. Consequently, there are difficult borderline cases that syntacticians disagree about. Nevertheless, there is considerable agreement that the distinction between complements and modifiers is a real one that should be reflected in a formal theory of grammar." (Sag et al., 2003).

The distinction between complements and modifiers is reflected in HPSG by the different rules that apply to them. The Head-Complement Rule, shown in Figure 3.5, states that a phrase can consist of a lexical head followed by all its complements. The Head-Modifier Rule and the Head-Complement rule reflect that complements are selected by the head, while modifiers specify what kind of head they can modify.

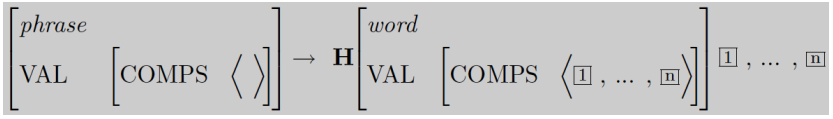

Figure 3.5: Head-Complement Rule, after Sag et al. (2003)

Whether a certain constituent must be analyzed as a modifier or a complement is complicated by the fact that complements can be optional too. In Example 3.5 for instance, the oblique object *'with a telescope'* is an optional complement of *'saw'*, while its direct object *'the star'* is a compulsory complement. Direct objects can be optional too, as Example 3.8 shows.

(3.5) I saw the star with a telescope.

(3.6) I saw the star.

(3.7) I ate an apple.

(3.8) I ate.

3.2.3 Adjuncts in this work

Syntactically, adjuncts and modifiers are opposed to complements in that they specify what head they can modify, while complements are selected by a head. The distinction between the two notions can be complicated as complements can be optional too.

We are interested in this work in syntactically optional elements in general. We will therefore use the term *'adjuncts'* to refer globally to adjuncts, modifiers and optional complements.

3.3 Adjunct identification

3.3.1 Identifying adjuncts with a phrase-structure parser

Section 3.2.1 showed that modifiers and complements cannot be told apart simply on the ground of their category. While the categories of adjectives and adverbs tend to be used as modifiers, other categories like prepositional phrases can be used both as modifiers and as complements. The distinction can be made using lexical information about the argument structure of the verbal head, but this information is absent from phrase-structure parses[1].

Nevertheless, phrase-structure trees allow to use two sorts of criteria: *categorial* criteria, as constituents are annotated with a category, and *distributional* criteria, as constituents can be related to a parent node, left and right siblings and children.

Besides, our aim is not to identify adjuncts with complete certainty. If compulsory complements should be identified as adjuncts, deleting them would lead to unusable phrase pairs, either because the French phrase would not be found in the test data or because the English phrase would be discarded by the language model at decoding.

Consequently, it is sufficient for our purpose to identify constituents that are *likely* to be optional. For this reason also, it is not necessary to define very precise identification criteria. One can assume for instance that all noun phrases are canonical, i.e. with a nominal head. This allows for example to regard adjectives and adjective phrases in noun phrases as adjuncts, without attempting to first establish the head of the noun phrase.

[1] While the distinction we make in this work is one between adjuncts as defined above and compulsory complements, this issue remains the same.

3.3.2 English adjunct categories

As a starting point to define adjunct-identification criteria, we analyze the first 50 parses of the treebank obtained with the Charniak parser from the English Europarl corpus to draw a list of adjunct categories and the associated restrictions on their distribution. The result is displayed in Table 3.1.

Table 3.1: Candidate English categories and selection criteria

category	parent	other restriction
ADJP	NP	
JJ	NP	
NNx	NP	NN/NNS right sister
VP	NP	
S	NP	
PP	$\neq$ PP	
SBAR	$\neq$ VP	
RB	$\neq$ ADVP	
ADVP		
PRN		
NP		adposed: left and right comma

The first items in Table 3.1 are adjuncts with a nominal head. The most common ones in our sample treebank are PP and JJ, followed by SBAR and VP, and finally RB, NNS and S. We add the prototypical ADJP to the list of noun modifiers, and we extend the category NNS to any nominal category (NN, NNS, NNP) or sequence starting and ending with a nominal tag, provided it is immediately followed by an NN or NNS tag. Constituents labeled PP, RB and SBAR nodes also occur in other contexts than noun phrases, the restriction on their parent node is there to exclude contexts where they are mostly compulsory complements. The remaining items are adjuncts in all (adposed NP and PRN[2]) or almost all cases (ADVP).

3.4 Adjunct alignment between English and French

3.4.1 Introduction

Our goal in this section is to verify a double assumption: First, that English adjuncts are paired by the word alignments to French phrases into pairs that are consistent with the same word alignments, and secondly, that the paired French phrases are adjuncts too.

[2] parenthetical

Regardless of the word alignments, English adjuncts may translate into French adjuncts of the same category as in Example 3.9, adjuncts of a different category as in Example 3.10, but they also may not be translated, or not as a separate constituent. In e.g. Example 3.11, *now* emphasizes the perfect aspect expressed by *has expressed*, which is done in French by the analytical structure *venir de + inf.*

(3.9) an objection [of that kind]$_{PP}$
une objection [de ce type]$_{PP}$

(3.10) [just a few months ago]$_{ADVP}$
[il y a quelques mois à peine]$_{S}$

(3.11) the document which the commission has presented [now]$_{RB}$
le document que la commission vient$_{V_{3sg}}$ de$_{I_{nf}M}$ présenter$_{V_{inf}}$

Such cases lead either to phrase pairs that are not consistent with the word alignment, or to phrase pairs in which the French phrase is not an adjunct. As in the case of adjunct identification, this is not critical: First, only phrase pairs that are consistent with the word alignment can be deleted, so other phrase pairs are not accounted with, and secondly, deleting phrase pairs with a non optional French phrase will presumably lead to phrase pairs with an unusable French phrase, which will not be found in the test data.

The question is thus in how far do English adjuncts form pairs that are consistent with the word alignments, and in how far can one expect the aligned French phrases to be adjuntcs. In the remainder of this section, the alignment of English adjuncts into French is assessed using an aligned English and French treebank, as presented in section 3.4.2. Test results are discussed in section 3.4.3.

3.4.2 Alignment-test set-up

The alignment of English adjuncts into French is tested using the French 'Arboratoire' treebank. This treebank roughly corresponds to the beginning of the French Europarl corpus, and can therefore be aligned to the parsed English side of the Europarl corpus.
Aligning both treebanks goes through the following steps:

- Arboratoire treebank extraction
- Alignment of the Arboratoire treebank with the Europarl corpus
- Alignment with the parsed English corpus
- Alignment with the GIZA++ word alignments

Arboratoire treebank extraction
The 'Arboratoire' treebank is collected using a TGrep search on the Arboratoire corpus[3]. The result is an HTML file containing 30421 sentences and parses,

[3] `http://corp.hum.sdu.dk/tgrepeye_fr.html`

automatically annotated with FRAG (**Fr**ench **A**nnotation **G**rammar)[4]. As Figure 3.6 shows for the first parse, each parse in the treebank is preceded by a line containing an identification number, a link to a graphical representation of the parse, the sentence and an additional tag.
We extract each sentence and its parse from the treebank and clean them. For the parses, this includes removing functional annotations.

```
#1 ID=ep-fr-00-01-17-ch1 Reprise de la session #A1/1
(STA+np (H+n Reprise) (DN+pp (H+prp de_la) (DP+n session)))
```

Figure 3.6: The first parsed sentence in the Arboratoire Treebank

Alignment of the Arboratoire treebank with the Europarl corpus
The Arboratoire sentences are matched with the Europarl sentences, and those that can be aligned are kept, with their parses, as well as their index (position) in the Europarl corpus. The result is a corpus containing 13624 French sentences and parses. The decrease in size from the original 30421 sentence pairs can be partially explained by the fact that the Arboratoire corpus are not filtered for length, whereas the Europarl corpus is filtered to contain sentences with 40 tokens at most.

Alignment with the parsed English corpus
The English side of the Europarl corpus is parsed using the Charniak parser. The parser fails to analyse some of the sentences, resulting in null parses.
The position indices of the French sentences and parses obtained at the previous step are used to select the English sentences from the Europarl corpus and their parses. Four of these parses being null parses, this results in a corpus containing 13620 French and English sentences and parses.

Alignment with the GIZA++ word alignments
The GIZA++ word alignments result from training the whole Europarl corpus (949408 sentence pairs) with Moses. The position indices collected above allow to directly access the relevant word alignments The result of this alignment is a parallel corpus containing 13620 French and English sentences, parses and unified word alignments.

This set-up makes it possible to identify English adjuncts in the English parses, locate their span in the English sentences, get the aligned spans in the French sentences, and locate the corresponding French strings in the French parses.

[4] `http://beta.visl.sdu.dk/visl/fr/info/taginfo_french.html`

3.4.3 Test results

For each of the English adjuncts specified in section 3.3.2, we gather the following measures, which are reported in Table 3.2:

frequency r_e : the frequency of the adjunct category relative to the number of sentences/parses.

lengths l_e **and** l_f : the lengths of the English and French adjunct, respectively

adjunct-pair measures. The adjunct pairs fall under five cases:

- **nc/A** : adjunct pairs that are not consistent with the word alignment
- $f_\varnothing$: The English adjunct is aligned to the empty string
- **c/P** : The French adjunct is consistent with the French parse, i.e. it forms a constituent or a sequence of constituents in the parse
- **nc/P** : The French adjunct is not consistent with the parse
- $f_?$: The French adjunct could not be located in the French parse

frequency r_{ap} : the frequency, relative to the number of sentences/parses, of the adjunct pairs with a French adjunct that is either consistent with the French parse or the empty string.

ratio x_{PC} : the ratio of parse-consistent and empty French adjuncts to the non-consistent adjuncts

Table 3.2: English-French adjunct alignment

	r_e	l_e	l_f	nc/A (%)	$f_\varnothing$ (%)	c/P (%)	nc/P (%)	$f_?$ (%)	r_{ap}	x_{PC}
JJ	0.98	1.0	1.0	18.4	3.5	74.7	3.0	0.4	0.76	26.2
RB	0.31	1.0	1.0	35.1	5.1	55.9	3.1	0.8	0.19	19.5
ADVP	0.63	1.4	1.4	24.0	6.4	63.9	4.9	0.9	0.44	14.3
SBAR	0.41	9.8	9.7	26.2	0.5	66.0	6.5	0.9	0.27	10.3
S	0.03	9.3	10.3	27.3	0.6	64.8	6.4	0.9	0.02	10.2
VP	0.09	6.4	6.9	30.4	0.9	61.5	6.5	0.8	0.06	9.6
PP	2.05	5.6	5.7	23.7	1.4	65.0	9.0	0.8	1.36	7.3
NNx	0.28	1.2	1.4	22.9	1.9	65.5	9.3	0.5	0.19	7.3
ADJP	0.10	3.0	2.9	26.9	1.0	62.3	9.1	0.7	0.06	7.0
PRN	0.04	5.6	3.5	19.3	5.9	59.9	10.6	4.3	0.02	6.2
NP	0.03	2.9	2.8	11.5	0.7	68.8	17.2	1.7	0.02	4.0

The English adjuncts in Table 3.2 are ordered by decreasing parse-consistency ratio.

The three highest ratio scores are obtained by JJ, RB and ADVP, which also yield the shortest constituents. The shorter the English constituent, the more likely it is to be aligned to the empty string or a single word. As these are always consistent with the parse, they automatically increase the parse-consistency score.

The three following categories, SBAR, S and VP, fare well given the average length of their constituents.

They are followed by ADJP, PP and NNx, which are subject to faulty parse attachments. Complex French adjective phrases can have their left side (head of the ADJP, or first adjective in a coordination) associated with the preceding noun, while the rest of the phrase is left out. This is the case in Example 3.12, where the French treebank gives the parse of Example 3.13, thereby failing to analyse the ADJP *équilibré et digne d'éloges*, and wrongly attaching the PP *M. Swoboda*.

(3.12) *le rapport équilibré et digne d'éloges de M. S.*
the report well-balanced and worthy of praise of Mr S.
Mr Swoboda's well-balanced and laudable report

(3.13) [le rapport équilibré]$_{NP}$ [et]$_{CONJ-C}$ [digne]$_{ADJ}$ [d'éloges de M. Swoboda]$_{PP}$

Similarly, most of the faulty NNx are aligned to a wrongly attached PP on the French side.

Finally, the categories PRN and NP suffer from the lack of punctuation handling by the French parser: the Arboratoire treebank contains virtually no punctuation marks, and this causes wrong attachments.

To complete the adjunct alignment tests, we investigate the parse derivations of the parse-consistent French translations of English adjuncts. For each English category, we look for the number of derivations of the French translation, the average number of top nodes per derivation, and the three most frequent derivations. The result of this test is shown in Table 3.3.[5]

The high number of derivations for each English adjunct can be explained by the number of tags used by the Arboratoire treebank (37, all of which but one appearing in the adjunct derivations), in combination with a flat structure.

Nevertheless, the most frequent derivations illustrate that English adjunct constituents tend to be aligned to French constituents of comparable nature. A notable exception is formed by the translation of English S constituents into French PP's. This is mainly due to the faulty parser analysis of French infinitive clauses, and in particular of the infinitive markers introducing these clauses. There are two infinitive markers in French: *à* (English *to*) and *de* (*from*), and as

[5]Most of the tags used by the Arboratoire treebank are self-explanatory, others are: ICL - infinitive clause, FCL - finite clause, PRP - preposition, V-PCP2 - past participle, PAR - coordination

Table 3.3: French derivations paired to English adjuncts

	total	nodes	Most frequent derivations		
JJ	86	1.0	ADJ - 69.0%	N - 10.1%	NUM - 3.0%
RB	70	1.0	ADV - 78.5%	ADJ - 3.5%	N - 2.1%
ADVP	251	1.1	ADV - 60.4%	PP - 5.8%	ADJ - 3.0%
SBAR	1233	2.7	FCL - 35.1%	PP - 6.2%	NP - 4.3%
S	94	2.7	PP - 45.7%	ICL - 6.3%	PRP ICL - 2.2%
VP	229	2.2	ICL - 18.6%	FCL - 10.2%	V-PCP2 PP- 9.8%
PP	1952	1.7	PP - 55.1%	PP PP - 7.0%	NP - 3.5%
NNx	64	1.1	N - 35.2%	ADJ - 26.6%	PP - 14.5%
ADJP	138	1.6	ADJP -29.7%	PAR - 9.4%	N ADJ - 6.6%
PRN	65	1.4	NUM - 29.9%	N-11.6%	NP - 10.5%
NP	52	1.7	NP - 28.6%	PROP - 24.6%	PROP NP - 11.2%

is the case for the English *to*, only context distinguishes them from prepositions. Similarly, some of the French phrases aligned to S constituents are analyzed as a preposition followed by an infinitive clause, PRP ICL, whereas it should be as an infinitive marker followed by an infinitive clause, INFM ICL.

It is clear in one case only that an aligned French derivation does not correspond to adjuncts. Table 3.3 reports that 35.2% of the NNx constituents are aligned to N constituents on the French side. While it is possible to qualify a French noun with another noun, this happens less often than in English, and one would expect most English noun qualifiers to be translated as adjectives or prepositional phrases in French. A closer look at these cases reveals that most of them concern nouns in a prepositional phrase, but that the word alignments only align the English noun to the French noun, instead of the whole phrase. Nevertheless, 41.1% of the NNx constituents are aligned to an ADJ or PP constituent, which are likely to be adjuncts.

3.5 Conclusion

The adjunct alignment tests have shown that all the selected English categories are mostly aligned to one or more French constituents. Looking at the most frequent derivations of the aligned French constituents moreover confirmed that English adjuncts tend to be aligned to French adjuncts.
Though not all English constituents are aligned equally well to French constituents, the lowest constituent-alignment scores concern French constituents

that are prone to wrong parse attachments, so that the low scores are imputable to the quality of the French parsing and not to a language feature.
All eleven English constituent categories identified in this section are subsequently regarded as adjuncts.

Chapter 4

Smoothing by factoring out adjuncts

4.1 Introduction

We showed in chapter 3 that English adjuncts tend to align to French adjuncts, and that the phrase pairs formed by English and French adjuncts tend to be consistent with word alignments.
In the present chapter, we describe how the English adjuncts identification criteria set out in section 3.3 are used to generate new training data, and how the model resulting from this new data is used to smooth the baseline.

Section 4.2 gives an overview of the steps involved in building an enriched model. Section 4.3 gives a detailed account of the training-data generation process, and section 4.4 describes how the phrase tables of the baseline and of the generated model are merged.

4.2 Overview

Building an enriched model goes through the following steps:

1. training the baseline
2. identifying adjunct pairs
3. deleting adjunct pairs
4. filtering the generated data
5. training a new model
6. enriching the baseline's phrase table with the new phrase table

The model-building process is illustrated in Figure 4.1.

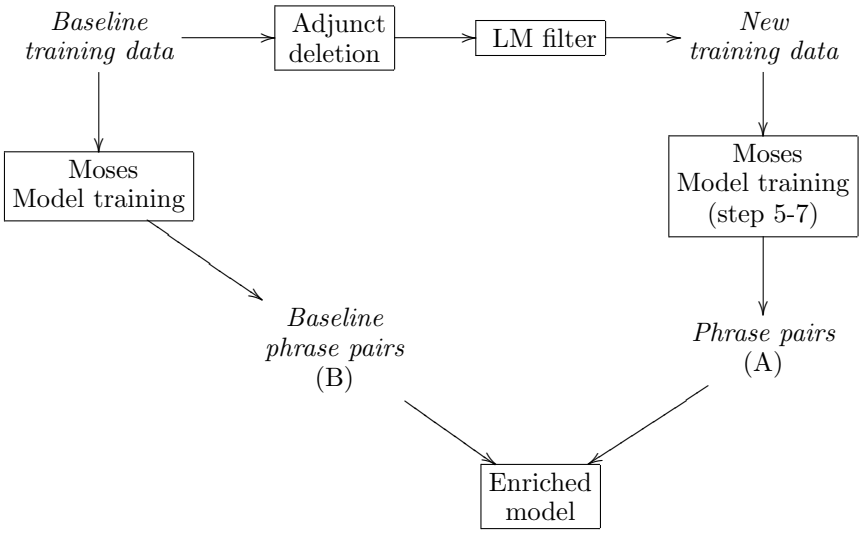

Figure 4.1: Building an enriched model

Training the baseline

The baseline is trained using the Moses toolkit (Koehn et al., June 2007). Besides, the training data and the unified word alignments resulting from the training are also used to generate new training data by adjunct deletion.

Identifying adjunct pairs

The Charniak parser is used to parse the English side of the baseline's training data. English adjuncts are then identified in the parsed data by means of the criteria given in section 3.3.2. Next, the baseline's unified word alignments allow to identify the French phrases paired to the English adjuncts.

Deleting adjunct pairs

Adjunct pairs that are consistent with the word alignments are deleted from the training-data sentence pairs, along with their alignment points in the word-alignment data. The result is a new set of aligned sentences, with their unified word alignments, which can be trained with the Moses toolkit.

Filtering the generated data

Given a sentence pair, we chose to generate a new sentence pair for each possible combination of adjunct pairs. A language-model filter is then used past the data-generation step to limit the amount of generated data.

Section 4.3 further describes the training-data generation step.

Training a new model

The new model is trained using part of Moses' training scheme. As a reminder, training a PBSMT model with Moses involves the following steps:

1. prepare the data
2. run GIZA++ to obtain word alignments
3. obtain unified word alignments
4. get lexical translation table
5. extract phrases
6. score phrases
7. reordering model
8. configuration file

The sentence pairs generated by adjunct-pair deletion, and the word alignments updated by deleting the word indices of these adjunct pairs, can be used to train a model from Step 5 of the training process: The generated sentence pairs take the place of the data at Step 1, and the updated word alignments that of the unified word alignments at Step 3. As no word pairs are added to the data in the process, the lexical translation table of the baseline can be reused, and the generated sentence pairs and word alignments can then be processed as training data by the Moses toolkit to extract and score phrases and build a reordering model.

Enriching the baseline

The training process results in a new phrase-pair table, which is interpolated with the baseline's phrase table, as explained in section 4.4.

4.3 Generating training data by adjunct-pair deletion

4.3.1 Introduction

Given identification criteria for adjunct pairs, one deletes these adjunct pairs from the baseline's training data to generate new training data. Provided that the adjunct pairs are consistent with the word alignments, one can then also delete the position indices of these pairs in the unified word alignments, thus generating new unified word alignments along with the generated sentence pairs.

As explained in section 4.4, generating sentence pairs with word alignments allows to use part of the Moses toolkit's training scheme for the generation of new phrase pairs.

The question that arises then concerns the number of sentence pairs to generate from each sentence pair. As a sentence may contain several adjuncts, one may choose to generate one sentence pair for each adjunct pair, or to delete a random combination of adjunct pairs to generate a single sentence pair, or again to generate all possible sentence pairs from all combinations of adjunct pairs. We chose for the latter, as it maximizes the number of phrase pairs that can later be extracted by the Moses toolkit. However, it also results in skewed phrase-pair counts on one hand, and on the exponential growth of generated sentence pairs with the amount of adjunct pairs per sentence pair on the other hand.

Phrase-pair counts are skewed as the number of generated sentences increases exponentially with the number of adjunct pairs. Given N adjunct pairs per sentence pair, there are 2^N combinations of adjunct pairs. Each adjunct pair is represented in half of these combinations, so the counts of the phrase pairs that can be extracted after the adjunct pair is deleted also grow exponentially with N. We assume that this issue is secondary as probability estimates are smoothed in the end.
The exponential growth of generated sentence pairs forms a practical challenge for processing with Moses. We limit it first by limiting the number of adjunct-pair combinations, and secondly by filtering the generated sentences according to their language-model score.

Section 4.3.2 describes how distance between adjunct pairs is used to cut down the number of adjunct-pair combinations. Section 4.3.3 describes the language-model filter used to further restrict the number of generated sentence pairs, and section 4.3.4 deals with qualitative measures to correct boundary effects of adjunct-pair deletion. Finally, section 4.3.5 presents the generated data.

4.3.2 Adjunct-pair filtering

By default, Moses extracts phrase pairs with a maximum length of 7 tokens. Consequently, if two consecutive adjunct pairs are distant enough, deleting them together will not yield more new phrase pairs than deleting them separately. See for an illustration Figure 4.2, where deleting the adjunct phrases $\bar{e}_1$ and $\bar{e}_2$ results in the new phrase $\bar{e}' = e_{i-1}e_{j+1}\dots e_{k-1}e_{l+1}$. Unless the distance $k - j$ between both adjuncts is inferior to the maximum phrase length, this new phrase will not be extracted by Moses, and it would then be redundant to generate it.

For ease of computation, one assimilates the distance between adjunct pairs to that between English adjuncts. One assumes thus if two English adjuncts are distant, their aligned French adjuncts are distant too.

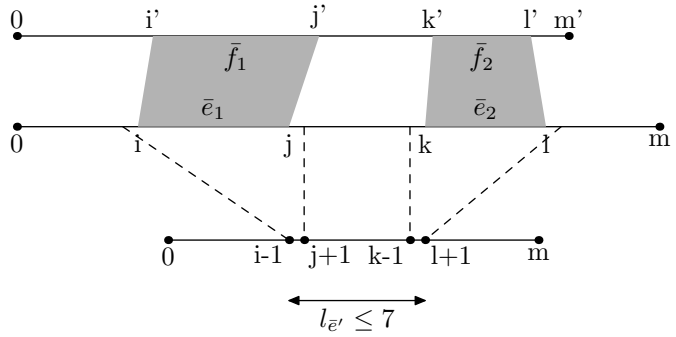

Figure 4.2: Adjunct-pair extraction.

The distance between consecutive phrases is used twice to filter adjunct-pair combinations (see Figure 4.3). First, the list of adjunct pairs is split into sublists, each sublist regrouping nearby phrases: if the distance between two consecutive English adjuncts is equal to or higher than the maximum phrase length, the left adjunct and all adjuncts before it are stored in a different sublist than the right adjunct and all adjuncts after it. Adjunct combinations are then computed for each sublist separately, thus decreasing the number of combinations from 2^N to:

$$1 + \sum_{i=1}^{L} (2^{k_i} - 1) \tag{4.1}$$

where L is the number of sublists and $\sum_i k_i = N$.
For instance, in Figure 4.3, the phrases $\bar{e}_1$ on one hand, $\bar{e}_2$, $\bar{e}_3$ and $\bar{e}_4$ on the other hand, are regrouped into distinct sublists, provided that $d_{12} \geq 7$. This reduces the number of possible combinations from 16 to 9.

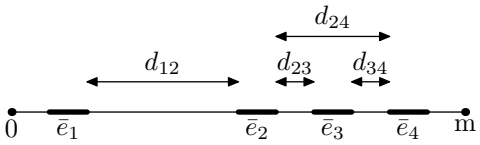

Figure 4.3: Using distance to filter adjunct-pair combinations. The extracted combinations are $(\varnothing)$, $(\bar{e}_1)$, $(\bar{e}_2)$, $(\bar{e}_3)$, $(\bar{e}_4)$, $(\bar{e}_2, \bar{e}_3)$, $(\bar{e}_3, \bar{e}_4)$ and $(\bar{e}_2, \bar{e}_3, \bar{e}_4)$.

Secondly, for each combination of adjuncts in a sublist, we check again if the distance between each consecutive adjunct is inferior to the fixed threshold.

Otherwise, the combination is filtered. This is the case with combination ($\bar{e}_2$, $\bar{e}_4$) in Figure 4.3, provided that $d_{24} \geq 7$.

4.3.3 Language-model filtering

The sentence pairs generated by adjunct-pair deletion are filtered by the language model trained on both the English and the French Europarl corpus. This allows to limit the amount of generated data while controlling its quality, as sentence pairs from which complements have been abusively removed can be excluded.

The French and English language models are first used to score each sentence pair in the training data, along with each sentence pair that is generated from it. The probability estimates are then corrected for length, and the sentence pairs that pass the filter are those with a score above that of the original sentence, modified by a threshold k.
Let $\langle e_0, f_0 \rangle$ a sentence pair in the Europarl corpus, and let $\langle e, f \rangle$ a sentence pair generated from $\langle e_0, f_0 \rangle$, we set the following criteria for $\langle e, f \rangle$ to pass the language-model filter:

$$P_{LM}(e)^{1/|e|} \geq k \cdot P_{LM}(e_0)^{1/|e_0|} \tag{4.2}$$

$$P_{LM}(f)^{1/|f|} \geq k \cdot P_{LM}(f_0)^{1/|f_0|} \tag{4.3}$$

4.3.4 Qualitative correction

While adjuncts are optional on the syntactic level, they can interact with surrounding words at other levels: For instance, as the form of the English indefinite article 'a/an' depends on the following word, deleting an adjunct following the article might lead to an illicit sequence of the type *'a + Vowel'*, or *'an + Consonant'*. Besides, as adjuncts can be marked by surrounding punctuation, deleting an adjunct might lead to an illicit sequence of punctuation marks, such as *',,'*.

Correcting the English indefinite article 'a/an'

If the indefinite article 'a/an' immediately precedes a deleted adjunct, one looks at the first letter of the following word. If it is a consonant, one ensures that the form 'a' of the article is used, and otherwise 'an'. In the case of the letter 'u', one states that 'u' is *vowel-like* if the second next letter is not a vowel-like 'u' or a vowel. For instance, in the word 'unusual', the third 'u' is vowel-like according to our definition, the second is not, and the first is.

Handling adjunct-marking punctuation

We regard as adjunct-marking punctuation any of the following tokens when they immediately precede or follow an extracted adjunct.

, . : ; ? ! - .. ...

Adjunct-marking punctuation is considered misplaced when: (1) it starts a sentence, or (2) it is preceded or followed by another punctuation mark.
Misplaced punctuation marks can be removed, possibly with their aligned counterpart, if one of the following conditions is met:

1. the punctuation mark is aligned to the empty string
2. the punctuation mark is aligned, together with some other token, to some token
3. the punctuation mark is aligned to a punctuation mark, and no other token is aligned to that punctuation mark

In the first two cases, the punctuation mark is removed, with no change in the aligned sentence. In the third case, the punctuation mark is removed together with its aligned counterpart. In other cases, notably when the punctuation mark is aligned to more than one token, the punctuation mark is regarded as unremovable.

Adjunct-marking punctuation is considered at two stages in the extraction process: First, adjunct pairs are discarded if their extraction would lead to unremovable, misplaced punctuation marks. Secondly, once a sentence pair has been generated, an attempt is made to remove the misplaced punctuation marks it may contain. If this fails, the sentence pair is discarded.

4.3.5 Generated data

Generating new training data starts with the identification of English adjuncts in the parsed Europarl corpus. One then searches for these adjuncts in the corpus proper and determines if they form adjunct pairs that are consistent with the word alignments. When an adjunct pair is not consistent with the word alignment, one tries to extend the word-span of the English adjunct to the token located immediately to its left and/or right. If the relevant left and/or right token is a punctuation mark, one checks if the extended adjunct forms an adjunct pair consistent with the word alignment. Next, one filters out adjunct pairs which would lead to uncorrectable misplaced punctuation marks (MPM) once deleted. At this stage, all the adjunct pairs that can be deleted are known. As shown in Table 4.1, this concerns 3.66M adjunct pairs, and 75% of the English adjuncts identified in the parsed corpus.

The rest of the Table gives the number of sentence pairs that can be generated, not including the original sentence pairs, at different filtering stages. One observes first that extracting all 3.66M adjunct pairs would result in the generation of 95.1M sentence pairs. This figure is reduced by splitting adjunct pairs based on distance, and by filtering out adjunct-pair combinations with overlapping or distant pairs. Combined, these filtering steps lead to 9.44M

sentence pairs. One then checks whether the sentence pairs would contain misplaced punctuation marks, in which case these are removed when possible. The resulting sentence pairs are then filtered by the language-model filter.

The language-model filter's threshold k is set to 0.7 to allow for the generation of an amount of sentence pairs, 4.12M, that is at the same time substantial while in the same order of size as the baseline's training data (949408 sentence pairs). This forms the new training data for most of the models tested in section 5.

Table 4.1: Adjunct pairs and sentence pairs obtained through the data generation process

adjunct pairs in TB	4.89M
adjunct pairs in corpus	4.85M
consistent pairs	3.72M
consistent pairs after extension	3.84M
pairs passed MPM filtering	3.66M
sentence pairs to generate	95.1M
after split	76.0M
after overlap check	17.1M
after distance check	9.44M
corrected punctuation	1.19M
generated sentence pairs	9.39M
passed LM filter, $k = 0.7$	4.12M

4.4 Model smoothing

4.4.1 Introduction

The training data generated by adjunct-pair deletion allows to train a new model with Moses, consisting of a new phrase-table and a new reordering table. The new model's phrase pairs consist of phrase pairs which are shared by the baseline and of new phrase pairs, with which the baseline is to be enriched. As the baseline's training data is not reproduced in the new training data, not all baseline phrase pairs are reproduced in the new model's tables, and consequently the new model's phrase pairs are not a superset of the baseline's phrase pairs.

The tables' contents are illustrated in Figure 4.4, where A is the set of phrase pairs in the tables resulting from adjunct deletion, and B the set of phrase pairs in the baseline tables.
Let T_B, T_A and T_C, the sets of target phrases that are constituent of phrase

pairs in B and not A, A and not B, and $A \cap B$ respectively.

$$T_B = \{\bar{t} | \exists \bar{s} : \langle \bar{s}, \bar{t} \rangle \in B \wedge \forall \bar{s} : \langle \bar{s}, \bar{t} \rangle \notin A\} \quad (4.4)$$
$$T_A = \{\bar{t} | \exists \bar{s} : \langle \bar{s}, \bar{t} \rangle \in A \wedge \forall \bar{s} : \langle \bar{s}, \bar{t} \rangle \notin B\} \quad (4.5)$$
$$T_C = \{\bar{t} | \exists \bar{s} : \langle \bar{s}, \bar{t} \rangle \in B \wedge \exists \bar{s'} : \langle \bar{s'}, \bar{t} \rangle \in A\} \quad (4.6)$$

One can partition B in B_B and B_C, the subsets of B whose phrase pairs have their target constituent phrase in T_B and T_C respectively:

$$B_B = \{\langle \bar{s}, \bar{t} \rangle \in B : \bar{t} \in T_B\} \quad (4.7)$$
$$B_C = \{\langle \bar{s}, \bar{t} \rangle \in B : \bar{t} \in T_C\} \quad (4.8)$$

Similarly, one can partition A into A_A in A_C:

$$A_A = \{\langle \bar{s}, \bar{t} \rangle \in A : \bar{t} \in T_A\} \quad (4.9)$$
$$A_C = \{\langle \bar{s}, \bar{t} \rangle \in A : \bar{t} \in T_C\} \quad (4.10)$$

The set of the phrase pairs that are common to the baseline and the new model is then a subset of the union set of B_C and A_C:

$$(B \cap A) \subset (B_C \cup A_C) \quad (4.11)$$

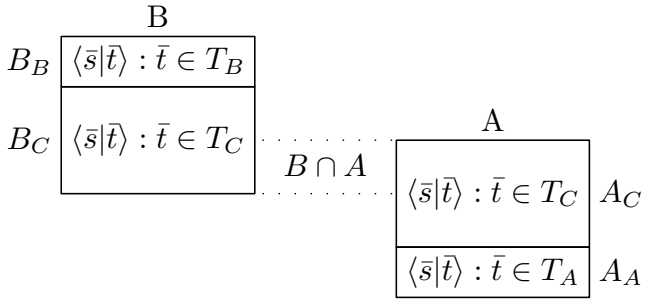

Figure 4.4: Partition of phrase-pair sets A and B according to the phrase pairs' target constituent phrases

4.4.2 Smoothing by interpolation

The baseline model is enriched with the new model by linear interpolation. Given a phrase pair $\langle \bar{s}, \bar{t} \rangle$ in $A \cup B$, the interpolated translation probability distribution $\phi_I(\bar{s}|\bar{t})$ is defined as:

$$\phi_I(\bar{s}|\bar{t}) = \lambda \phi_B(\bar{s}|\bar{t}) + (1 - \lambda)\phi_A(\bar{s}|\bar{t}) \quad (4.12)$$

where $\phi_B(\bar{s}|\bar{t})$ and $\phi_A(\bar{s}|\bar{t})$ are the translation probability distributions in the baseline and the new model, respectively.
The probability distributions are normalized to ensure model consistency. The normalization factor depends on which T_X set a target phrase $\bar{t}$ belongs to, as shown in Table 4.2.

Table 4.2: Interpolated estimates

phrase-pair set	T_X set	interpolated distribution	normalisation factor
B_B	T_B	$\lambda\phi_B(\bar{s}\|\bar{t}) + 0$	$1/\lambda$
$B_C - A$	T_C	$\lambda\phi_B(\bar{s}\|\bar{t}) + 0$	1
$B \cap A$	T_C	$\lambda\phi_B(\bar{s}\|\bar{t}) + (1-\lambda)\phi_A(\bar{s}\|\bar{t})$	1
$A_C - B$	T_C	$0 + (1-\lambda)\phi_A(\bar{s}\|\bar{t})$	1
A_A	T_A	$0 + (1-\lambda)\phi_A(\bar{s}\|\bar{t})$	$1/(1-\lambda)$

The phrase-pair tables contain both translation probability estimates conditioned on the target phrases, and inverse translation probability estimates conditioned on the source phrases. Interpolation is performed for both distributions.

4.4.3 Lambda setting

The models use either a constant interpolation parameter λ or one inspired from the Good-Turing estimate.
In this case, the probability mass allocated to the probability distributions $\phi_A(\bullet, \bar{t})$ increases with the relative frequency of single-occurrence phrase pairs with a constituent $\bar{t}$. The interpolation parameter $\lambda(\bar{t})$ is defined by:

$$1 - \lambda(\bar{t}) = \frac{n1}{n1 + N} \tag{4.13}$$

where $n1$ is the count of single-occurrence phrase pairs, and N the total count of phrase pairs with a constituent target phrase $\bar{t}$.

As most target constituent phrases in the baseline are associated with singleton phrase pairs, adding n_1 to the denominator of Equation 4.13 ensures that $1 - \lambda(\bar{t})$ never reaches 1. To prevent the opposite, in the eventuality that n_1 would be null, $1 - \lambda(\bar{t})$ is set to 10^{-4} by default.

4.4.4 Reordering model

Probabilities in the reordering model are estimated individually for each phrase pair, consequently one can directly enrich the reordering table with the new

model's table without smoothing. The enriched reordering model consists of the baseline model and of the new model's reordering probabilities for the phrase pairs in $A - B$.

4.5 Summary

The work presented in this chapter showed it is possible and fairly simple to enrich a PBSMT model through adjunct deletion.

An essential requirement for this to succeed was a reasonably high degree of correspondence between adjunct pairs and phrase pairs, as only adjunct pairs that are consistent with the word alignments can be deleted. We found that about 75% of the adjunct pairs identified in the baseline's training data were consistent with the word alignments, which is comparable with what we observed for the adjunct-alignment tests of section 3.4.

The adjunct-deletion step is the only subprocess that is not completely straight-forward. Qualitatively, while adjuncts are optional on the syntactic level, they interact with surrounding words at other levels: phonological in the case of the English indefinite article, typographical when they are marked by punctuation. These issues require modifying part of the data, which was done as efficiently as possible.

Quantitatively, one must filter adjunct-pair combinations to limit the amount of generated data. This is done both during the adjunct-deletion process, by filtering adjunct pairs and combinations of these, and after by filtering the generated sentence pairs. One can note first that more than half of the combinations of adjunct pairs are cancelled as they contain overlapping adjuncts. Other combinations were filtered using the distance between English adjunct pairs. While this is the simplest way to proceed, it is based on an assumption that relies heavily on word-order similarity between English and French, namely that if two English phrases are distant enough, i.e. separated by more than five words, their French counterparts will also be distant. While it may be acceptable for the French/English language pair, we find with hindsight that it would have been better to filter adjunct-pair combinations based on the distance between the French phrases and the English ones.

The second filtering mechanism consists in a language-model filter that selects the sentence pairs with the highest probability estimates with regard to the sentence pair they are generated from. The threshold of this filter makes it simple to adjust the amount of generated data.

Once a training data set is generated, a model can be trained using the Moses training scheme, and the resulting phrase table can be merged with the baseline table through interpolation. In section 5, we evaluate the performance of a number of models enriched by factoring out adjuncts.

Chapter 5

Experiments

5.1 Introduction

The previous chapter described how to build an enriched model by factoring out adjuncts. In the present chapter, one tests whether enriching a baseline system in this manner yields any significant improvement on the BLEU evaluation metric.

To obtain a broad picture of the enriched models' performance, tests are run on two in-domain test sets and two out-of-domain test sets, using different smoothing parameters and variations on the basic set-up, all of which are explained in section 5.2. Test results are presented in section 5.3, showing that the enriched models generally do not perform significantly better than the baseline. Section 5.4 provides an analysis of the test results. We conclude in section 5.5.

5.2 Experimental Set-up

5.2.1 Test set-ups

The basic set-up for the experiments uses the *2007 Workshop on Machine Translation* (WMT07) baseline's training data. The training data generated for the models is obtained with a language-model filter threshold of 0.7, as described in section 4.3.5. Models are built to decode from French to English. The tuning parameters of the baseline are re-used for the enriched models.
Variations on the basic set-up are: a small baseline training set, consisting of the 10000 first sentence pairs of the baseline's training data; a small generated training-data set, resulting from increasing the language-model-filter threshold from 0.7 to 1.0; reversing the language direction to decode from English to French; retuning the models.

5.2.2 Lambda settings

Given a training data set, three models M_1, M_2 and M_3 were built with a constant interpolation factor λ of 0.9, 0.99 and 0.999 respectively. A fourth model, M_{GT}, uses a Good-Turing inspired λ.

5.2.3 Test sets

The models are tested on four test sets:

- the in-domain WMT07 test set 'devtest2006': `devtest`
- an test set derived from the in-domain test set by adjunct deletion: `adjpoor`
- the out-of-domain WMT07 news-comment test set 'nc-test2007': `nc-test`
- a test set derived from the Hansards test set: `hansards`

`adjpoor` test set

The `adjpoor` test set is derived from `devtest` by adjunct-pair deletion, following the same procedure as for the training data: The new test set contains the sentence pairs that are generated by removing combinations of adjunct pairs in `devtest`[1], without replication of the original sentence pairs. The language-model threshold is set to 1.0 in order to enhance the quality of the generated sentence pairs while limiting their number.
The resulting test set consists of 8586 sentence pairs. While not all sentence pairs are equally grammatical, the test set allows to compare the performance of the generated models and of the baseline on adjunct-poor data.

`hansards` test set

The `hansards` test set consists of the 2000 first non-comment sentence pairs of the Hansards' House Debates Test Set[2]. We define as non-comment sentence pair one for which the English sentence ends with a period[3]. This is necessary as the Hansards test-set files begin with a list of contents, as illustrated in Figure 5.1. The selected sentence pairs are tokenized and lowercased as for the WMT07 test sets.

[1] Unlike the training set sentences, the `devtest` sentences are not limited in length. As a result, eight sentence pairs in the test set contain more than 20 adjunct pairs. In these cases, only single adjunct pairs and combinations of two adjunct pairs are removed.

[2] Starting from file "hansard.36.2.house.debates.073".

[3] ... and does not start with 'Bill'. The last measure excludes sentences of the type *"Bill C-463."*.

```
$ head hansard.36.2.house.debates.073.e
36 th Parliament, 2 nd Session
EDITED HANSARD * NUMBER 73
CONTENTS
Tuesday, March 28, 2000
ROUTINE PROCEEDINGS
GOVERNMENT RESPONSE TO PETITIONS
Mr. Derek Lee
ORDER IN COUNCIL APPOINTMENTS
Mr. Derek Lee
```

Figure 5.1: Begin of Hansards test-set file

5.3 Experiments

5.3.1 Introduction

The test results on the BLEU metric are reported below for each set-up. Significant changes in performance are reported when relevant. Significance is measured at a p-value $p = 0.05$ through approximate randomization[4].

5.3.2 Basic set-up

Table 5.1 reports the BLEU scores obtained by the baseline and the enriched models on the in-domain test set, `devtest`, the out-of-domain test sets, `nc-test` and `hansards`, and on the adjunct-poor test set `adjpoor`.

Table 5.1: BLEU scores with the basic set-up

	devtest	adjpoor	nc-test	hansards
baseline	32.47	33.18	24.41	22.24
M_1	32.50	33.34	24.35	22.15
M_2	32.53	33.33	24.38	22.18
M_3	32.52	33.35	24.38	22.16
M_{GT}	32.51	33.49	24.42	22.12

The enriched models perform slightly better than the baseline on the in-domain test set, but not significantly so. Performance is significantly better for all systems on `adjpoor`, with a maximum increase of 0.31 points for M_{GT}.

[4]FastMtEval: `http://www.computing.dcu.ie/~nstroppa/softs/fast_mt_eval.tgz`

All models underperform the baseline on the out-of-domain test sets, except for M_{GT} on `nc-test`.

5.3.3 Small generated-data set

Increasing the language-model-filter threshold makes the filter more stringent. As a result, less training data is generated for the enriched models. Increasing the threshold from 0.7 to 1.0 reduces from 4.12M to 1.03M sentence pairs the amount of training data that is generated from the baseline's training data . BLEU scores are reported in Table 5.2.

Table 5.2: BLEU scores with less generated training data

	`devtest`	`adjpoor`	`nc-test`	`hansards`
baseline	32.47	33.18	24.41	22.24
M_1	32.45	33.26	24.38	22.11
M_2	32.47	33.30	24.38	22.19
M_3	32.47	33.31	24.42	22.18
M_{GT}	32.50	33.30	24.44	22.09

The models' performance decreases with regard to the basic set-up on `devtest` and `adjpoor`, reflecting the smaller difference in training data between the enriched models and the baseline. The models' performance remains unchanged on the out-of-domain test sets. All models still perform significantly better than the baseline on `adjpoor`.

5.3.4 Small baseline training set

Table 5.3 reports the BLEU scores obtained from a baseline and enriched models trained on the first 10000 sentence pairs of the normal training set.

Table 5.3: BLEU scores with small baseline training set

	`devtest`	`adjpoor`	`nc-test`	`hansards`
baseline	25.94	26.24	15.77	16.56
M_1	25.95	26.16	15.70	16.63
M_2	25.96	26.19	15.72	16.55
M_3	25.97	26.18	15.74	16.56
M_{GT}	25.97	26.16	15.67	16.66

Again, the enriched models perform only slightly better than the baseline on the in-domain test set, showing no gain of relative performance when training on a small amount of data.
Moreover, all models now perform significantly worse on `adjpoor`. For the out-of-domain test sets, M_{GT} performs significantly worse than the baseline on `nc-test`.

5.3.5 English-to-French models

Table 5.4 reports the BLEU scores obtained on English-to-French models.

Table 5.4: BLEU scores on English-to-French models

	devtest	adjpoor	nc-test	hansards
baseline	31.32	32.56	25.01	21.94
M_1	31.26	32.51	24.98	21.98
M_2	31.30	32.55	24.99	21.96
M_3	31.31	32.60	24.95	21.96
M_{GT}	31.34	32.53	25.00	21.95

The models perform as well as the baseline on all test sets, the only significant improvement over the baseline being obtained by M_3 on `adjpoor`. Consequently, testing on adjunct-poor data does not yield the same kind of improvement as it does for the French-to-English language direction.
All scores are inferior to that obtained in the opposite language direction, except for `nc-test`, where all models including the baseline perform better than the French-to-English models.

5.3.6 Effect of retuning

All test results above were obtained by using the baseline tuning parameters for all models. We retuned the enriched models to find if this could benefit the enriched models. Table 5.5 reports the BLEU scores obtained on the `devtest` set after each model is individually retuned.

The best scores are obtained by the M_3 and M_{GT} models. We used the tuning parameters of M_{GT} to test all models again, in the basic set-up, on all test sets. BLEU scores are reported in Table 5.6.

Retuning improves the performance of all models, including the baseline, on the out-of-domain test sets and on `adjpoor`. In contrast, all models see their performance decrease slightly on `devtest`. The decrease being smallest for M_{GT}. Consequently, while retuning hardly affects the performance of M_{GT}, this model now performs significantly better than the baseline on the in-domain test set, with +0.19 BLEU points.

Table 5.5: In-domain BLEU scores.

	baseline parameters	retuned parameters
baseline	32.47	
M_1	32.50	32.42
M_2	32.53	32.46
M_3	32.52	32.50
M_{GT}	32.51	32.50

Table 5.6: BLEU scores obtained with the tuning parameters of M_{GT}

	`devtest`	`adjpoor`	`nc-test`	`hansards`
baseline	32.31	33.56	24.55	22.27
M_1	32.37	33.80	24.50	22.27
M_2	32.36	33.57	24.54	22.27
M_3	32.35	33.78	24.56	22.27
M_{GT}	32.50	33.79	24.46	22.27

5.3.7 Summary

If one compares the models' performance on the different test sets, one finds that, for the in-domain test set `devtest`, all models provide a marginal improvement over the baseline in the French-to-English test set-ups. Only through retuning does one model, M_{GT}, performs significantly better than the baseline, with +0.19 BLEU points.
On the adjunct-poor test set `adjpoor`, all models perform significantly better than the baseline when trained on the normal baseline's training set for the French-to-English language direction. They perform significantly worse when trained on the small baseline training set, and as well as the baseline when decoding from English to French.
Test results on the out-of-domain test sets `nc-test` and `hansards` show little difference in performance between the baseline and the enriched models.

Comparing the global performance of the enriched models shows that M_{GT} tends to perform the best, and M_1 the worse.

Comparing the models' performance on the different test sets show that all models perform better on `adjpoor` than on `devtest`. While all models perform worse on the out-of-domain test sets, results are better on `nc-test` than on

hansards, except when training on a small baseline training set.

5.4 Results analysis

5.4.1 Introduction

In this section, we try to elucidate why the enriched models, and more specifically M_{GT}, yield so little improvement on the baseline. We first attempt to explain the difference in model performance on the different test sets in section 5.4.2. We then move on to the question of the enriched models' performance with regard to the baseline. In section 5.4.3, we count how many generated phrase pairs are used in decoding, and in section 5.4.4 how many output sentences differ between the baseline and the enriched model. Section 5.4.5 discusses the effect of the langue-model filter on the selection of generated data. Section 5.4.6 conclude this analysis.

5.4.2 Comparing performance across test sets

Performance on a given test set depends on the ability of the model to match the input corpus on one hand, and the reference corpus on the other hand.

To evaluate how well enriched models match the input test data of the different test sets, we count the number of unknown words and the distribution of input phrase length for the M_{GT} model in the basic set-up. Figures are reported in Table 5.7.

Table 5.7: Matching the model with input test data

	devtest	adjpoor	nc-test	hansards
unknown words:				
tokens	173	680	669	524
types	168	119	538	315
input-corpus size, c	362k	1.85M	346k	281k
tokens / c (‰)	0.478	0.367	1.93	1.87
input-phrase-length distribution (%):				
$n = 1$	39.8	49.4	51.1	44.6
$n = 2$	27.2	24.4	27.6	27.5
$n = 3$	17.7	16.6	13.6	16.3
$n = 4+$	15.3	9.6	7.7	11.7

The ratio of unknown tokens to the input-corpus size may serve to some extent as an indication of the difficulty that a test set represents for a model: It is lowest for adjpoor and highest for the out-of-domain test sets. However,

BLEU scores are lower for `hansards` than for `nc-test`, while the tokens-to-size ratio is lower for `hansards`.

One can observe that the ratio of unknown tokens to types is lowest for `devtest` and highest for `adjpoor`. This test set was obtained by generating all possible new sentence pairs from `devtest`, with some sentence pairs leading to the generation of only one sentence pairs, and others leading to many sentence pairs. The high unknown tokens-to-types ratio can thus be explained by the replication of unknown words in sentence pairs generated from the same sentence pair.

The input-phrase-length distribution shows no correlation with system performance, as longer phrases are selected for `devtest` and `hansards`, while more unigrams are selected for `adjpoor` and `nc-test`.

Obviously, the coverage of input data by the model is not sufficient to explain its performance. To compare the test sets based on their reference output, we measured the language-model perplexity of each test-set reference. We also measure the 5-grams coverage, i.e. the fraction of 5-grams in the test data that can be found in the language-model, following Brants et al. (2007). Besides, we report the length of the output test corpora for the reference, the baseline and the enriched model M_{GT}. Figures are reported in Table 5.8.

Table 5.8: Model adequacy on the target side

	`devtest`	`adjpoor`	`nc-test`	`hansards`
Fitting of reference test data to the language model:				
perplexity	72.5	80.5	258	127
5-grams coverage	0.236	0.221	0.0676	0.130
Target-corpus length:				
reference, r	58.1k	299k	279k	237k
baseline, b	84.9k	445k	519k	421k
model,	84.6k	442k	517k	417k
r/b	0.684	0.672	0.537	0.564

The language-model perplexity and 5-gram coverage are the best for `devtest`. This is explained by the fact that the language model is trained on the target side of the training corpus, and thus on the same kind of data as `devtest`. Figures are somewhat worse for `adjpoor`, showing a lesser fit with the language model. Figures are worse for `nc-test` than for `hansards`, which stands again in contradiction with the BLEU scores.

Target-corpus lengths show that both the baseline and the enriched model M_{GT} produce long translations with regard to the reference corpus, on all test sets. As BLEU is a precision-based metric, this penalizes both the baseline and the enriched model. In this respect, the reference-to-baseline-target length

ratio, r/b, gives an idea of the global unigram precision that can be reached on a given test set. Again, we find that this figure is lower for `nc-test` than for `hansards`.

In conclusion, one can attribute the difference in performance between the in-domain test sets and the out-of-domain test sets both to the fact that the models and the baseline are less well adapted to match both the input and the output data of the out-of-domain test sets. The coverage of input words decreases, as well as the fitness of the language-model to the output data.

While the figures presented here seem to illustrate well the differences between the in-domain test sets and the out-of-domain test sets, they do not explain why performance, against all appearances, is better for `nc-test` than for `hansards`.

One can also note that the comparison of the in-domain test set `devtest` and `adjpoor` is complicated by the fact that the sentence pairs in `devtest` do not contribute uniformly to the generation of sentence pairs for `adjpoor`. Furthermore, while the generated data is filtered by the language-model, there is no actual means to control its quality. Consequently, `adjpoor` is only a rough attempt to provide a test set with adjunct-poor data. Nevertheless, test results show that it does allow the models to perform better than on the in-domain test set, and also to perform significantly better than the baseline, so this test set provides the enriched models with the same advantage than `devtest` does for the baseline.

5.4.3 Enriched-model contents

This section deals with the repartition of phrase pairs at different stages of the decoding process: in the training table, in the table filtered for decoding, and in decoding proper. As explained in section 4.4, phrase pairs can be unique to the baseline, shared by the baseline and the generated model, or they can be unique to the generated model. In this case, either their source constituent phrase is already used by baseline phrase pairs, and these new phrase pairs then provide new translation options to the decoder; or their source constituent phrase is new to the baseline, and the new phrase pairs then provide new input phrases.

Table 5.9 shows the repartition of phrase pairs in the enriched-model training tables, in the filtered tables prior to decoding, and in the phrase pairs used in decoding for the different test set-ups. The decoding tables and phrase pairs are those of the in-domain test set.

The figures for the training tables, filtered tables and decoding data show that while the proportion of new training phrase pairs is substantial across all set-ups, hardly any new phrase pairs are used in decoding.
When tables are filtered for decoding, the proportion of phrase pairs with new input phrases shrinks, showing that the enriched models bring proportionally

Table 5.9: Repartition of enriched-model phrase pairs

	table size	baseline only (%)	shared (%)	trans. options (%)	new input (%)
basic set-up:					
training	67.1M	10.4	52.0	7.2	30.4
filtered	4.84M	10.0	72.9	17.0	0.2
decoding	26.7k	1.4	98.4	0.0	0.2
less generated data:					
training	49.2M	49.5	35.7	3.4	11.5
filtered	4.27M	47.0	47.0	5.9	0.0+
decoding	26.8k	6.6	93.3	0.0	0.0+
small training set:					
training	819k	11.7	54.5	4.2	29.6
filtered	124k	11.1	76.0	10.5	2.4
decoding	41.7k	0.3	99.6	0.0	0.1
en-fr:					
training	68.7M	12.3	48.5	10.5	28.7
filtered	6.97M	14.8	57.7	27.4	0.1
decoding	29.6k	3.0	96.9	0.0	0.1

little input phrase pairs that match the test data[5]. Consequently, the new phrase pairs selected prior to decoding mostly bring new translation options. Nearly all phrase pairs used for decoding are shared by the baseline and the generated table, while none of the new phrase pairs with new translation options are used. It may be interesting to note that regardless of their origin, all the phrase pairs used at decoding have a target constituent that is used both by baseline and generated phrase pairs, i.e. belonging to the set T_C as defined in section 4.4. Consequently, even when a new phrase pair with a new input is used, it provides the system with an existing translation option.

If one compares the figures of the different test set-ups with the basic set-up, one finds that as less new training data is generated, less new phrase pairs are added to the model at all stages. Consequently, filtering the generated training data with a more stringent language-model filter does not generate proportionally more valuable phrase pairs.
When less baseline training data is used, the repartition of phrase pairs is comparable to that of the basic set-up. The enriched model provides 41.7k phrase pairs at decoding, against 26.7k for the basic set-up, showing that shorter phrase pairs are used. The average length of input and target phrase pairs is 1.39 and 1.52 for the small training set, and 2.18 and 2.38 for the basic set-up. That the repartition of phrase pairs is comparable for both set-ups shows that the enriched model suffers as much as the baseline from a decrease of training data.
In the English-to-French set-up, the repartition of phrase pairs is similar to that of the basic set-up. The main difference lies in the filtered tables, which contain more phrase pairs, and proportionally more phrase pairs with new translation options. Consequently, test-data English input phrases are coupled to more French translation options than in the opposite language direction. This phenomenon concerns both the baseline and the enriched models.

5.4.4 Effect of model enrichment on output translation

As the contribution of the enriched models in terms of phrase pairs is minimal, it is interesting to see how many output sentences actually differ from the baseline. Table 5.10 gives, for each test-set in the basic set-up and for M_{GT}, the number of sentences with a different translation and the associated BLEU scores. When translation output is identical, one distinguishes sentences with an identical or a different segmentation.

Table 5.10 shows that although the enriched model contributes few new phrase pairs, output translation is different for 30% to 43% sentences, indicating that the smoothed probability estimates lead to a different choice of output phrases. This is also reflected by the number of identical translations with a different segmentation (22% to 29%). Note that the difference seems very localized, as it tends to concern only a sequence of two phrases.

[5]Although only the in-domain test data is reported here, the figures are similar for all test sets.

Table 5.10: Effect of the models on output translation

	devtest	adjpoor	nc-test	hansards
diff. translation	645	3722	687	597
BLEU base	29.73	29.71	22.95	20.77
BLEU M_{GT}	29.81	30.31	22.99	20.44
diff. segmentation	488	2504	440	460
same translation	867	2360	880	943
BLEU	34.88	36.38	26.10	23.84

If one only considers different translations, the performance improvement of the enriched model M_{GT} over the baseline on `devtest` is slightly higher than overall, but still not significant. It does however indicate that smoothing helps to improve results.

5.4.5 Effect of the language-model filter

Consider the sentence pair of Example 5.1, where English adjuncts and their aligned French counterparts are emphasized.

(5.1) i would *[therefore]*$_{ADVP1}$ *[once more]*$_{ADVP2}$ ask you to ensure that we get a *[dutch]*$_{JJ}$ channel *[as well]*$_{ADVP3}$.
je vous demande *[donc]*$_{ADVP1}$ *[à nouveau]*$_{ADVP2}$ de faire *[le nécessaire]*$_{ADVP3}$ pour que nous puissions disposer d' une chaîne *[néerlandaise]*$_{JJ}$.

One can observe first that all English adjuncts are paired with French adjuncts, except for *'as well'*, which is not translated in the French sentence. Instead, the word alignments wrongly pair this phrase to *'le nécessaire'*, whereas *'faire le nécessaire'* is the translation equivalent of *'ensure'*.

Example 5.2 shows the English sentences generated from the sentence pair of Example 5.1 that pass the language-model filter with a threshold $k = 0.7$. Only two of these sentences also pass the filter when $k = 1.0$, the other one is emphasized.

(5.2) *i would once more ask you to ensure that we get a dutch channel as well .*
i would ask you to ensure that we get a dutch channel as well .
i would therefore ask you to ensure that we get a dutch channel as well .

In both settings, the language-model successfully discards the faulty phrase pair *'as well'/'le nécessaire'*, but it is otherwise to stringent in general, as it also discards sentence pairs resulting from the deletion of the pair *'dutch'/'néerlandaise'*.

In its more stringent setting, with $k = 1$, the filter also discards the sentence pair resulting from the deletion of *'therefore'/'donc'*.

In the present case, a language-model-filter threshold of 0.55 would allow to let pass all the sentence pairs resulting from the deletion of the three correct adjunct pairs, but a threshold of 0.50 would also let pass sentence pairs resulting from the deletion of the ungrammatical adjunct pair *'as well'/'le nécessaire'*.

Besides, the filter also lets pass ungrammatical sentences even with a high threshold, as Example 5.3 shows:

(5.3) my question relates to something that will come up and .

Although the sequence *'and .'* is ungrammatical, the language-model estimates the probability $p(.|and)$ to be higher than $p(dutch|a)$, which is itself higher than $p(channel|a)$.

The explanation for this is that the language-model is trained on the English side of the baseline's training data. Consequently the filter cannot discriminate very well between ungrammatical phrases and unfamiliar ones resulting from adjunct-pair deletion.

In conclusion, one cannot assume as we did that increasing the threshold of the language-model filter allows to operate a qualitative filtering of the generated data. While it may be true in general, it does not apply well to generated data that the language model is unfamiliar with. Consequently, the best setting for the language-model filter-threshold would be one low enough to let pass all the sentence pairs resulting from the deletion of correct adjunct pairs.

This issue is echoed at decoding, where the language model penalizes the new phrase pairs of the enriched model because it was not trained on them.

5.4.6 Summary

The analysis of test results performed here shows that performance on a given test set is indebted to how well the tested model and the language-model fit the data, which explains why the baseline and the models perform worse on the out-of-domain test sets.

The enriched models contribute very few new phrase pairs at decoding, and their output phrases are all already used by phrase pairs in the baseline.

Though the enriched models provide very few new input phrases, the smoothed probability estimates contribute to different output translations, so that they play the biggest part in improving performance.

Finally, the language-model filter is not a very good qualitative filter when it comes to new phrases, as it is trained on the baseline's training data. This also explains why the language model prefers to use baseline phrase pairs for decoding.

5.5 Conclusion

We started these experiments in the hope that the enriched model presented above would perform better than the baseline. We found that improvement on the in-domain test set was only significant after retuning, and still then very limited, with +0.19 BLEU points.

Experiments and their analysis point to a number of tracks concerning the potential and limitations of our model.

The language model plays a double role in limiting performance. First, the language-model filter identifies many new, grammatical phrase pairs with ungrammatical ones as they are not part of its training data. Secondly, and for the same reason, the language-model forces the decoder to prefer output phrase pairs that are already in the baseline.
A remedy to the first issue would be to lower the threshold of the language-model filter. For both issues, the language-model should ideally be trained on a larger and broader training set. As the language-model for the baseline is trained on the target side of the baseline's training data, the enriched models should also be trained on their own training-data's target side, which is to say that the baseline's language model should be smoothed with the language model trained on the data generated by adjunct-pair deletion.

The enriched models use information that is latent in the baseline training data. This explains that performance relative to the baseline does not increase as the baseline training data decreases.

Very few of the new input phrases provided by adjunct deletion are actually used. Their part concerns 0.2% of the phrase pairs used at decoding. However, 32% of the output translations differ from the baseline output for the in-domain test set. This suggests that the phrase-table smoothing is the main contributor to performance improvement.

Chapter 6

Conclusion

We presented in this work a novel manner to enrich a PBSMT system. The starting point of this work was that as adjuncts are optional constituents, they can be removed from grammatical sentences to derive new grammatical sentences, and that if adjuncts in a parallel corpus are paired by word alignments to adjuncts in the aligned corpus, then adjunct pairs can be removed from training data to generate more training data. This data serves to train a new PBSMT model, with new phrase pairs and probability estimates, which are then used to enrich the baseline model by smoothing.

We started out by defining what we meant by adjuncts. As our interest resided in the optional character of adjuncts, we have taken the term in a broad sense, covering adjuncts and modifiers, but also optional complements. We then laid out categorial and distributional criteria to identify English adjuncts in a PCFG treebank, and we used a parallel treebank to confirm that English adjuncts tend to be aligned to French adjuncts by word alignments.

We then presented a simple scheme to generate new training data through adjunct deletion, train it to obtain a new model and smooth the baseline with the new model. Driving constraints in this process were the limitation of generated data on one hand and simplicity on the other hand. For the baseline smoothing, we proposed to interpolate probability estimates using either a constant interpolation parameter or a Good-Turing estimate.

We extensively tested our enriched model, using several test sets, test setups, and smoothing parameters. Results show that enriched models work best when smoothed with a Good-Turing estimate, and when as much training data as possible is generated through adjunct deletion.

While the latter might seem obvious, we had assumed that the language-model filter would perform a qualitative selection of the generated sentence pairs. We found instead that the language-model, which is trained on the baseline's training data, considers many new phrases resulting from adjunct-pair deletion as improbable, and consequently discards grammatical phrases in a

fairly indiscriminate fashion.

The main result of our analysis was that very few of the new phrase pairs obtained by adjunct-pair deletion are actually used in decoding. In parallel, we observe an improvement, albeit very small, on the BLEU metric, which can be imputed to our smoothed estimates.

PBMST enrichment by adjunct-pair deletion could be improved in a number of ways.

The most important improvement would concern the language-model. As our language model was trained on the baseline's training data, it naturally favors baseline phrase pairs. Smoothing the baseline's language model with a language model trained on data generated by adjunct-pair deletion would therefore give a fairer chance to generated phrase pairs, both at the language-model filtering stage and in decoding.

In defining adjuncts we started from the traditional notion of adjuncts and modifiers, and we did not consider other constituents. In hindsight, it would be interesting to extend the class of deleted constituents to coordinated phrases. Factoring out some, but not all phrases of a coordination, results in a new coordinated phrase with less phrases or a single uncoordinated phrase. A first concern in this case would be to prevent the loss of subject-verb agreement.

Prior to generating data by adjunct-pair deletion, English-adjunct combinations are filtered depending on the distance between consecutive adjuncts. The assumption behind this relies heavily on word order, and while it may be acceptable for the French/English language pair, it would be better to filter adjunct-pair combinations based on the distance between both the English and the French adjuncts.

We believe that enriching PBSMT by factoring out adjuncts lends itself to a few developments.

It would be interesting to test our model on other language pairs. The French-English language pair has proved very fit to Machine Translation, and equally difficult to improve upon. While we verified that word alignments tend to align English adjuncts to French adjuncts, it would be interesting to investigate in how far this also applies to other language pairs, and whether a higher performance gain can be obtained then.

Finally, it would be interesting to factor out adjuncts in a hierarchical system. Using simply one adjunct-marking non-terminal beside the general terminal X could allow to add linguistic information of a very general nature. Besides, one could derive new rules by deleting adjunct markers.

Acknowledgments

I would like to thank Khalil Sima'an for his excellent supervision all along the arduous gestation of this thesis. Khalil was supportive at all times, and gave me critical and stimulating guidance on contents and form while always inciting me to explore my own ideas. This work would not have been half as personal without his input, and certainly not half as good.
I would also like to thank Markos Mylonakis for his help on Moses and his advice on the thesis and on practical matters, and Rens Bod for his stimulating comments at the defense.
My final thanks go to Ari de Jong, for his unaltered support and encouragements, to Ulle Endriss for looking past all the times I told him I was nearly done, and to everybody who reminded me in one way or another that I could actually finish this thesis.

Bibliography

T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large Language Models in Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 858–867, 2007.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

X. Carreras and M. Collins. Non-projective parsing for Statistical Machine Translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 200–209, 2009.

D. Chiang. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, 2005.

D. Chiang. Learning to Translate with Source and Target Syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

G. Foster, R. Kuhn, and H. Johnson. Phrasetable Smoothing for Statistical Machine Translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006.

H. Hassan, K. Sima'an, and A. Way. Syntactically Lexicalized Phrase-Based SMT. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7), September 2008.

L. Huang, K. Knight, and A. Joshi. Statistical Syntax-Directed Translation with Extended Domain of Locality. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 66–73, 2006.

A. K. Joshi and Y. Schabes. Tree-Adjoining Grammars. In G. Rosenberg and A. Salomaa, editors, *Handbook of Formal Languages*. Springer-Verlag, New York, NY, 1997.

A. K. Joshi, L. S. Levy, and M. Takahashi. Tree Adjunct Grammars. *Journal of Computer and System Sciences*, 10(1), 1975.

P. Koehn, F. J. Och, and D. Marcu. Statistical Phrase-Based Translation. In *Proceedings of HLT/NAACL*, pages 127–133, 2003.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, F. Marcello, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual meeting of the ACL, demonstration session*, Prague, Czech Republic, June 2007.

R. Kuhn, G. Foster, S. Larkin, and N. Ueffing. PORTAGE Phrase-Based System for Chinese-to-English Translation. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 75–80, 2006.

D. Marcu, W. Wang, A. Echihabi, and K. Knight. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, 2006.

M. Mylonakis and K. Sima'an. Learning Hierarchical Translation Structure with Linguistic Annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 642–652, 2011.

M. Nagao. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In *Proceedings of the international NATO symposium on Artificial and human intelligence*, pages 173–180. Elsevier North-Holland, Inc., 1984.

F. J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29:19–51, 2003.

F. J. Och and H. Ney. Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics*, pages 440–447, 2000.

K. Papineni, S. Roukos, T. Ward, and Z. Wei-Jing. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, 2002.

C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, 1994.

C. Quirk and A. Menezes. Dependency Treelet Translation: The convergence of statistical and example-based machine translation? *Machine Translation*, 20:43–65, March 2006a.

C. Quirk and A. Menezes. Do we need phrases? Challenging the conventional wisdom in Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 9–16, 2006b.

I. A. Sag, T. Wasow, and E. M. Bender. *Syntactic Theory. A Formal Introduction.* CSLI, 2003.

H. Schwenk, M. R. Costa-Jussà, and J. A. Fonollosa. Smooth Bilingual N-gram Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 430–438, 2007.

A. Zollmann and A. Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of NAACL 2006 - Workshop on statistical machine translation*, pages 138–141, 2006.