Resourcing Machine Translation with Parallel Treebanks

John Tinsley

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing

Supervisor: Prof. Andy Way

December 2009

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.:

Date:

Contents

Abstract

The benefits of syntax-based approaches to data-driven machine translation (MT) are clear: given the right model, a combination of hierarchical structure, constituent labels and morphological information can be exploited to produce more fluent, grammatical translation output. This has been demonstrated by the recent shift in research focus towards such linguistically motivated approaches. However, one issue facing developers of such models that is not encountered in the development of state-of-the-art string-based statistical MT (SMT) systems is the lack of available syntactically annotated training data for many languages.

In this thesis, we propose a solution to the problem of limited resources for syntax-based MT by introducing a novel sub-sentential alignment algorithm for the induction of translational equivalence links between pairs of phrase structure trees. This algorithm, which operates on a language pair-independent basis, allows for the automatic generation of large-scale parallel treebanks which are useful not only for machine translation, but also across a variety of natural language processing tasks. We demonstrate the viability of our automatically generated parallel treebanks by means of a thorough evaluation process during which they are compared to a manually annotated gold standard parallel treebank both intrinsically and in an MT task.

Following this, we hypothesise that these parallel treebanks are not only useful in syntax-based MT, but also have the potential to be exploited in other paradigms of MT. To this end, we carry out a large number of experiments across a variety of data sets and language pairs, in which we exploit the information encoded within the parallel treebanks in various components of phrase-based statistical MT systems. We demonstrate that improvements in translation accuracy can be achieved by enhancing SMT phrase tables with linguistically motivated phrase pairs extracted from a parallel treebank, while showing that a number of other features in SMT can also be supplemented with varying degrees of effectiveness. Finally, we examine ways in which synchronous grammars extracted from parallel treebanks can improve the quality of translation output, focussing on real translation examples from a syntax-based MT system.

Acknowledgements

First and foremost, I'd like to extend a huge thanks to my supervisor Andy Way. He was the person who sparked my interest in MT initially and has been a constant source of pragmatic advice and encouragement, not only over the course of this thesis, but since I started in DCU back in 2002. He's almost the ideal supervisor – if only he supported Liverpool.

Secondly, I'd like to thank Mary Hearne, without whom the initial hurdles encountered by me as a fresh-faced PhD student would have been a lot more difficult to overcome. She was a fountain of information not only on technical aspects of my work, but also in terms of practical advice. Thanks also to Ventsi for his collaboration and company as we began our theses together. I'd like to think we made things a little easier for one another.

I wish to acknowledge the support of the various bodies who funded my research, notably Science Foundation Ireland and Microsoft Ireland, and the Irish Centre for High End Computing for the use of their resources.

Big thanks go to the members of the NCLT/CNGL, both past and present — including Ankit, Declan, Grzegorz (for his help with the Spanish parser especially), Harold, Joachim, Josef, Karolina, Nicolas, Patrik, Sara, Sergio, Sylwia and Yvette to name a few/lot — for their questions, discussions and general interest regarding my work. And thanks to Augusto for weaning me off Windows before it was too late.

At the beginning of 2009, I spent a number of weeks at Carnegie Mellon University in Pittsburgh. This was a particularly enjoyable and fruitful period and for that I'd like to thank Alon Lavie. Thanks also to Greg, Vamshi and Jon for their discussions and help while I was there (and when I came back home), particularly in terms of getting the Stat-XFER system up and running so that I could write Chapter 5! and to Stephan for hosting me in Pittsburgh.

A special thanks goes my friends, particularly Rose, for providing me with sufficient distraction from my work over the last 3+ years so that it never consumed me (too much). Finally, a wholehearted thank you to my family for their support, in all senses of the word, over the course of the last 8 years I have spent in university. Don't worry, I'll get a real job soon.

List of Figures

List of Tables

Chapter 1

Introduction

Data-driven approaches have long succeeded rule-based methods as the primary research direction when addressing the problem of machine translation (MT). Such approaches learn models of translation from large corpora of parallel data. Statistical MT (SMT) has been the dominant data-driven paradigm for a number of years and this can be attributed in large part to the availability of free open-source software, e.g. GIZA++ (Och and Ney, 2003), Moses (Koehn et al., 2007), SRILM (Stolcke, 2002), and parallel corpora, e.g. Europarl (Koehn, 2005), for training. Another aspect which has contributed to the popularity of SMT is the fact that, in terms of parallel training corpora, unannotated 'plain text' data is all that is required and in today's multicultural climate, such bilingual data is abundant, at least for major languages.

More recently, there has been widespread discussion as to whether pure statistical approaches to MT have hit a ceiling with regards to the quality of translations they can achieve. As a consequence of this, there has been an obvious trend towards the development of more linguistically-aware models (predominantly syntax-based) of translation. A prerequisite of such models is parallel data with some level of a priori analysis/annotation. While monolingual treebanks are widely available thanks to large-scale annotation projects (e.g. Marcus et al. (1994); Civit and Martí (2004); Telljohann et al. (2004) amongst others), bilingual parallel corpora with syntactic annotation on both sides — so-called parallel treebanks — of any size are few and

far between. This can mainly be attributed to the huge effort required to produce such a resource. Because of this, there has been a lot of research carried out on tree-to-string MT models,[1] e.g. Yamada and Knight (2001), while the development of tree-to-tree based models, despite their potential, has suffered.

In this thesis, we seek to address the dearth of resources for syntax-based MT by exploiting existing monolingual technologies as well as novel techniques to develop a methodology for the automatic generation of large-scale parallel treebanks. This gives rise to our first research question.

> **RQ1:** Can we develop a method to facilitate the automatic generation of large-scale high-quality parallel treebanks for use in MT?

To this end, we design a novel algorithm for inducing sub-sentential translational equivalence links between pairs of parallel trees produced using monolingual constituent parsers. In order to address concerns regarding the propagation of errors given the multiple automated processes involved in the generation of parallel treebanks, we rigorously assess their viability by employing them as training data in series of tree-to-tree MT systems. Furthermore, we perform a detailed analysis of the treebanks in two ways: intrinsically by comparing the automatically generated parallel treebanks to a manually crafted version of the same, and by carrying out a manual assessment of the induced sub-tree alignments.

Following on from this, we hypothesise that, despite their obvious applicability for syntax-based MT, parallel treebanks also have the potential to be exploited in statistical paradigms of translation. This leads is to our next two research questions.

> **RQ2:** Can syntactically motivated phrase pairs extracted from a parallel treebank be exploited to improve phrase-based SMT?

> **RQ3:** What other features of the phrase-based model can be enhanced by exploiting the information encoded in parallel treebanks?

[1] Tree-to-string models almost always include English on the 'tree' side as it is heavily resourced in terms of annotated data and annotation tools.

Taking advantage of the many open-source tools available for SMT, we design an exhaustive set of experiments in which we supplement phrase-based translation models with parallel treebank-induced phrase pairs and carry out further tests aimed at discovering various ways in which parallel treebanks can be used in SMT, for example, using parallel treebank word alignments to seed the SMT phrase extraction process. Experiments are performed across a range of data sets and language pairs in order to ascertain the conditions under which parallel treebanks can be optimally exploited in SMT.

Returning to our original problem, the lack of resources for syntax-based MT, we present an additional research question.

> **RQ4:** To what extent are our automatically generated parallel treebanks useful in syntax-based MT?

In addressing this question, we analyse the performance of a syntax-based MT system when using a parallel treebank as training material by performing both an automatic evaluation of translation quality plus a detailed manual assessment of observed improvements in translation output.

Thesis structure The remainder of this thesis is structured as follows. In Chapter 2, we present background information on relevant topics related to this work. In Chapter 3, we describe a novel algorithm for the induction of sub-sentential alignments between parallel trees. Chapters 4 and 5 detail a series of experiments carried out investigating the exploitability of automatically generated parallel treebanks in both statistical MT and syntax-based MT respectively. Finally, in Chapter 6, we conclude and present some avenues for future work. A more detailed description of the work is given in the following.

Chapter 2 Parallel treebanks are a relatively new concept in the area of natural language processing (NLP). In this chapter, we describe the characteristics of

a parallel treebank and the challenges faced when building one, particularly the issue of sub-sentential alignment and how this differs from 'regular' word alignment. Following this, we give an overview of the phrase-based SMT (PB-SMT) paradigm, providing additional details on those aspects especially pertinent to the experiments presented in later chapters, i.e. the phrase extraction process and the translation and log-linear models. We then present the concept of syntax-based MT and summarise a number of techniques for incorporating linguistic information into the translation process, e.g. tree-to-string and tree-to-tree models. Specific details are given for two systems, the data-oriented translation (DOT) model (Poutsma, 2000; Hearne and Way, 2003; Hearne, 2005) and the CMU statistical transfer (Stat-XFER) framework (Lavie, 2008; Hanneman et al., 2009) as we employ these systems directly throughout this thesis. Finally, we describe the automatic metrics used to evaluate the translation quality of our various MT system configurations in this work.

Chapter 3 In this chapter, we present the novel sub-tree alignment algorithm we have developed in terms of design and performance (Tinsley et al., 2007b; Zhechev, 2009). Firstly, we describe the conditions to which we endeavour to adhere over the course of the development, namely language pair- and task-independence. Following this, we present the notion of a well-formed alignment and our baseline algorithm. A number of extensions and configurations are introduced to resolve various issues that arose during development and a description of the scoring functions used to seed the greedy search algorithm is provided. We then go on to intrinsically evaluate the performance of our algorithm by comparing the resulting alignments to a set of manually inserted alignments, and we carry out an extrinsic evaluation using the automatically generated parallel treebanks to train DOT systems. Finally, we manually assess the performance of the sub-tree alignment algorithm by examining its ability to capture a number of translational divergences present in the data (Hearne et al., 2007).

Chapter 4 We hypothesise that automatically generated parallel treebanks may be of use beyond syntax-based approaches to MT. To this end, we design a number of experiments to investigate ways in which treebanks can be exploited in phrase-based SMT. In this chapter, we present initial pilot experiments in which syntactically motivated phrase pairs extracted from parallel treebanks are used to supplement the translation model of a PB-SMT system (Tinsley et al., 2007a). Following the success of these experiments, we build a parallel treebank almost two orders of magnitude larger than that of Tinsley et al. (2007a) — to our knowledge, the largest parallel treebank exploited for MT training at the time — and replicate the pilot experiments, as well as investigating a number of innovative techniques for combining our syntax-based phrase pairs with non-syntactic SMT phrases pairs in the PB-SMT model (Tinsley et al., 2009). Additionally, we examine further ways in which parallel treebanks can be exploited in the PB-SMT pipeline. We use the treebank-based word alignments to seed the phrase-extraction process and to inform the lexical weighting feature in the log-linear model. In the remainder of the chapter, we investigate the effect the size of the training data set has on the influence of parallel treebank phrase pairs in the PB-SMT model (Tinsley and Way, 2009) and describe our combination techniques as applied in the shared translation task at the International Workshop on Spoken Language Technologies (IWSLT-08) (Ma et al., 2008). Finally, we present initial experiments designed to investigate the feasibility of using our sub-tree alignment algorithm to align dependency structures for SMT phrase extraction (Hearne et al., 2008).

Chapter 5 In order to fully exploit the information encoded in parallel treebanks, we need to employ them in an appropriate syntax-based MT system. Accordingly, we build a parallel treebank — almost twice as large as that of Tinsley et al. (2009) — and evaluate its performance when used to train a Stat-XFER system. We observe improvements in translation quality, based on both automatic and manual analysis, when using a small-scale grammar extracted from our parallel treebanks.

We suggest there is significant research required to find out how best to extract efficient grammars for syntax-based MT. Finally, for completeness we replicate the phrase combination experiments of Chapter 4 with this larger parallel treebank. We confirm our intuition that the influence of syntax-based phrases pairs would diminish as the training set size grows and discuss the implications of this going forward. However, we also address our findings that the parsing formalism has a telling effect on the set of extractable phrase pairs.

Chapter 6 Finally, we conclude and present a number of opportunities for future work based on open research questions that have arisen throughout the course of this thesis.

The work presented in Chapter 3 of this thesis (Tinsley et al., 2007b; Hearne et al., 2007) was carried out as part of a joint project with Ventsislav Zhechev at the National Centre for Language Technology at Dublin City University (DCU). Both Ventsislav and the author contributed in equal part to the design, development and evaluation of the alignment algorithm as described here. Further extensions to the algorithm were made by Ventsislav in the pursuit of his PhD thesis (Zhechev and Way, 2008; Zhechev, 2009). Similarly, the experiments presented in Section 4.5 (Hearne et al., 2008) were carried out in collaboration with Mary Hearne and Sylwia Ozdowska at DCU. The author's principal contributions to this portion of work were the design and execution of the MT experiments along with analysis of the resulting translation performance. The conversion of dependency parses to constituency structures was carried out by the collaborators. All other research presented in this dissertation was the author's own work.

Chapter 2

Background and the Current State-of-the-Art

In this chapter, we describe the state-of-the-art and related research within the areas explored by this thesis, paying particular attention to those aspects directly related to our novel approaches. More specifically, in section 2.1, we discuss parallel treebanks and the motivation behind our need to design a sub-sentential alignment algorithm. In section 2.2, we present the various components in a PB-SMT pipeline, notably the phrase extraction process and the translation model. Syntax-based approaches to MT are discussed in section 2.3 including the Data-Oriented Translation model and the Statistical Transfer engine used during our experiments in Chapters 3 and 5 respectively. Finally, in section 2.4, we describe the various metrics used to carry out automatic evaluation of translation quality throughout this thesis.

2.1 Parallel Treebanks

Parallel treebanking is a relatively recent concept which has stemmed from a combination of interest in the development of monolingual treebanks and parallel corpora. A parallel treebank is defined as a sententially aligned parallel corpus in which both the source and target sides are annotated with a syntactic tree structure and the sen-

tences are aligned at sub-sentential level (word, phrase and clause level) (Volk and Samuelsson, 2004; Samuelsson and Volk, 2006). The sub-sentential alignments hold the implication of translational equivalence between the constituents dominated by the aligned node pair. An example parallel treebank entry is illustrated in Figure 2.1.

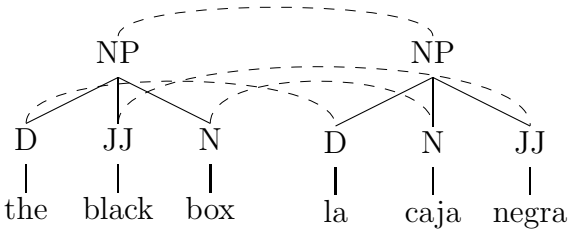


Figure 2.1: An example English–Spanish parallel treebank entry depicting syntactically annotated trees and sub-sentential alignments.

Parallel treebanks are a rich linguistic resource which can be used across a variety of NLP tasks, e.g. MT, translation studies and grammar inference amongst others, as demonstrated at the 2006 International Symposium on Parallel Treebanks.[1] Building parallel treebanks, however, is a non-trivial task. Manual construction is an expensive, time-consuming and error-prone process which requires linguistic expertise in all languages in question.[2] Because of this, parallel treebanks are not widely available in the NLP community, and those that are available tend to be too small for tasks such as data-driven MT. Table 2.1 presents a list of parallel treebanks known to us at the time of writing along with further information on their makeup.

Recent advances in monolingual parsing e.g. Bikel (2002); Nivre et al. (2007); Petrov and Klein (2007), have paved the way for automatic generation of parallel treebanks by providing the necessary architecture for syntactic annotation. What still remains, however, is a means to automatically induce sub-sentential relations between parallel trees. For the remainder of this section, we discuss parallel treebanks and alignment in terms of context-free phrase structure trees.

[1] http://www.ling.su.se/DaLi/education/parallel_treebank_symposium_2006

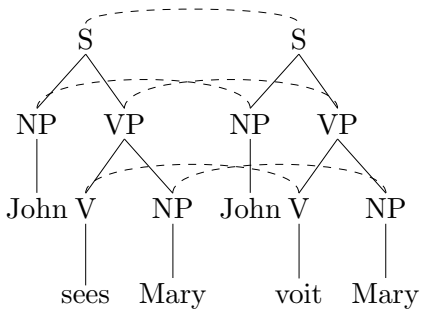
[2] As with parallel corpora (cf. Europarl (Koehn, 2005)), parallel treebanks can be built across more than two languages e.g. the SMULTRON English–German–Swedish parallel treebank (Gustafson-Čapková et al., 2007).

Reference	Languages	#Treepairs
Čmejrek et al. (2004)	Cz–En	21,600
Gustafson-Čapková et al. (2007)	Sv–De–En	∼1,473
Han et al. (2002)	Ko–En	5,083
Ahrenberg (2007)	Sv–En	1,180
Megyesi et al. (2008)	Sv–Tu	n/a^*
Hansen-Schirra et al. (2006)	De–En	$n/a^\dagger$

Table 2.1: Summary of reported parallel treebanks. *This parallel treebank contains 140,000 Swedish tokens and 165,000 Turkish tokens, but no details were reported on the number of tree pairs. †No size of any kind was reported in the literature for this parallel treebank.

2.1.1 Sub-sentential Alignment

The tree-to-tree alignment process assumes a parsed, translationally equivalent sentence pair and involves introducing links between non-terminal nodes in the source and target trees. Inserting a link between a node pair indicates that the substrings dominated by those nodes are translationally equivalent, i.e. that all the meaning in the source substring is encapsulated in the target string and vice versa. An example aligned English–French tree pair is given in (2.1). This illustrates the simplest possible scenario: the sentence lengths are equal, the word order is identical and the tree structures are isomorphic.



(2.1)

However, most real-world examples do not align so neatly. The example given in Figure (2.2) illustrates some important points. Not every node in each tree needs to be aligned, e.g. *es* translates not as *is*, but as *she is*,[3] yet each node is aligned at most once. Additionally, as we do not link terminal nodes, the lowest links are at the part-of-speech (POS) level. This allows for 1-to-many alignments between

[3]We can not align to *she is* as it does not correspond to a single constituent node in the tree.

single lexical items and phrasal constituents, e.g. the alignment between *housewife* and *ama de casa*. Furthermore, depending on the parsing scheme, a phrase like *ama de casa* may be realised as a multi-word unit (MWU). Aligning at POS level also allows us to preserve such MWUs during alignment.

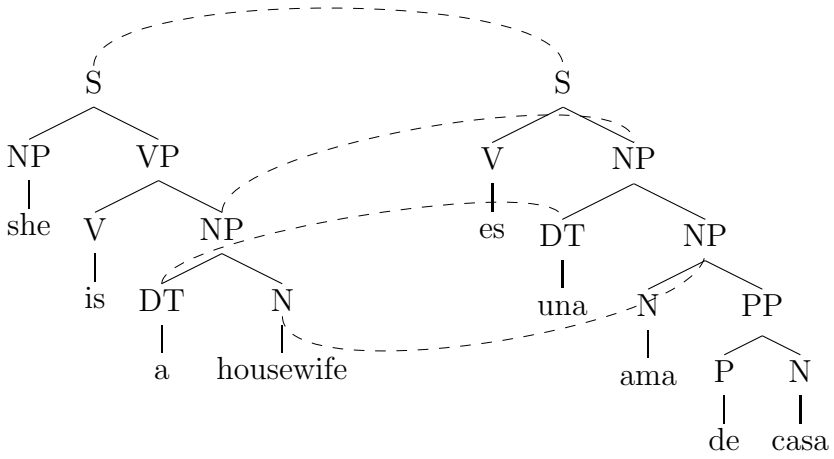


Figure 2.2: Example of a tree pair exhibiting lexical divergence.

Tree Alignment vs. Word Alignment

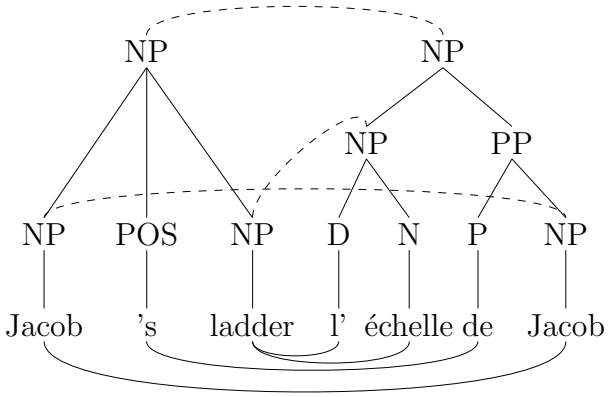
When deciding how to go about sub-sententially aligning a given tree pair, the logical starting point would seem to be with word alignment. However, some analysis reveals the differences between the tasks of tree alignment and word alignment. We illustrate the differences by referring to the Blinker annotation guidelines (Melamed, 1998) which were used for the word alignment shared tasks at the workshops on *Building and Using Parallel Texts* at HLT-NAACL 2003[4] and ACL 2005.[5]

According to these guidlines, if a word is left unaligned on the source side of a sentence pair, it implies that the meaning it carries was not realised anywhere in the target string. On the other hand, if a node remains unaligned in a tree pair there is no equivalent implication. Because tree alignment is hierarchical, many other nodes can carry indirect information regarding how an unaligned node (or group of unaligned nodes) is represented in the target string, e.g. *she is* $\leftrightarrow$ *es* in Figure 2.2. Some consequences of this are as follows.

[4]http://www.cse.unt.edu/∼rada/wpt
[5]http://www.cse.unt.edu/∼rada/wpt05

Firstly, the strategy in word alignment is to leave as few words unaligned as possible "even when non-literal translations make it difficult to find corresponding words" (Melamed, 1998). Contrast this with the more conservative guidelines for tree alignment given in Samuelsson and Volk (2006): nodes are linked only when the sub-strings they dominate "represent the same meaning ... and could serve as translation units outside the current sentence context". This latter strategy is affordable because alignments at higher levels in the tree pair will account for the translational equivalence. Secondly, word alignment allows many-to-many alignments at the word level but not at the level of phrase alignments unless every word in the source phrase is linked to every word in the target phrase and vice versa. Tree alignment, on the other hand, allows each node to be linked only once but facilitates phrase alignment by allowing links higher up in the tree pair.

The constrasting effects of these guidelines are illustrated by the example given in (2.2)[6] where the dashed links represent tree alignments and the solid links represent word alignments. We see that the word alignment must link *ladder* to both *l'* and *échelle* whereas the tree alignment captures this with a single 1-to-many alignment between the nodes dominating the substrings *ladder* and *l'échelle*.



(2.2)

Note also that the word alignment explicitly links *'s* with *de* where the tree alignment does not; it is arguable as to whether these strings really represent precisely the same meaning. However, the relationship between these words is not ignored by the tree alignment; rather it is captured by the alignments between the three NP

[6]The sentence pair and word alignments were taken directly from Melamed (1998).

links in combination.

In fact, many different pieces of information can be inferred from the tree alignment given in (2.2) regarding the relationship between *s* and *de*, despite the fact that they are not directly linked. Examples exhibiting varying degrees of contextual granularity are given in Figure 2.3.

$$
\begin{array}{ccc}
\textit{'s} & \longrightarrow & \textit{de} \\
\text{X } \textit{'s} \text{ Y} & \longrightarrow & \text{Y } \textit{de} \text{ X} \\
\text{NP}_1 \textit{ 's } \text{NP}_2 & \longrightarrow & \text{NP}_2 \textit{ de } \text{NP}_1 \\
\text{NP} \rightarrow \text{NP}_1 \textit{ 's } \text{NP}_2 & : & \text{NP} \rightarrow \text{NP}_2 \textit{ de } \text{NP}_1
\end{array}
$$

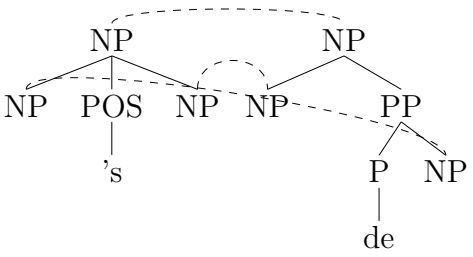


Figure 2.3: Example of varying granularity of information encapsulated in a tree alignment.

The 'rules' in Figure 2.3 are representative of the type of information encoded in parallel treebanks that is exploitable in syntax-based MT systems, as we will show in section 2.3.1.

2.1.2 Automatic Approaches to Tree Alignment

There have been numerous approaches proposed for the automatic induction of subtree alignments. It should be noted, however, that none of these approaches were designed with the explicit intention of building parallel treebanks, but rather with some other end-task in mind. An early algorithm was presented by Kaji et al. (1992) who made use of bilingual dictionaries to infer correspondences between ambiguous chart parses for the extraction of EBMT-style translation templates. Imamura

(2001) describes an approach to alignment which begins with statistically induced word alignments and proceeds to align at phrase level using heuristics based on lexical similarity and constituent labelling. Eisner (2003) describes an approach to tree alignment for dependency structures which performs expectation maximisation (Dempster et al., 1977) over all possible alignment hypotheses in order to select the optimal set. However, this approach, which can also be applied to phrase-structure trees, is very computationally expensive. An inspiration for the work presented in Chapter 3 of this thesis, the rule-based approach to French–English sub-tree alignment of Groves et al. (2004) (which in turn is influenced by the dependency-based alignment approach of Menezes and Richardson (2003)), firstly extracts a bilingual dictionary automatically using statistical techniques. The dictionary is then applied in conjunction with a number of hand-crafted rules to induce alignments. This method was employed to extract synchronous tree-substitution grammars for data-oriented translation (cf. section 2.3.2). A more recent approach is presented in Lavie et al. (2008) who use a clever mathematical trick based on prime factorisation to induce sub-tree alignments in order to create training data for their statistical transfer-based MT engine (cf. section 2.3.1). However, this approach is superceded by that of Ambati and Lavie (2008) who induce a statistical word alignment between the words in the tree pairs and then allow all hierarchical alignments which are consistent with the word alignment. In addition to this, Ambati and Lavie present an extension to this algorithm in which target trees are restructured in order to increase isomorphism with the source tree. The intended effect of this is to increase the number of alignments induced and consequently improve the coverage of the MT system trained directly on the aligned output. In his Ph.D. thesis, Zhechev (2009)[7] presents a detailed comparison of the approaches described in Ambati and Lavie (2008) and our novel method presented in Chapter 3.

We take a somewhat different perspective on tree alignment than that of Wellington et al. (2006) for example, who view trees as constraints on alignment. Our pur-

[7]Ventsislav Zhechev was a collaborator on the work presented in Chapter 3 of this thesis.

pose in aligning monolingual syntactic representations is to build parallel treebanks which make explicit the syntactic divergences between sentence pairs rather than homogenising them; significant structural and translational divergences are to be expected across different languages. We are not seeking to maximise the number of links between a given tree pair, but rather find the set of links which most precisely expresses the translational equivalences between the tree pair. In Chapter 3, we present a novel algorithm for the automatic induction of sub-sentential alignments between parallel trees reflecting this philosophy.

Our motivation for developing such a tool stems from the desire to build large-scale parallel treebanks for data-driven MT training. A further requirement to this end is that the algorithm is language pair-independent and preferably makes use of minimal external resources beyond (say) a statistical word aligner (cf. section 2.2.1). While the methods outlined above achieved competitive results in their reported tasks, none of them met all of our prerequisites (as summarised in Table 2.2) and so we felt it better to develop our own approach in order to ensure that our objectives were closely matched.

Prerequisite	**Kaji..'92**	**Groves..'04**	**Imamura'01**	**Ambati&Lavie'09**
Preserve Trees	$\sim$	✓	$\sim$	✓
Language Indepdendent	✓	×	✓	✓
Labelling Indepedent	×	✓	×	✓
Task Independent	×	✓	✓	×
No External Resources	×	×	✓	✓

Table 2.2: Summary of previous approaches to sub-tree alignment relative to our needs.

2.2 Phrase-Based Statistical Machine Translation

Statistical Machine Translation (SMT) (Brown et al., 1990, 1993) has dominated the research landscape of MT for most of the last decade. Originally based on the noisy channel approach for speech recognition, the SMT model exploits Bayes' Theorem, given in (2.3), to reformulate the automatic translation problem.

$$p(t|s) = \frac{p(s|t).p(t)}{p(s)} \tag{2.3}$$

In (2.3), $p(t|s)$ represents the likelihood that a target language translation t will be produced given a source language input sentence s. As $p(s)$ is constant for each value of t considered, we can find the most likely translation by maximising the probability of t in $p(t|s)$ as shown by the equation in (2.4).

$$\arg\max_t p(t|s) = \arg\max_t p(s|t).p(t) \tag{2.4}$$

In this equation, we maximise the product of the two remaining probabilities: $p(s|t)$, the probability of a candidate translation t being translated as s,[8] and $p(t)$, the probability of the candidate translation t being produced in the target language, known as the **translation model** (TM) and the **language model** (LM) respectively in SMT nomenclature. We discuss these aspects of the model further in sections 2.2.2 and 2.2.4. Finding the value of t which maximises (2.4) is thus a search problem, referred to as **decoding**, and is discussed in more detail in section 2.2.5. Given these definitions, we can further simplify the equation in (2.4) as shown in (2.5).

$$\arg\max_t p(t|s) = \arg\max p_{TM} p_{LM} \tag{2.5}$$

In initial incarnations of SMT, the fundamental unit of translation was the word. Given a parallel corpus of sententially aligned bilingual data, word-to-word correspondences were learned using algorithms which induced a set of mappings, or **word alignments**, between the source and target sentences (Brown et al., 1993). However, these word-based models were inadequate as they were unable to translate well between language pairs with high 'fertility'.[9] Thus, word-based systems ran into dif-

[8]Note the translation direction is reversed from a modelling standpoint when using Bayes' theorem.

[9]Fertility is the ratio of the lengths of sequences of translated words. A high fertility language pair is one in which single source words often correspond to multiple target words.

ficulty if (say) a sequence of source language words mapped to only a single target language word. This issue was overcome with the development of phrase-based SMT (PB-SMT) models (Marcu and Wong, 2002; Koehn et al., 2003), which allow for the mapping of sequences of n words in the source language, so-called phrases, to sequences of m words in the target language. However, these phrase pairs are still learned using the original word alignment techniques of Brown et al. (1993). Decoding for PB-SMT is carried out in much the same way as for word-based models by searching for the most likely sequence of target language candidates matching the source language input, given a translation model and a language model.

The end-to-end translation process of a PB-SMT system can be broken down into a number of sequential steps, forming a pipeline. Given a parallel corpus, this process proceeds roughly as follows:

- A set of **word alignments** are induced between the source and target sentences in the parallel corpus (Brown et al., 1993; Och and Ney, 2003).
- Phrase pair correspondences are learned given these alignments and used to build a weighted **translation model** (Och and Ney, 2003, 2004).
- A **language model** is estimated for the target language (Stolcke, 2002).[10]
- A **decoder** takes the translation and language model and searches for the optimal target language translation given some source language input (Koehn et al., 2007).

Obviously, some of the details of the various stages mentioned above have been underspecified here. In the remainder of this section, we describe these steps in the PB-SMT pipeline in greater detail, paying particular attention to those aspects pertinent to our work in this thesis.

[10]This is sometimes estimated from the target language side of the parallel training corpus, but any amount of target language data can be used.

2.2.1 Word Alignment

Word alignment – the task of determining translational correspondences at lexical level in a parallel corpus – is not only the starting point in the PB-SMT pipeline, but also a fundamental component in all SMT variants as well as numerous other NLP tasks. An example word alignment is shown in Figure 2.4.

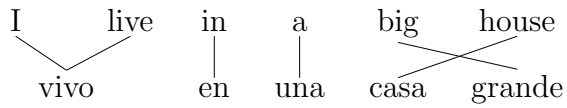


Figure 2.4: An example of an English-to-Spanish word alignment.

In this example, where the connecting lines between words represent alignments, we can see some of the challenges of inducing word alignments. For instance, the fertility issue mentioned previously where a single word in one language can align to many words in the other is demonstrated where the Spanish word *vivo* aligns to the two English words *I live*. The most common approach to word alignment is to use generative models. The first and most popular instance of generative word alignment models are the so-called 'IBM Models' (Brown et al., 1990, 1993) which describe a number of different models for the induction of word alignments. The first two models, IBM Models 1 and 2, are non-fertility models: they do not allow for 1-to-many alignments. These models operate using expectation maximisation, firstly assuming a uniform distribution between all source and target words, and then learning a refined distribution by iterating over the data. The remaining models, IBM Models 3–5, are more complicated as they introduce fertility. That is, these models first determine the fertility of each source word, e.g. *not* $\rightarrow$ *ne...pas* would mean *not* has a fertility of 2 (French words). The target words are then rearranged to produce a target string according to the model. This is known as a 'distortion' model. In IBM Model 3, each target word aligned to a particular source word is positioned independently, whereas in IBM Model 4 target word positioning has a first-order dependence, i.e. the context of the neighbouring previous word is considered. These

models allow for some target words to be assigned the same position in the target string in order to simplify training. This so-called 'deficiency' is resolved in IBM Model 5.

All of these models are implemented in a freely available open source toolkit called GIZA++ (Och and Ney, 2003).[11] Throughout the course of this thesis, we employ IBM Model 4[12] as implemented in GIZA++ when we carry out word alignment.

2.2.2 Phrase Extraction and Translation Models

Phrase extraction is the process of learning translationally equivalent pairs which may span sequences of n words. As we mentioned previously, word-based SMT systems learn lexical translation models describing one-to-one mappings between a given language pair. However, words are not the best units of translation because we can have fertility between languages. Furthermore, by translating word for word, no contextual information is made use of during the translation process. In order to overcome this, PB-SMT models translate together certain sequences of words, so-called phrases (not phrases in the linguistic 'constituent' sense of the word). By using phrases as the core translation unit in the model, it is possible to avoid many cases of translational ambiguity and better capture instances of local reordering. An example of this is illustrated in Figure 2.5.

There are a number of ways to extract a phrase table from a parallel corpus. In this section, we describe in detail the commonly used method which we employ throughout the course of this thesis, while providing a brief summary of alternative approaches. The basis for phrase extraction from a parallel corpus is the word alignment described in the previous section. For each word-aligned sentence pair, a set of phrase alignments that is *consistent* with the word alignment is extracted.

[11] http://www.fjoch.com/GIZA++.html

[12] IBM Model 4 is the default setting for GIZA++. Due to the large number of parameters which must be estimated for IBM Model 5, it takes significantly longer to train than Model 4 yet the gains in performance are not that much. For this reason, we believe Model 4 is sufficient to demonstrate our hypotheses in this thesis.

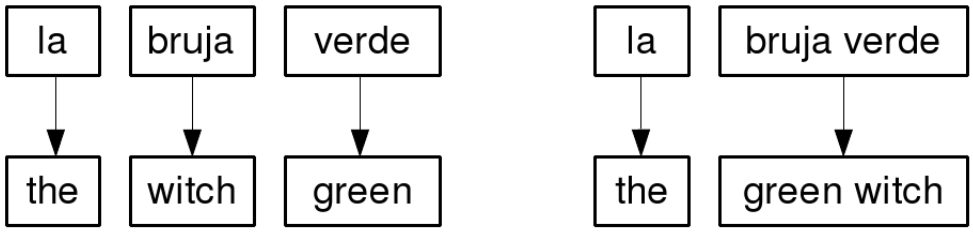


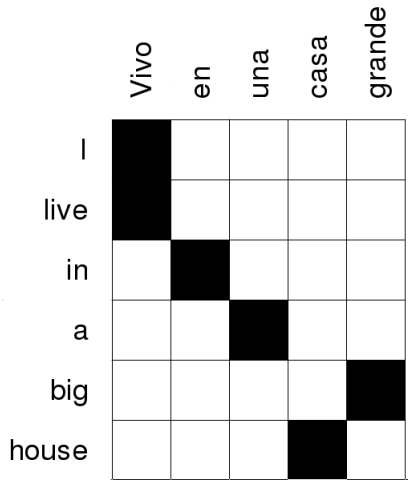
Figure 2.5: In the word-based translation on the left we see the noun-adjective reordering from Spanish into English is missed. On the right in the phrase-based translation, the noun and adjective are translated as a single phrase and the correct ordering is modelled.

Consider Figure 2.6, which illustrates the word alignment of Figure 2.4 as a matrix in which the blackened squares represent alignments. If we take, for example, the two word alignments *big* → *grande* and *house* → *casa*, we can extract the phrase pair *big house* ↔ *casa grande* as the words in the source phrase are only aligned to words in the target phrase and vice versa. Below the matrix in Figure 2.6, we see the entire set of phrase pairs extractable from this sentence pair.

A more formal definition of *consistency* is as follows: a phrase pair $(\bar{s}|\bar{t})$ is consistent with an alignment A, if all words $s_1,\ldots,s_n$ in $\bar{s}$ that have alignment points in A have these with words $t_1,\ldots,t_n$ in $\bar{t}$ and vice versa (Koehn, 2009). The phrase extraction process proceeds by extracting all phrase pairs for a given sentence pair that are consistent with the word alignment.

Refined Word Alignments for Phrase Extraction

Both the quality and the quantity of word alignments have a significant effect on the extracted phrase translation model. Obviously, the more accurate the word alignments the better the quality of the subsequently extracted phrase pairs. Word alignment is a directional task, and the IBM models allow for a target word to be aligned to (at most) one source word. This is undesirable as it may be correct in many instances to have a target word map to multiple source words. In order to overcome this problem, we carry out *symmetrisation* of the word alignments (Och et al., 1999).



I live	↔	Vivo
I live in	↔	Vivo en
I live in a	↔	Vivo en una
I live in a big house	↔	Vivo en una casa grande
in	↔	en
in a	↔	en una
in a big house	↔	en una casa grande
a	↔	una
a big house	↔	una casa grande
big	↔	grande
big house	↔	casa grande
house	↔	casa

Figure 2.6: English–Spanish word alignment matrix and the entire set of extractable phrase pairs.

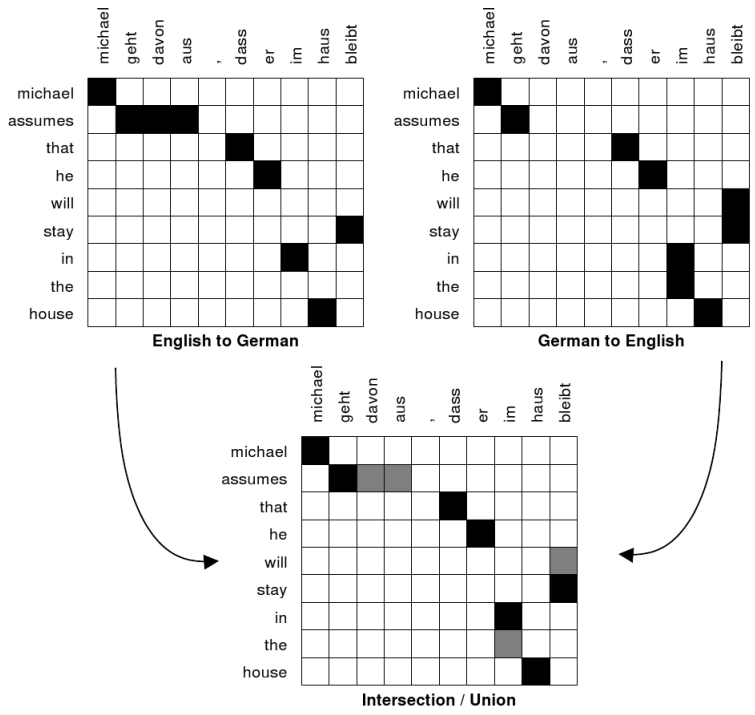


Figure 2.7: Merging source-to-target and target-to-source alignment sets by taking their union (from Koehn (2009)).

As illustrated in Figure 2.7, this process involves running word alignment in both directions: source-to-target and target-to-source. The resulting sets of alignments are then merged by taking their union or intersection. Generally, choosing between the union and intersection of the word alignments involves deciding whether we want a high recall or a high precision word alignment. Koehn et al. (2003) demonstrated that for PB-SMT the best option is to explore the space between the union and the intersection. This is done using heuristics initially proposed by Och et al. (1999) and extended upon in Koehn et al. (2003), which begin with the alignment points in the intersection and then *grow* the alignment, progressively adding neighbouring alignment points from the union. A neighbouring point, as illustrated by the shaded squares in Figure 2.8, is any hypothetical alignment point in the matrix that occurs in the direct vicinity of an existing alignment point. This stage of the heuristic is known as *grow-diag*. It can be further extended by allowing additional points from the union with the only restriction being that the source and target words in question must be heretofore unaligned. This extension is known as *-final*.

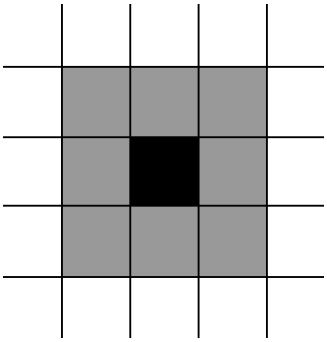


Figure 2.8: Example of neighbouring alignment points: the black square is the alignment points in question and the shaded squares are its neighbouring points.

In all experiments in this thesis, we perform phrase extraction on the source–target intersection refined with the *grow-diag-final* heuristic as implemented in the Moses toolkit (Koehn et al., 2007).[13]

2.2.3 Scoring and the Log-Linear Model

A probability distribution is estimated over the set of phrase pairs, extracted using the methods of the previous section, where the probability of a phrase pair $P(s|t)$ is its relative frequency in the entire set of phrase pairs, as in 2.6:

$$P(\bar{s}|\bar{t}) = \frac{count(\bar{t}, \bar{s})}{\sum_{\bar{s}_i} count(\bar{t}, \bar{s}_i)} \tag{2.6}$$

Traditionally, this function would be included in the noisy channel model along with the language model. However, more recent research in SMT has departed from this approach, adopting a more flexible model structure known as a log-linear model (Och and Ney, 2002; Och et al., 2004). This model is extensible and allows for the addition of new features to the system beyond the translation and language models. Furthermore, each feature h_i is assigned a weight λ_i which can be optimised given some objective function (normally BLEU score (Papineni et al., 2002), cf. Section 2.4.1) using a tuning algorithm, e.g. minimum error-rate training (MERT) (Och, 2003) or the margin infused relaxed algorithm (MIRA) (Chiang et al., 2009). The

[13]Moses is a widely used, free and open-source SMT system which implements many of the components described in this chapter. It is available from http://www.statmt.org/moses/

formula for the log-linear PB-SMT model is given in (2.7).

$$P(t|s) = \arg\max_{t}\{\sum_{m=1}^{M} \lambda_m h_m(t,s)\} \tag{2.7}$$

Theoretically, any number of feature functions can be used in the log-linear model,[14] e.g.(Chiang et al., 2009). However, in our experiments presented throughout this thesis we make use of seven features as implemented in Moses (unless otherwise stated). These features are:

- phrase translation probabilities, both source-to-target and target-to-source;
- an n-gram language model, discussed in section 2.2.4;
- a reordering model;
- a phrase penalty;
- lexical weights, again source-to-target and target-to-source.

The reordering model accounts for the movement of phrases during translation. For example, when translating from English into German, we may want to move the verb to the end of the translated sentence. Moses implements a distance-based reordering model which estimates, for each extracted phrase pair, how often it occurred out of continuous order in the aligned training data. Three different orientations are modelled: monotone, the phrase occurred in order; swap, the phrase swapped one position with another phrase; and discontinuous, the phrase occurred completely out of order with the rest of the extracted phrases.

The phrase penalty is a means to bias towards longer phrase pairs when building translation hypotheses, the motivation being that the less we segment an input sentence in to phrases, the more reliable the longer phrases will be as they will contain more context. Thus, by penalising shorter phrases, if the model has the choice of using a longer phrase during decoding, it will tend to use it.

[14] Although training may take some time if there are too many!

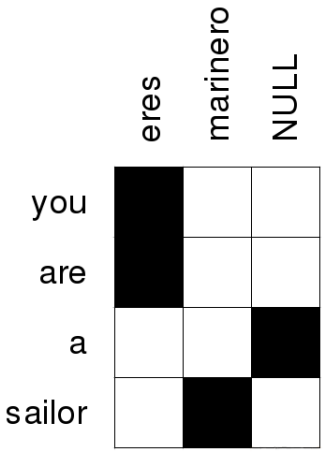
The lexical weighting feature (Koehn et al., 2003) allows for further validation of extracted phrase pairs by checking how well the words in the source and target sides of a given phrase pair translate to one another. It helps to ensure that good rare phrases, which will have a low probability given the phrase translation distribution, can still be used, by exploiting richer lexical translation statistics. This is done using a lexical translation probability distribution $lex(s|t)$ estimated by relative frequency from the same set of word alignments used for phrase extraction, according to (2.8).

$$lex(s|t) = \frac{count(s,t)}{\sum_{s'} count(s',t)} \tag{2.8}$$

Then, given a phrase pair $(\bar{s},\bar{t})$ and a word alignment a between source word positions i and target positions j, a lexical weight p_{lex} is calculated via the equation in (2.9).

$$p_{lex}(\bar{s}|\bar{t},a) = \prod_{i=1}^{length(\bar{t})} \frac{1}{|\{j|(i,j)\in a\}|} \sum_{\forall(i,j)\in a} lex(s_i|t_j) \tag{2.9}$$

If multiple source words are aligned to a single target word, the average word translation probability is taken. In addition to this, to account for cases in which a source word has no alignment on the target side, a special NULL word is added to the target string and the probability of the source word translating as NULL given the distribution is calculated. This process is exemplified in Figure 2.9, where we have the English source phrase *you are a sailor* aligned to the Spanish target phrase *eres marinero*. The two English words *you are* are aligned to the Spanish word *eres*, so we calculate the average of both words translating as the target word. The English word *a* has no alignment on the Spanish side, so we calculate $lex(a|\text{NULL})$ from the lexical translation distribution. Finally, the English word *sailor* is aligned to *marinero* so we calculate $lex(sailor|marinero)$. We calculate this lexical weighting feature in both translation directions – $p_{lex}(\bar{s}|\bar{t})$ and $p_{lex}(\bar{t}|\bar{s})$ – using our source-to-target and target-to-source word alignments, and these two additional features are added to the log-linear model.



$$p_{lex}(\bar{s}|\bar{t}) = \frac{1}{2}\,(lex(\text{you}|\text{eres}) + lex(\text{are}|\text{eres})) \times lex(\text{a}|\text{NULL}) \times lex(\text{sailor}|\text{marinero})$$

Figure 2.9: An example of how lexical weighting is calculated for an English–Spanish sentence pair.

As we mentioned earlier, the optimal weight for each of these features, based on some development corpus, is assigned using a tuning algorithm, optimising usually on the BLEU metric. Throughout this thesis, we employ the MERT optimisation algorithm as implemented in the Moses toolkit.

2.2.4 Language Modelling

The language model feature p_{LM}, mentioned at the beginning of this section in terms of the noisy-channel model, measures how likely it is that a hypothetical translation proposed by the translation model exists in the target language. This is done by calculating how likely a word is to occur given its history, i.e. all the preceding words in the string, as shown in (2.10).

$$p(w_1, w_2, \ldots, w_n) = p(w_1)p(w_2|w_1)\ldots p(w_n|w_1, w_2, \ldots, w_{n-1}) \tag{2.10}$$

However, calculating probabilities for all possible histories is impractical as sparse data issues would lead to unreliable statistics. For this reason, the history is limited

to n words, giving rise to the term n-gram language modelling. Most commonly, values of n between 3–5 are used for MT. In order to estimate trigram model[15] probabilities for a word sequence $p(w_3|w_1, w_2)$, we count how often w_3 is preceded by the sequence w_1, w_2 in some training corpus. This is done according to maximum likelihood estimation (Manning and Schütze, 1999, p. 197) as shown in (2.11).

$$p(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{\sum_w count(w_1, w_2)} \tag{2.11}$$

The example in (2.12) demonstrates how the probability of the sentence "you are a sailor" is calculated given an English trigram language model.[16]

$$\begin{aligned} \text{p(you,are,a,sailor)} \approx \quad & \text{p(you <s>,<s>)} \times \\ & \text{p(are | <s>,you)} \times \\ & \text{p(a | you,are)} \times \\ & \text{p(sailor | are,a)} \end{aligned} \tag{2.12}$$

Despite the fact that language models are often trained on large amounts of monolingual data, we still run into sparse data issues as the likelihood is high that we will encounter some n-gram in our translation output that was not seen in our training data. In order to counteract this problem, a number of smoothing methods are applied, for example weighted linear interpolation (Manning and Schütze, 1999, p. 322). Taking this approach, we estimate probabilities over all values of n up to our maximum (3) and take the sum of these values, weighting the model orders as required. For a trigram language model, this means calculating unigram, bigram, and trigram scores for each input string including some smoothing in the case a word was not observed in the training data. This is illustrated in (2.13), where V is the vocabulary size and λ_n is the weight assigned to each order of n.

[15]For clarity, we will explain language models in terms of trigrams for the remainder of this section.

[16]The symbol <s> signifies the beginning of the sentence.

$$
\begin{aligned}
p(w_3|w_1, w_2) = \quad & \lambda_3 p(w_3|w_1, w_2) + \\
& \lambda_2 p(w_3|w_2) + \\
& \lambda_1 p(w_3) + \\
& \lambda_0 \tfrac{1}{V}
\end{aligned}
\tag{2.13}
$$

The intend effect of this approach is that, for a given input string, if we have never seen a particular trigram in our training data, rather than assigning it zero score, we essentially backoff and see if we have observed two of the words cooccurring, or even any of the words individually.

In our experiments in this thesis, we employ language models as implemented using the SRI Language Modelling (SRILM) toolkit (Stolcke, 2002)[17] which also provides for the use of modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1996). In this approach to smoothing, which is in a similar vein to weighted linear interpolation, rather than explicitly weighting the higher order n-grams, a discount is subtracted based on estimation using a held-out set. Furthermore, backing off to the lower-order models in the interpolation is only considered when the score for the higher order models is very low. This helps to ensure that the best fitting model is chosen.

2.2.5 Decoding

The final phase in the PB-SMT pipeline involves generating the most likely target language string given some source input.[18] This process is known as decoding, and involves searching through the phrase table to find the $P(t|s)$ that maximises the sum of feature functions $h_1, \ldots, h_m$ in the log-linear model. It proceeds by constructing the output translation based on some segmentation of the input, incrementally computing the translation probability. Evaluating all possible target strings, how-

[17] http://www.speech.sri.com/projects/srilm/

[18] We note that up to this point in the pipeline, no actual translation has been carried out.

ever, is an NP-complete problem (Knight, 1999) and so heuristic methods must be applied. The most common approach, as implemented in the Moses toolkit, is to use a beam search algorithm.

Following this approach, partial translation hypotheses are arranged in stacks based on the number of input words they cover, as illustrated in Figure 2.10. These stacks are pruned as required in order to keep the search space size manageable. Two methods for pruning are commonly used: in *histogram pruning*, a maximum of n hypotheses are stored in a stack at any one time (the n highest scoring hypotheses), while in *threshold pruning*, hypotheses with a running probability which differs from the current best hypothesis by more than a fixed threshold value α are discarded. Adjusting these values allows for some compromise between speed and quality of translation, e.g. the higher the value we have for n the larger the search space will be, but the lower the chance we will have pruned out the best translation.

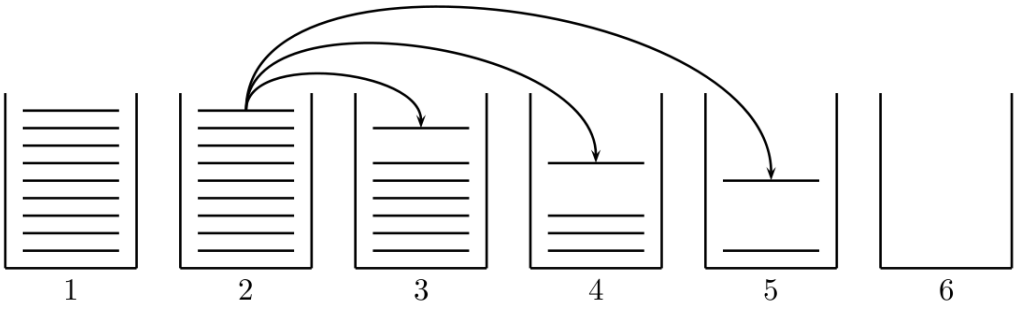


Figure 2.10: Hypothesis stacks: Partial translations are placed in stacks based on the number of input words covered (the indices below each stack) and expanded into new stacks (as indicated by the arrows) as new words are translated (from Koehn (2009)).

The translation process is initialised by creating an empty hypothesis stack. Then, for all possible segmentations of the input string, translation options are added to stacks and new stacks are created as hypotheses are expanded to cover more of the input string. Probabilities for the new hypotheses are updated and pruning of weak hypotheses is carried out as necessary. Aside from the probability assigned according to the log-linear model, a future cost score is estimated for each hypothesis based on how difficult it will be to translate the remainder of the input

string. The intended effect of this is to balance the discrepancy in scores between those hypotheses which have so far translated "easy" parts of the input and those which have translated more difficult parts. The expansion of hypotheses continues until the entire input string has been covered, at which point the most probable hypothesis is output as the 1-best target language translation.

Throughout this thesis, we use the beam search decoder as implemented in Moses in our PB-SMT systems

PB-SMT: Summary

In this section, we have described the principal elements which comprise a PB-SMT system, highlighting the process by which phrase pair correspondences are extracted and employed in the translation model. In Chapter 4, we present experiments in which we exploit syntax-based resources — namely, automatically generated parallel treebanks — at various stages in the PB-SMT pipeline (particularly phrase extraction and in the log-linear model) in order to increase the syntactic awareness of the SMT framework.

2.3 Syntax-Based Machine Translation

From our description of phrase-based statistical MT as presented in previous section, it may be apparent that the entire end-to-end translation process has no linguistic motivation: word alignments are induced via statistical methods, phrase extraction is heuristics-driven etc. Syntax-based paradigms of MT, on the other hand, comprise those approaches to MT which exploit syntactically annotated data directly in training. There has been a significant amount of research concerning the incorporation of linguistic information into the PB-SMT process, e.g. Carpuat and Wu (2007); Koehn and Hoang (2007); Haque et al. (2009a,b); Hassan et al. (2009), and while many of these approaches have successfully achieved improvements in translation performance, they do not constitute fully syntax-based systems and,

thus remain restricted by the limitations of the PB-SMT framework, namely string-based decoding. While the development of syntax-based systems is not necessarily a new development — cf. the system of Yamada and Knight (2001); Germann et al. (2001) — there has been a trend in recent years within the MT community towards the development of such systems. In this section, we give details of the two syntax-based systems used in this thesis and summarise other recent developments in the area of syntax-based MT.

2.3.1 Statistical Transfer-Based MT

The CMU Statistical Transfer Framework (Stat-XFER) (Lavie, 2008) is a general framework for developing syntax-driven MT systems. The principal component of the framework is a syntax-based transfer engine which exploits two language pair-specific resources: a grammar of weighted synchronous context-free rules (SCFG), and a probabilistic bilingual lexicon. Translation is carried out in two phases; firstly, the lexicon and grammar are applied to synchronously parse the input sentence, producing a lattice of translation options. Following this, a monotonic decoder runs over the resulting lattice of scored translation segments to produce the final output. The decoder is monotonic as all necessary reordering is carried out based on the syntactic grammar during the transfer phase.

Bilingual Lexicon

The bilingual lexicon of the Stat-XFER system is an extension of the PB-SMT phrase table (cf. section 2.2.2) in which each side of the source–target translation pair is associated with a syntactic category. Each entry in the lexicon can be described formally as an SCFG expression, as demonstrated in (2.14), where c_s and c_t represent source- and target-side syntactic category labels respectively, and w_s and w_t represent the source- and target-side phrase strings.

$$c_s :: c_t \rightarrow [w_s] :: [w_t] \tag{2.14}$$

Entries are assigned two scores, $r_{(t|s)}$ and $r_{(s|t)}$, based on maximum-likelihood estimates. The $r_{(t|s)}$ score, calculated as per (2.15), is a maximum-likelihood estimate of the distribution of target language (TL) translations and source- and target-side category labels given the source language (SL) string. Conversely, the $r_{(s|t)}$ score is calculated as in (2.16) over the SL translations and syntactic categories given the TL string.

$$r_{(t|s)} = \frac{\#(c_t, w_t, c_s, w_s)}{\#(w_s) + 1} \tag{2.15}$$

$$r_{(s|t)} = \frac{\#(c_t, w_t, c_s, w_s)}{\#(w_t) + 1} \tag{2.16}$$

Add-one smoothing (Manning and Schütze, 1999, p. 202) is employed in the denominator to counteract overestimation of scores given low counts for w_s and w_t.

Stat-XFER Grammar

The Stat-XFER grammar rules have a similar form to the bilingual lexicon entries, as shown in (2.17). The SCFG rule can be lexicalised and may include both non-terminals and pre-terminals. Constituent alignment information, shown in (2.17) as co-indices on the nodes, indicate correspondences between the source- and target-side constituents. Rule scores $r_{(t|s)}$ and $r_{(s|t)}$ for the SCFG rules are calculated in the same manner as the scores for the bilingual lexicon entries.

$$NP :: NP \rightarrow [D^1 N^2 A^3] :: [DT^1 JJ^3 N^2] \tag{2.17}$$

Both of the resources described above – bilingual lexicon and the SCFG – can be extracted from parallel treebanks as we mentioned in section 2.1 (cf. Figure 2.3). We will demonstrate this in practice in Chapter 5.

Transfer Engine

The transfer engine, described in detail in (Peterson, 2002), carries out the three main processes involved in transfer-based MT: parsing of the SL input; transfer of the parsed SL constituents to their corresponding TL structures; and generation of the TL output. All processes are carried out using the SCFG in an extended chart parsing algorithm which operates by, firstly, populating a chart with the SL constituent using the left-hand side of the SCFG rules. A TL chart is constructed in parallel using the right-hand sides of the corresponding SCFG rules. The TL chart is then lexicalised by taking translation options for the source words from the bilingual lexicon. The TL chart maintains stacks of scored translation options for all substrings in the SL input which are ultimately collated into a lattice that is passed on to the decoder. The decoder employed is akin to that described in section 2.2.5 without a reordering model. An illustration of the entire end-to-end translation process is shown in Figure 2.11.

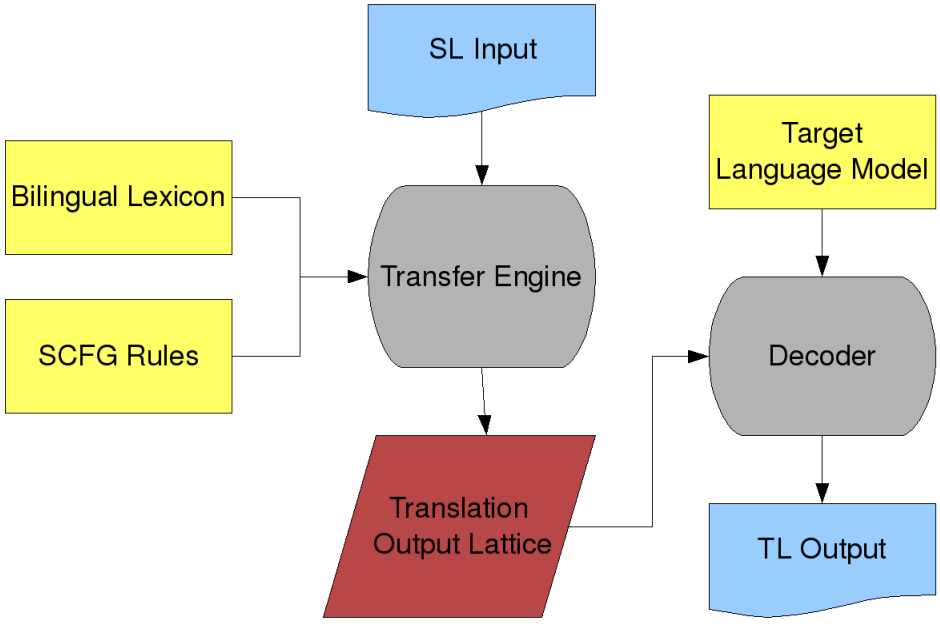


Figure 2.11: Architecture of Stat-XFER translation framework (adapted from Lavie (2008)).

The Stat-XFER framework has been used to build small-scale MT systems for lesser resourced language by exploiting manually-crafted resources (Lavie, 2008;

Monson et al., 2008), while also being employed in large-scale MT evaluation tasks (Hanneman et al., 2008, 2009), which demonstrates its scalability. Additionally, there has been significant research in the area of resource extraction for Stat-XFER systems from parallel treebanks (Lavie et al., 2008) and in tree-to-string scenarios (Ambati and Lavie, 2008; Ambati et al., 2009).

In Chapter 5, we describe the construction of a number of Stat-XFER systems using bilingual lexicons and SCFGs extracted from automatically generated parallel treebanks.

2.3.2 Data-Oriented Translation

Data-Oriented Translation (DOT) (e.g. (Poutsma, 2003; Hearne and Way, 2006)), which is based on Data-Oriented Parsing (DOP) (e.g. (Bod et al., 2003)), combines examples, linguistic information and a statistical translation model. Tree-DOT assumes a sub-sententially aligned parallel treebank as direct training data, such as the one given in Figure 2.12(a), from which it learns a generative model of translation. This model takes the form of a synchronous stochastic tree-substitution grammar (S-STSG) whereby pairs of linked generalised subtrees are extracted from the linked tree pairs contained in the training data via *root* and *frontier* operations:

- given a copy of tree pair $\langle S, T \rangle$ called $\langle S_c, T_c \rangle$, select a **linked** node pair $\langle S_N, T_N \rangle$ in $\langle S_c, T_c \rangle$ to be *root* nodes and delete all except these nodes, the subtrees they dominate and the links between them, and
- select a set of **linked** node pairs in $\langle S_c, T_c \rangle$ to be *frontier* nodes and delete the subtrees they dominate.

Thus, every fragment $\langle f_s, f_t \rangle$ is extracted such that the root nodes of f_s and f_t are linked, and every non-terminal frontier node in f_s is linked to exactly one non-terminal frontier node in f_t and vice versa. Some fragments extracted from the tree pair Figure 2.12(a) are given in Figure 2.12(b).

During translation, fragments are merged in order to form a representation of the source string within which a target translation is embedded. The composition operation ($\circ$) is a leftmost substitution operation; where a fragment has more than one open substitution site, composition must take place at the leftmost site on the source subtree of the fragment. Furthermore, the synchronous target substitution must take place at the site *linked to* the leftmost open substitution site on the source side. This ensures (i) that each derivation is unique and (ii) that each translation built adheres to the translational equivalences encoded in the example base. An example composition sequence is given in Figure 2.12(c).

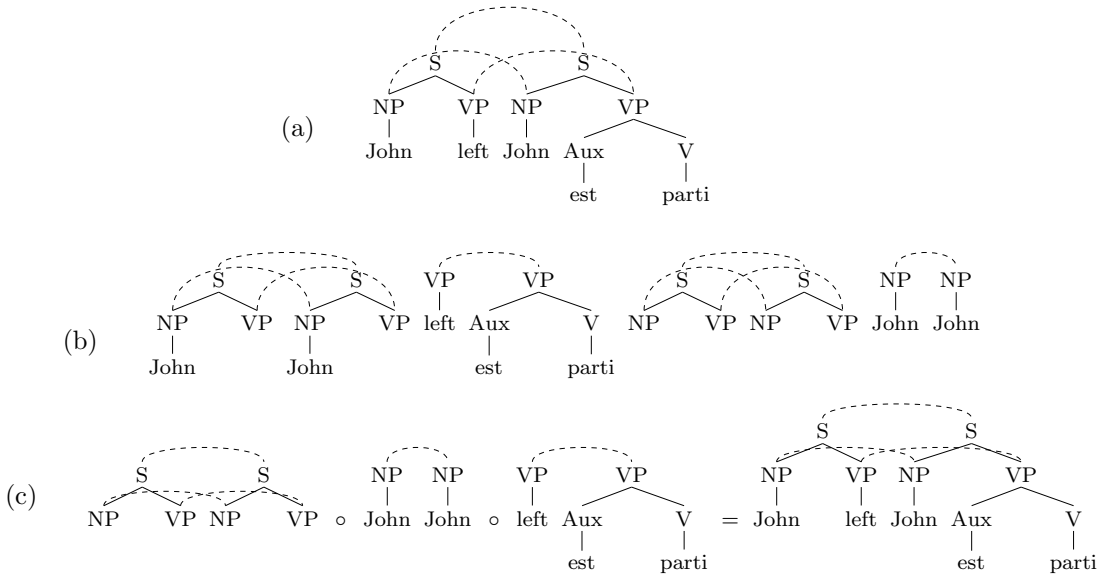


Figure 2.12: Data-Oriented Translation: (a) gives an example representation, (b) gives a subset of the possible fragments of (a) and (c) gives an example composition sequence yielding a bilingual representation.

Many different representations and translations can be generated for a given input string, and the alternatives are ranked using a probability model. Although there has been considerable research carried out into how best to estimate the probability model (Johnson, 2002; Bonnema and Scha, 2003; Sima'an and Buratto, 2003; Galron et al., 2009), the version of the DOT system employed in this thesis estimates fragment probabilities using relative frequencies and derivation probabilities computed by multiplying the probabilities of the fragments used to build them. For each

input string, the n-best derivations are generated and then reduced to the m-best translations where the probability of translation t is computed by summing over the probabilities of those derivations that yield it. Where no derivation spanning the full input string can be generated, the n-best sequences of partial derivations are generated instead and the translations ranked as above. Unknown words are simply left in their source form in the target string. Thus, every input string is translated but the system output indicates which strings achieved full coverage.

While the DOT model has yet to scale to larger data sets (it has to date been used with parallel treebanks of up to 10,000 sentence pairs (Galron et al., 2009)[19]), we exploit it in Chapter 3 to carry out an extrinsic evaluation of our sub-tree alignment algorithm given a relatively small training set.

2.3.3 Other Approaches

Further approaches to syntax-based MT have been developed in recent years incorporating varying degrees of linguistic information. Chiang (2005, 2007) present a *hierarchical* phrase-based model which allows for generalisations over sub-phrases within a baseline phrase table. This model, formally a weighted SCFG, can generate phrases in the target language output that were not previously seen in the training data by combining generalised templates with existing phrase table entries. Chiang makes the distinction between this model being *formally* rather than *linguistically* syntax-based as the generalised templates are not informed by any syntactic theory. However, there have been some efforts centred on extending the hierarchical model with varying degrees of syntactic constraints, during both the decoding phase (Marton and Resnik, 2008) and directly into the log-linear model during training (Vilar et al., 2008). Similarly, Zollmann and Venugopal (2006) and Zollmann et al. (2008) describe a "syntax-augmented" system in which the target side of the hierarchical translation model is syntactified and a number of new features are added

[19]The parallel treebank used in the work of Galron et al. (2009) was produced using the methods described in this thesis.

to a log-linear model.

Tree-to-string models, popularised in the aforementioned work of Yamada and Knight (2001), have also been widely developed. Aside from extensions to the Yamada and Knight (2001, 2002) model as seen in the work of Galley et al. (2004, 2006), Liu et al. (2006) present a tree-to-string alignment template model in which syntactically annotated source-side data is word-aligned to plain target language data and transformation templates are learned. At decoding time, the input sentence is parsed and a search algorithm applies the most appropriate set of transformation templates to generate the target language output. Similarly, using the projection technique of Ambati and Lavie (2008), as described in section 2.1.1, the Stat-XFER framework can also be applied to the tree-to-string scenario.

Finally, aside from the Stat-XFER framework and the DOT model, direct tree-to-tree models have also received some attention in recent years. Nesson et al. (2006) describe such a system, modelled as a probabilistic synchronous tree-insertion grammar, which efficiently translates via decisions trees during parsing of the input sentence. The authors espouse the flexibility of their approach with respect to linguistic formalism and potential for hybridity with other MT models, e.g. example-based MT. In addition to this, Bojar and Hajič (2008); Bojar et al. (2009) describe a system for English–Czech tree-to-tree translation at a deep syntactic (tectogrammatical) layer. Using parallel trees annotated with dependency information to the tectogrammatical layer, translation is modelled as an SCFG (similar to DOT), decomposing trees into a grammar of smaller treelets. Given the input, these trees are then composed to build target language output.

In Chapter 5, we demonstrate the effectiveness of parallel treebanks as a training resource for syntax-based MT, while in section 6.1, we discuss how we could potentially employ the techniques presented in this thesis to some of these approaches to syntax-based MT.

2.4 MT Evaluation

Over the last decade, automatic evaluation metrics have become an integral component in the development cycle of any MT system. They allow for fast, cheap and large-scale analysis of MT systems by comparing the output translations to one or more reference translations. This is based on the rationale that the closer the output translation is to the professionally produced reference translations, the better it is. In this section, we describe the three metrics we use for automatic evaluation in this thesis. We chose multiple metrics for evaluation as an improvement in a single metric cannot be guaranteed to indicate improved translation accuracy, as has been previously demonstrated (Callison-Burch et al., 2006; Chiang et al., 2008). However, if we see correlations across multiple metrics, we can be more confident in our findings. We chose these three metrics in particular as they are used extensively in large-scale MT evaluation campaigns and have become the *de facto* standard for the automatic evaluation of MT quality.

2.4.1 BLEU

The **BLEU** metric (Papineni et al., 2002) evaluates MT quality by comparing translations output by the MT system against one or more reference translations in terms of the number of co-occurring n-grams between the two strings. BLEU rewards those candidate translations with longer contiguous sequences of matching words. The main score calculated by this metric is a **modified n-gram precision** score p_n for each candidate translation and its reference(s). It is modified in that it avoids giving inflated precision to those candidates which overgenerate or repeat words. For example, if an n-gram occurs j times in the candidate translation and i times in a reference translation such that $i \leq j$, then this sequence is only counted i times. Thus, modified n-gram precision p_n is calculated according to the equation given in (2.18):

$$p_n = \frac{|c_n \cap r_n|}{|c_n|} \tag{2.18}$$

where

- c_n is the multiset of n-grams occurring in the candidate translation,
- r_n is the multiset of n-grams occurring in the reference translation,
- $|c_n|$ is the number of n-grams occurring in the candidate translation,
- $|c_n \cap r_n|$ is the number of n-grams occurring in c_n that also occur in r_n such that elements occurring j times in c_n and i times in r_n occur maximally i times in $|c_n \cap r_n|$.

Generally when automatically evaluating MT output, scores are calculated over a test set of sentences rather than on individual input strings. In this scenario, p_n is the proportion of co-occurring n-grams in the set over the total number of n-grams in that set.

While p_n can be calculated for any value of n, Papineni et al. (2002) mention that greater robustness can be achieved by combining scores for all values of n into a single metric. However, as the value of n increases, we see an almost exponential decrease in p_n, as longer matching n-gram sequences are more difficult to find. In order to make BLEU more sensitive to longer n-grams, a weighted average is calculated by summing over the logarithm of each p_n for a range of values of n,[20] and multiplying by a uniform weight $\frac{1}{N}$. This equation is given in (2.19):

$$p_N = \exp(\sum_{n=1}^{n=1} \frac{1}{N} \log(p_n)) \tag{2.19}$$

Candidate translations that are longer than their reference(s) are implicitly penalised when calculating p_n. In order to compensate for this, a corresponding *brevity penalty* BP is imposed which penalises those candidate translations shorter than their reference(s). The final BLEU score is calculated as the product of p_N and BP.

[20]While scores can be obtained for any value of n, Papineni et al. (2002) found that considering a maximum value for n of 4 was sufficient for adequate correlation with human judgements.

Papineni et al. (2001) state that the BP is a decaying exponential in the length of the reference sentence over the length of the candidate translation. This effectively means that if the candidate translation is the same length (or longer) than the reference, then BP is 1, and BP is greater than 1 if the candidate is shorter than the reference. Thus, BP is calculated according to equation (2.20):

$$BP = e^{max(1-\frac{length(R)}{length(C)},0)} \tag{2.20}$$

In order to avoid punishing shorter candidates too harshly, BP is calculated over the entire corpus rather than on a sentence-by-sentence basis and taking the average. That is, in equation (2.20), $length(R)$ refers to the total number of words in the reference set and $length(C)$ refers to the total number of words in the candidate set. The penalty is then applied to the modified precision score, to give a single score for the entire candidate translation set, according to the equation in (2.21):

$$BLEU = BP \cdot p_N \tag{2.21}$$

All BLEU score calculations in this thesis were made using BLEU as implemented in the mteval-v11b.pl script,[21] released as part of the annual NIST Open MT evaluation campaign.[22]

2.4.2 NIST

The **NIST** metric (Doddington, 2002) is a variation on the BLEU metric which makes three specific alterations to the way in which scores are calculated. The first change addresses the issue of n-gram informativeness; when calculating the modified n-gram precision, BLEU assigns equal weights to each n-gram. NIST, on the other hand, assigns more weight to co-occurring n-grams that occur less frequently in the reference corpus. The intuition here is that finding a co-occurring n-gram pair in

[21] Downloaded from ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl
[22] http://www.itl.nist.gov/iad/mig/tests/mt/

the candidate and reference translations that occurs frequently is not as indicative of the quality of translation as finding a rare co-occurring n-gram pair. Information weights are calculated over the n-gram counts in the reference sets according to the equation in (2.22):

$$Info(w_1 \dots w_n) = log_2(\frac{count(w_1 \dots w_{n-1})}{count(w_1 \dots w_n)}) \tag{2.22}$$

This is then incorporated into the modified n-gram precision formula in (2.18) as shown in (2.23):

$$p_n = \frac{\sum w_1 \dots w_n \in |c_n \cap r_n| Info(w_1 \dots w_n)}{|c_n|} \tag{2.23}$$

The second change deals with the way the precision scores for all values of n are combined into a single score p_N. BLEU sums over the logarithm of each value of p_n and multiplies by a weight $\frac{1}{N}$ in order to make p_N more sensitive to larger values of n. However, Doddington (2002) points out that this method of scoring is equally as sensitive to varying co-occurrence frequencies regardless of the value of n. In order to overcome this, Doddington (2002) simply takes the arithmetic average of the values of p_n as shown in equation (2.24):

$$p_N = \sum_{n-1}^{N} \frac{\sum w_1 \dots w_n \in |c_n \cap r_n| Info(w_1 \dots w_n)}{|c_n|} \tag{2.24}$$

The final change involves altering how the brevity penalty is calculated. In BLEU, BP is particularly sensitive to any variation in translation length. NIST changes the calculation in order to minimise changes in scores given small variations in length. This is done by introducing a value β, which is chosen such that BP is 0.5 when the number of words in all candidate translations C is $\frac{2}{3}$ the average length of the number of words in all references R. Thus, NIST is calculated according to the equation in (2.25):

$$BP = exp\left(\beta \cdot log_2[min(\frac{length(R)}{length(C)}), 1]\right) \tag{2.25}$$

As with BLEU, all NIST score calculations in this thesis were made using NIST as implemented in the mteval-v11b.pl script.

2.4.3 METEOR

The **METEOR** (Banerjee and Lavie, 2005; Lavie and Agarwal, 2007) metric evaluates MT output by placing high emphasis on the recall of the candidate translation given the reference. The authors motivate this by pointing out that recall "reflects to what degree the translation covers the entire context of the translated sentence [reference]". METEOR computes a score for candidate translations using a combination of unigram-precision, unigram-recall and a measure of fragmentation given the candidate sentence, reference sentence and a set of generalised unigrams between the two. This method is designed to overcome potential issues with the BLEU and NIST metrics such as the lack of recall, the use of higher-order n-grams to evaluate grammaticality (or word order), and scores being calculated over the entire testset as opposed to sentence-level.

Given a candidate translation and a reference, METEOR first creates an alignment between the two strings such that every unigram in one string maps to zero or one unigrams in the other string. This alignment is performed incrementally in a series of stages, with each stage comprising two phases.

The first phase creates all possible alignments between the two strings. Alignments can be created based on three criteria:

(i) *exact matches* where the two unigrams are identical (e.g. "parliament" maps to "parliament" but not to "parliamentary");

(ii) *stems* where the unigrams are identical after they are stemmed using the Porter stemmer (Porter, 1980) (e.g. "parliament" maps to both "parliament" and "parliamentary");

(iii) *synonyms* where two unigrams are mapped if they are synonymous according to WordNet (Miller, 1995).

The second phase selects the largest subset of these alignments that are well-formed as the final mapping. If there is more than one well-formed subset, METEOR selects the set with the least number of crossing alignments, i.e. that set in which the word order in the candidate is most similar to the reference.

Once a final alignment has been chosen, METEOR first calculates unigram-precision P and unigram-recall R of the candidate translation, as shown in equations (2.26) and (2.27) respectively:

$$P = \frac{a}{u_c} \tag{2.26}$$

$$R = \frac{a}{u_r} \tag{2.27}$$

where

- a is the number of candidate unigrams aligned to reference unigrams.
- u_c is the total number of unigrams in the candidate translation.
- u_r is the total number of unigrams in the reference translation.

METEOR then calculates the harmonic mean F_{mean} of P and R placing most of the weight on recall[23] using the formula in (2.28):

$$F_{mean} = \frac{(1+\alpha) \cdot PR}{R + \alpha P} \tag{2.28}$$

F_{mean} is calculated based solely on unigram matches. To reward longer matches, and provide a direct alternative to averaging over values of p_n as is done in BLEU and NIST, METEOR calculates a *penalty* based on the number of consecutive unigram alignments (n-grams) between the sentences, or *chunks* (ch). The longer the n-gram matches, the fewer chunks there are and consequently the lower the penalty. In one extreme case, the entire candidate string matches the entire reference and there is

[23]Lavie and Agarwal (2007) set α to 9.0 based on previous experimentation, while alternative values have also been suggested, cf. (He and Way, 2009).

one single chunk. In the other extreme, there are no bigram or longer matches so the number of chunks is equal to the number of unigram alignments. The penalty is calculated according to the equation in (2.29):

$$Penalty = \gamma(\frac{ch}{a})^{\beta} \tag{2.29}$$

where

- γ determines the maximum penalty possible ($0 \leq \gamma \leq 1$).[24]
- β determines the functional relation between fragmentation and the penalty.[25]
- U_r is the total number of unigrams in the reference translation

Thus, the final METEOR score is calculated according to (2.30):

$$METEOR = F_{mean} \cdot (1 - Penalty) \tag{2.30}$$

All METEOR scores presented in this thesis were calculated using METEOR version 0.5.1.[26]

2.4.4 Drawbacks of Automatic Evaluation

In recent years, there has been considerable focus in the MT community on the perceived inadequacy of automatic evaluation metrics when it comes to accurately reflecting human judgements of translation quality (Zhang et al., 2004; Callison-Burch et al., 2006; Chiang et al., 2008; Owczarzak, 2008). There are many instances in which the n-gram-based metrics will score translations poorly despite them being perfectly acceptable. For example, in (2.31) the translation will receive a low score according to the metrics presented previously, despite being adequate output, as it has only two of three unigram matches with the reference and no higher order n-gram matches.

[24] γ is set to 0.5 by default in the literature.
[25] β is set to 3.0 by default in the literature.
[26] Downloaded from http://www.cs.cmu.edu/∼alavie/METEOR/meteor-0.5.1.tar.gz .

Translation John quit yesterday
Reference Yesterday John resigned (2.31)

This may not be surprising to the developers of these metrics and researchers working in the area of MT evaluation, as one the earliest of the evaluation metrics, BLEU, was not intended to be a substitute for human assessment of translation, but rather as an "understudy" to human evaluators (Papineni et al., 2002). Additionally, n-gram-based metrics have been shown to favour the output of SMT systems over that of rule- and syntax-based ones (Callison-Burch et al., 2006).

For these reasons, particularly in Chapter 5, we endeavour to supplement the automatic evaluation of our MT output with manual analysis in this thesis in order to provide a clearer view of the relative merits and drawbacks of our methods.

2.4.5 Statistical Significance

Where stated, statistical significance was carried out on the results in this thesis, for the BLEU and NIST metrics,[27] using bootstrap resampling (Koehn, 2004). A confidence value of p=0.05 was used (unless otherwise stated) with 1,000 resampled test sets. If no explicit mention to statistical significance testing is made, the results are statistically significant.

2.5 Summary

In this chapter, we have provided a general description of the concept of parallel treebanks as well as our motivation for developing a new algorithm for the automatic induction of sub-sentential alignments between parallel tree pairs. We described the main components comprising a phrase-based statistical MT system, particularly the phrase extraction process and various features of the log-linear model, demonstrating

[27]The software we used to calculate statistical significance — downloaded from http://projectile.is.cs.cmu.edu/research/public/tools/bootStrap/tutorial.htm — did not facilitate testing with the METEOR metric.

the lack of linguistic motivation throughout which was our stimulus for investigating the exploitability of parallel treebanks in this paradigm. Following this, we introduced syntax-based MT and provided detailed descriptions of the systems in which we employed our parallel treebanks as training data, as well as providing a summary of alternative approaches. Finally, we described the various automatic measures used to evaluate the quality of MT output in the various experiments presented in this thesis.

In the next chapter, we address the first research question (**RQ1**) posed in Chapter 1 by describing the development of a sub-tree alignment tool for the automatic generation of parallel treebanks.

Chapter 3

Sub-Tree Alignment: development and evaluation

In the previous chapter, we described the current state-of-the-art in PB-SMT and the field of parallel treebanking. We noted in our discussion that there existed no adequate means by which we can automatically generate parallel treebanks that suited our requirements, thus providing the rationale for the development of such a technique. In this chapter, we document the novel sub-tree alignment algorithm (Hearne et al., 2007; Tinsley et al., 2007b; Zhechev, 2009) we have developed in terms of design and performance. The design reflects our motivation to develop an efficient tool for the automatic generation of parallel treebanks that is language pair- and task-independent and whose output may be useful in a variety of natural language applications. The alignment algorithm induces links between the nodes of paired linguistic structures which indicate translational equivalence between the surface strings dominated by the linked node pairs. Accordingly, in sections 3.1.1 and 3.1.2 we outline our design principles and criteria for ensuring well-formed alignments. The main alignment algorithm constitutes the core of this body of work and is detailed in section 3.2 along with a series of variations and extensions. We then carry out a systematic evaluation of the automatically induced alignments produced using our algorithm. Firstly, the quality of the alignments is assessed against a set of man-

ually annotated gold standard alignments. We then perform a task-based evaluation by employing parallel treebanks created with the aid of the alignment algorithm as training data for a Data-Oriented Translation (DOT) system (Hearne, 2005). Finally, we manually evaluate the alignments in terms of their ability to capture some predefined translational divergences between the language pair in question. These evaluations are presented in section 3.3 and discussed further in section 3.3.5.

3.1 Prerequisites

In this section, we present a set of prerequisites we considered when developing our alignment algorithm. We describe some guiding principles and our motivation behind them in section 3.1.1, while in section 3.1.2 we define the criteria to which alignments must conform in order to be considered well-formed.

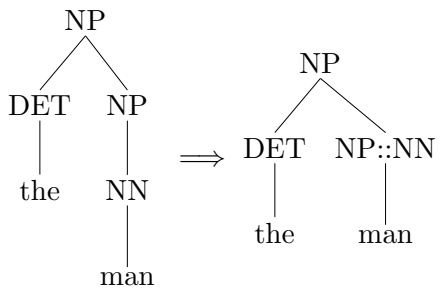
3.1.1 Alignment Principles

The novel algorithm we present in this chapter is designed to discover an optimal set of alignments between a pair of parallel trees while adhering to the following principles:

1. independence with respect to language pair and constituent labelling schema;
2. preservation of the given tree structures;
3. minimal external resources required;
4. word-level alignments not fixed *a priori*.

The algorithm we will present makes use of a single external resource, namely source-to-target and target-to-source word translation probabilities generated by performing statistical word alignment on the sentence pairs encoded in the parallel treebank. The algorithm does not, however, fix *a priori* on any proposed word alignment at this juncture. Rather, these word translation probabilities are used to

calculate scores for possible node alignments as is described fully in section 3.2.4. The alignment algorithm does not edit or transform the trees; as we discussed in section 2.1.2, significant structural and translational divergences are to be expected and the aligned tree pair should encode these divergences. This may not be the most appropriate approach for certain tasks, such as phrase-extraction for MT, as restricting the space of extractable phrases to those corresponding to linked nodes between tree pairs leads to sparseness issues as has been demonstrated by Koehn et al. (2003) and Ambati and Lavie (2008) amongst others. However, as we wish to retain the linguistic integrity of the trees and develop a task-independent algorithm, we preserve the given tree structures. We demonstrate in later chapters that the resulting parallel treebanks can still be beneficial for the translation process. However, there is one instance in which trees are altered from their original structure. This occurs when unary productions are collapsed into a single node. As links are induced based on surface strings dominated by constituent nodes (as opposed to the tree structures), unary productions would introduce redundancy into the alignment process as there would be more than one node representing the same sub-string in the tree. We resolve this by collapsing unary productions into a single node, as illustrated in (3.1), packing sufficient information into the node such that it can be expanded to the original structure based on the requirements of any end task. Finally, the algorithm accesses no language-specific information beyond the (automatically induced) word-alignment probabilities and does not make use of the node labels in the parse trees, so the labelling schema is irrelevant.



(3.1)

3.1.2 Alignment Well-Formedness Criteria

Links are induced between tree pairs such that they meet the following well-formedness criteria:

1. a node can only be linked once;
2. descendants of a source linked node may only link to descendants of its target linked counterpart;
3. ancestors of a source linked node may only link to ancestors of its target linked counterpart.

These criteria are in place in order to ensure the translational equivalence implications of a link, as discussed in section 2.1. For example, the first criterion states that a node can only be linked once. If we were to have two links coming from a particular source node it would imply that the string dominated by this node is translationally equivalent to two distinct sub-phrases in the target sentence, and this is not desirable. This is illustrated in Figure 3.1(a). Given the existing dashed link between nodes `A` and `W`, the solid link from `C` to `W` is now illegal. Figure 3.1(b) illustrates violations of the second and third constraints. Given the dashed link between nodes `C` and `W`, descendants of these two nodes may only link to one another; that is, nodes `D` and `E` on the left tree may only link to nodes `Y` and `Z` on the right tree. Thus, the solid link between `E` and `V` is illegal. This link is also ill-formed in that node `V` is an ancestor of linked node `W` and thus can only aligned to ancestors of `W`'s linked correspondant `C`, which in this case is only node `A`. The criteria are akin to the "crossing constraints" described in (Wu, 1997) which forbid alignments that cross each other. Our criteria differ from those of Wu because we impose them on a pair of fully monolingually parsed trees, so our criteria are more strict. The constraints in (Wu, 1997), on the other hand, are imposed inherently during the bilingual parsing and alignment phase.

In what follows, a hypothesised alignment is ill-formed with respect to all existing alignments if it violates any of these criteria.

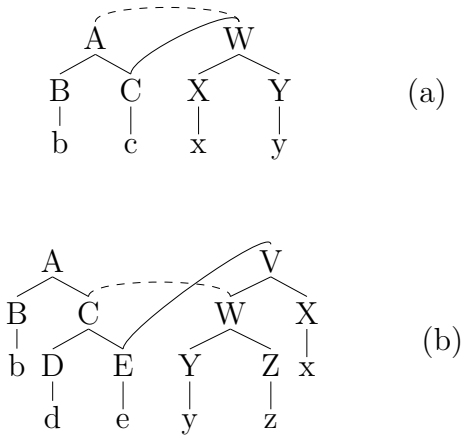


Figure 3.1: Examples of ill-formed links given the well-formedness criteria.

3.2 Algorithm

In this section, we present a precise description of our alignment algorithm, originally introduced in Tinsley et al. (2007b), in terms of hypothesis initialisation, hypothesis selection and hypothesis scoring. We introduce the basic algorithm in section 3.2.1 by describing how we initialise the process, and select between all hypothetical alignment options. Following this, we discuss a number of extensions and alterations to the basic algorithm, motivated by various considerations, in sections 3.2.2 and 3.2.3. Finally in section 3.2.4, we describe how we use word alignment probabilities to calculate scores for our alignment hypotheses.

3.2.1 Basic Configuration

For a given tree pair $\langle S, T \rangle$, the alignment process is initialised by proposing all links $\langle s, t \rangle$ between nodes in S and T as hypotheses and assigning scores $\gamma(\langle s, t \rangle)$ to them. All zero-scored hypotheses are blocked before the algorithm proceeds. The selection procedure then performs a greedy search by iteratively fixing on the highest-scoring link, blocking all hypotheses that contradict this link and the link itself, until no non-blocked hypotheses remain. These initialisation and selection procedures are given in **Algorithm 1** *basic*.

Algorithm 1 *basic*

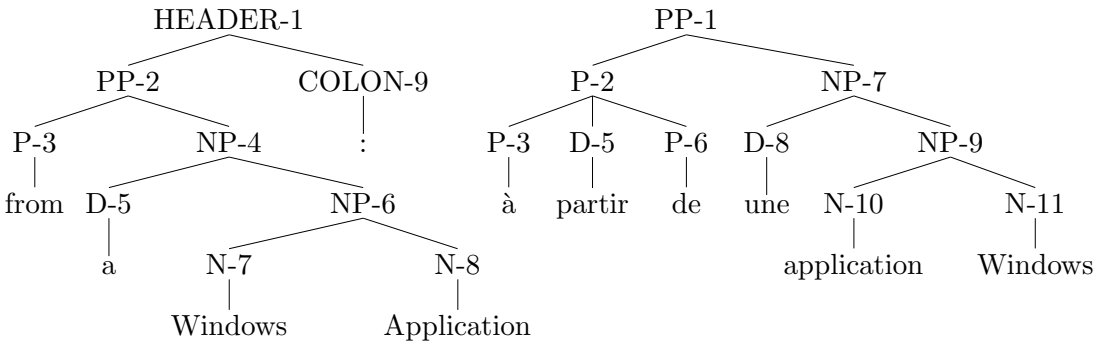
Initialisation

for each source non-terminal s **do**
  for each target non-terminal t **do**
    generate scored hypothesis $\gamma(\langle s, t \rangle)$
  end for
end for
block all zero-scored hypotheses

Selection *underspecified*

while non-blocked hypotheses remain **do**
  link and block the highest-scoring hypothesis
  block all contradicting hypotheses
end while

Figure 3.2 illustrates the **Algorithm 1 Selection** *basic* procedure. The constituents in the source and target tree pair are numbered. The numbers down the left margin of the grid correspond to the source constituents while the numbers across the top correspond to the target constituents, and each cell in the grid corresponds to a scored hypothesis. Within each cell, circles denote selected links and brackets denote blocked links. The number inside a given cell indicates the iteration during which its link/block decision was made, with 0s indicating hypotheses with score zero. For example, hypothesis $\langle 1, 1 \rangle$ (i.e. nodes `HEADER-1` and `PP-1` in the English and French trees respectively) was linked during iteration 1, and hypothesis $\langle 2, 1 \rangle$ was blocked, hypothesis $\langle 5, 8 \rangle$ was linked during iteration 2 and hypotheses $\langle 5, 6 \rangle$, $\langle 6, 7 \rangle$ and $\langle 9, 8 \rangle$ were blocked, and so on. There were 7 iterations in total, and the last iteration linked the remaining non-zero hypothesis $\langle 7, 11 \rangle$. As reported in Zhechev (2009), the complexity of the basic algorithm is quadratic in the number of source and target language tokens.



	1	2	3	5	6	7	8	9	10	11
1	**1**	0	0	0	0	0	0	0	0	0
2	**(1)**	0	0	0	0	0	0	0	0	0
3	0	**3**	0	0	0	0	0	0	0	0
4	0	0	0	0	0	**6**	0	0	0	0
5	0	0	0	0	**(2)**	0	**2**	0	0	0
6	0	0	0	0	0	**(2)**	0	**5**	**(4)**	0
7	0	0	0	0	**(3)**	0	0	0	0	**7**
8	0	0	0	0	0	0	0	0	**4**	0
9	0	0	0	0	**(3)**	0	**(2)**	0	0	**(5)**

Figure 3.2: Illustration of how **Algorithm 1 Selection** *basic* induces links for the tree-pair on the left.

3.2.2 Resolving Competing Hypotheses (skip)

The **Selection** procedure given in **Algorithm 1 Selection** *basic* is incomplete as it does not specify how to proceed if two or more hypotheses share the same highest score. We propose two alternative solutions to this problem. Firstly, we can simply skip over tied hypotheses until we find the highest-scoring hypothesis with no competitors of the same score, as given by **Algorithm 2 Selection** *skip1*.

The skipped hypotheses will, of course, still be available during the next iteration, assuming that they have not been ruled out by the newly selected link. If all but one of the tied hypotheses have been ruled out, the remaining one will be selected on

Algorithm 2 Selection *skip1*

```
while at least one non-blocked hypothesis with no tied competitors remains do
  while the highest-scoring hypothesis has tied competitors do
    skip
  end while
  link and block the highest-scoring non-skipped hypothesis
  block all contradicting hypotheses
  re-enable all non-blocked skipped hypotheses
end while
```

the next iteration. If all remaining non-zero-scored hypothesis have tied competitors then no further links can be induced.

A second alternative is to skip over tied hypotheses until we find the highest-scoring hypothesis $\langle s, t \rangle$ with no competitors of the same score *and where neither s nor t has been skipped*, as given in **Algorithm 3 Selection** *skip2*.

Algorithm 3 Selection *skip2*

```
while at least one non-blocked hypothesis with no tied competitors remains do
  if the highest-scoring hypothesis has tied competitors then
    mark the constituents of all competitors as skipped
  end if
  while the highest-scoring hypothesis has a skipped constituent do
    skip
  end while
  link and block highest-scoring not-skipped hypothesis
  block all contradicting hypotheses
  re-enable all non-blocked skipped hypotheses
end while
```

This alternative is proposed in order to avoid the situation in which a low-scoring hypothesis for a given constituent is selected in the same iteration as higher-scoring hypotheses for the same constituent were skipped, thereby preventing one of the higher-scoring competing hypotheses from being selected and resulting in an undesired link. The issue is illustrated in Figure 3.3. The best-scoring hypotheses, of which there are several, involve source constituent D-21 and include the correct hypothesis ⟨D-21, D-16⟩. The *skip1* solution simply selects the best non-tied hypothesis, ⟨D-21, D-4⟩, which is clearly incorrect. The *skip2* solution, however, skips over all hypotheses involving skipped constituent D-21 and selects ⟨D-16, D-4⟩ as the best hypothesis. On the next iteration, all hypotheses for source constituent D-21 are

again skipped, and hypothesis ⟨PP-18, PP-13⟩ is selected. This selection blocks all but one hypothesis involving source constituent D-21, the correct hypothesis ⟨D-21, D-16⟩, and so this link is selected on the following iteration.

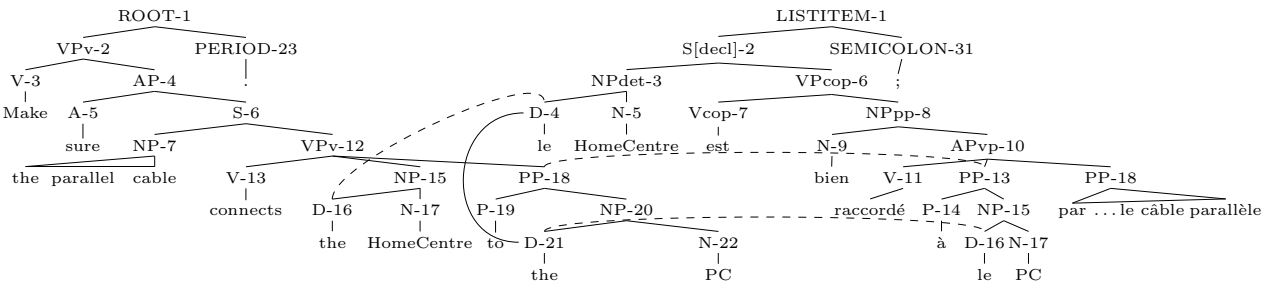


Figure 3.3: This example illustrates the differing effects of the **Selection** *skip1* and **Selection** *skip2* strategies: with *skip1* the undesirable solid link is induced whereas with *skip2* the correct dashed links are induced.

3.2.3 Delaying Lexical Alignments (span)

It is frequently the case that the highest-scoring hypotheses are at the word level, i.e. a node has a span of 1 on the source and/or target side. However, selecting links between frequently occurring lexical items at an early stage is intuitively unappealing. Consider, for instance, the situation where source terminal x most likely translates as target terminal y but there is more than one occurrence of both x and y in a single sentence pair. It may be better to postpone the decision as to which instance of x corresponds to which instance of y until links higher up in the tree pair have been established, as given in **Algorithm 4 Selection** *span1* (where span-1 hypotheses have span 1 on the source and/or target sides and non-span-1 refers to all other hypotheses).

The effects of the **Selection** *span1* strategy are illustrated by the example given in Figure 3.4: without *span1*, the English node ⟨D-8 *the*⟩ is immediately linked to the French node ⟨D-13 *le*⟩ rather than being correctly linked to the node ⟨D-4 *le*⟩ and also the English node ⟨D-17 *the*⟩ is linked to the French node ⟨D-4 *le*⟩ rather than ⟨D-13 *le*⟩. Not only are these alignments incorrect, but their presence means that

Algorithm 4 Selection *span1*

```
while non-blocked non-lexical hypotheses remain do
  link and block the highest-scoring hypothesis
  block all contradicting hypotheses
  if no non-blocked non-lexical hypotheses remain then
    while non-blocked lexical hypotheses remain do
      link and block the highest-scoring hypothesis
      block all contradicting hypotheses
    end while
  end if
end while
```

the remaining desirable hypotheses are no longer well-formed. However, the correct alignments are induced by first allowing the English node ⟨NP-7 *the scanner*⟩ to link to the French node ⟨NP-3 *le scanner*⟩ and ⟨NP-16 *the HomeCentre*⟩ to ⟨NP-12 *le HomeCentre*⟩, which is the case when *span1* is applied.

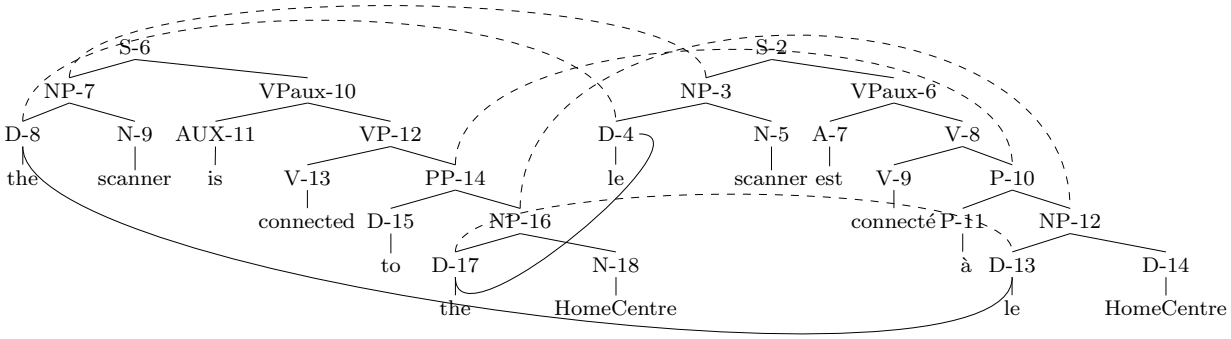


Figure 3.4: This example illustrates the effects of the Selection *span1* strategy: without *span1* the solid links are induced whereas switching on *span1* results in the dashed alignments.

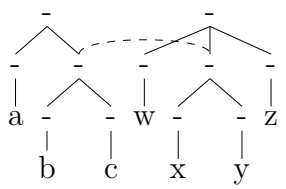
3.2.4 Calculating Hypothesis Scores

We will now describe the process by which we assign scores to the hypothesised links. Inserting a link between two nodes in a tree pair indicates that (i) the substrings dominated by those nodes are translationally equivalent and (ii) all meaning carried by the remainder of the source sentence is encapsulated in the remainder of the target sentence. The scoring method we propose accounts for these indications.

Given a tree pair $\langle S, T \rangle$ and hypothesis $\langle s, t \rangle$, we compute the following strings:

$$s_l = s_i ... s_{ix} \quad \overline{s_l} = S_1 ... s_{i-1} s_{ix+1} ... S_m$$
$$t_l = t_j ... t_{jx} \quad \overline{t_l} = T_1 ... t_{j-1} t_{jx+1} ... T_n$$

where $s_i ... s_{ix}$ and $t_j ... t_{jx}$ denote the terminal sequences dominated by s and t respectively, and $S_1 ... S_m$ and $T_1 ... T_n$ denote the terminal sequences dominated by S and T respectively. These string computations are illustrated in Figure 3.5.



s_l = b c
t_l = x y
$\overline{s_l}$ = a
$\overline{t_l}$ = w z

Figure 3.5: Values for s_l, t_l, $\overline{s_l}$ and $\overline{t_l}$ given a tree pair and a link hypothesis.

The score for the given hypothesis $\langle s, t \rangle$ is computed according to (3.2).

$$\gamma(\langle s, t \rangle) = \alpha(s_l|t_l)\ \alpha(t_l|s_l)\ \alpha(\overline{s_l}|\overline{t_l})\ \alpha(\overline{t_l}|\overline{s_l}) \tag{3.2}$$

Individual string-correspondence scores $\alpha(x|y)$ are computed using word translation probabilities retrieved using a statistical word aligner.[1] Two alternative scoring functions are given by *score1* (3.3) and *score2* (3.4), which are loosely based on IBM Model 1 for word alignment as described in Brown et al. (1990). In *score1*, for a given source word x_j we sum over the probabilities of it translating as each target word $y_1 \ldots y_i$. This gives us the probability of the target string corresponding to each source word. We take the product of these probabilities for each source word to obtain a correspondence score for the entire string pair.

The alternative approach presented in *score2* sums over the probability of each source word $x_1 \ldots x_j$ translating as a given target word y_i. We then take the average score, dividing by the number of words in the target string (i). Following this, we again take the product of these scores for each target word to give us a correspondence score for the entire string pair. The intended effect of the *score2* function, as

[1] We use GIZA++ to calculate word translation scores throughout this thesis (cf. Section 2.2.1).

with *span1*, is to reduce any bias in favour of aligning shorter span constituents over longer ones.

Score *score1*

$$\alpha(x|y) = \prod_{i=1}^{|x|} \sum_{j=1}^{|y|} P(x_i|y_j) \tag{3.3}$$

Score *score2*

$$\alpha(x|y) = \prod_{i=1}^{|x|} \frac{\sum_{j=1}^{|y|} P(x_i|y_j)}{|y|} \tag{3.4}$$

Similar to the lexical weighting feature described in Section 2.2.3, to account for cases in which source and target words have no correspondents according to the word translation probability distribution, we add a special NULL word to the target string. In the distribution estimated using GIZA++, probabilities are calculated for words translating to NULL, but this is not so for all words. In cases where no probability is present for a word translating as NULL, it receives a score of zero. If, for a given hypothesis, a source-side word has no correspondents on the target side according to the word translation distribution, we can safely assume the overall hypothesis is poor. In this case, the sum over this word will be zero and consequently the product will also amount to zero for the hypothesis and thus we have the desirable effect of omitting this hypothesis from the selection process.

3.3 Aligner Evaluation

In section 3.3.1, we describe the dataset we used and the basic experimental set-up for all experiments. Section 3.3.2 details experiments we carried out in terms of evaluating the alignment quality against gold-standard human alignments. We then perform a task-based evaluation of the alignments as described in section 3.3.3, and finally in section 3.3.4 we manually investigate the quality of the alignments in terms of a number of translational divergences.

3.3.1 Data

The experiments in sections 3.3.2 and 3.3.3 evaluate all possible configurations of the aligner. When configuring the alignment algorithm, we must choose either *skip1* or *skip2* and we must choose either *score1* or *score2*. Using *span1* is optional, so it can be switched on or off. This gives us eight possible configurations of the algorithm, as shown in Figure 3.6:

skip1_score1	skip1_score1_span1
skip1_score2	skip1_score2_span1
skip2_score1	skip2_score1_span1
skip2_score2	skip2_score2_span1

Figure 3.6: The 8 possible configurations of the alignment algorithm.

The corpus we use for all evaluations is the English–French section of the HomeCentre corpus, which contains 810 parsed, sentence-aligned translation pairs.[2] This corpus comprises a Xerox printer manual, which was translated by professional translators and sentence-aligned and annotated at Xerox PARC. As one would expect, the translations it contains are of extremely high quality.

We produced a set of automatic alignments for each configuration of the aligner.[3] Word alignment probabilities, used to calculate the hypothesis scores for the aligner, were obtained by running GIZA++ (Och and Ney, 2003) on the 810 sentence pairs in the corpus. The manual alignments were provided by a single annotator, who is a native English speaker with proficiency in French (Hearne, 2005).

3.3.2 Intrinsic Evaluation

In this section, we evaluate the precision and recall of induced alignments over the 810 English–French tree pairs described previously, using the manually aligned version as a gold standard and discuss the results.

[2]The average numbers of English and French words per sentence are 8.83 and 10.05 respectively, and the average numbers of English and French nodes per tree are 15.33 and 17.52 respectively.

[3]Alignment took approximately 0.004 seconds per tree on an Apple machine with a 2.33GHz dual-core processor and 2GB of RAM; time variations of aligner configurations are insignificant.

	all links			*non-lexical links*		
Configurations	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
skip1_score1	0.6096	0.7723	0.6814	**0.8424**	0.7394	0.7875
skip1_score2	0.6192	0.7869	0.6931	0.8107	0.7756	0.7928
skip2_score1	0.6162	0.7783	0.6878	0.8394	0.7486	0.7914
skip2_score2	0.6215	0.7867	0.6944	0.8107	0.7756	0.7928
skip1_score1_span1	0.6229	**0.8101**	0.7043	0.8137	0.7998	0.8067
skip1_score2_span1	0.6220	0.7963	0.6984	0.8027	0.7871	0.7948
skip2_score1_span1	**0.6256**	0.8100	**0.7060**	0.8139	**0.8002**	**0.8070**
skip2_score2_span1	0.6245	0.7962	0.7001	0.8031	0.7871	0.7950

Table 3.1: Evaluation of the automatic alignments against the manual alignments.

Evaluation Metrics

Given a tree pair T, its automatically aligned version T_A and its manually aligned version T_M, we calculate precision according to the equation in (3.5). The precision rate of a set of alignments is the proportion of automatic alignments that correspond to manual alignments in the gold standard.

$$Precision = \frac{|T_A \cap T_M|}{|T_A|} \tag{3.5}$$

Recall is calculated according to equation (3.6), where the recall rate of a set of alignments is the proportion of total number of automatic alignments corresponding to a manual alignment with respect to total number of manual alignments.

$$Recall = \frac{|T_A \cap T_M|}{|T_M|} \tag{3.6}$$

In addition to calculating precision and recall over all links, we also calculate scores of non-lexical links only, where a non-lexical link aligns constituents which both span more than one word. The motivation behind this is to allow us to determine how successful the algorithm is at inducing alignments above the word level.

Results

The results shown in Table 3.1 give precision and recall scores for all eight algorithm configurations against the gold standard for both the entire set of links and non-

lexical links only. Looking firstly to the *all links* column, it is immediately apparent that recall is significantly higher that precision for all configurations. We note that all aligner configurations consistently induce more links than exist in the manually aligned treebank, with the average number of links per tree pair ranging between 10.3 and 11 for the automatic alignments versus 8.3 links per tree pair for the manual version. Regarding the differences in performance between the aligner variants, we observe that all configurations which include *span1* outperform all configurations which exclude it.

Looking now at the *non-lexical links* column, we observe that the balance between precision and recall is reversed and that precision is now higher than recall in all cases. This indicates that those phrase-level alignments we induced were reasonably accurate and conversely suggests that the accuracy of our lexical-level alignments were relatively poor. Regarding the differences in performance between the aligner variants, we note that both the highest precision and lowest recall were achieved using *skip1_score1* and *skip2_score1*. However, the best balance between precision and recall is again achieved when the *span1* option is used. This is due to the fact that *span1* allows for increased recall by omitting instances in which poor lexical choice limits the number of available hypotheses, and subsequently recall. The remaining decisions on word alignments are then easier to make and chances of increased precision are improved.

3.3.3 Extrinsic Evaluation

In this section, we carry out a task-based evaluation of the automatic alignments. We use the manually aligned parallel treebank to train a DOT system (Hearne, 2005). We assess translation performance using a number of established metrics for automatic MT evaluation, described in section 2.4, to give us a baseline. We then use the automatically aligned parallel treebanks produced by the 8 configurations of the alignment algorithm to train a number of DOT systems and evaluate performance such that the only difference across MT system configurations is the sub-sentential

alignments in the parallel treebank.

Experimental Setup

We used 9 versions of the HomeCentre parallel treebank to train DOT systems: one aligned manually as described in Section 3.3.1, and the others using the 8 aligner configurations specified in the same section. In order to make full use of our limited training resources, we generated 6 training/test splits for the HomeCentre data such that (i) all test words also appeared in the training set, (ii) all splits have English as the source language and French as the target language and (iii) each test set contains 80 test sentences and each training set contains 730 tree pairs. We then applied the 6 splits to each of the 9 versions of the dataset, trained the MT system on each training set and performed translation on each corresponding test set. Final evaluation scores are presented as the average over the 6 splits.

For the MT experiments presented in this chapter and all subsequent chapters, we evaluate translation performance using three automatic metrics described in section 2.4: BLEU, NIST and METEOR. Statistical significance testing was not carried out for the experiments in this chapter due to the relatively small size of our test set and the nature of our evaluation framework. Finally, an additional measure we use to extrinsically evaluate the automatic alignments in this section is the translation coverage achieved by the DOT system.[4]

DOT Coverage Measure

Recalling how the DOT system works from section 2.3.2, target language translations are built synchronously as the source input is parsed by the DOT grammar. In some cases, a full target-side parse tree cannot be built and some heuristics are applied to piece the tree fragments together. In cases where a full target-side parse is built, that sentence is said to have full coverage. Thus, when we calculate DOT coverage we are measuring the percentage of translations that received a full target-side parse.

[4]This measure is only applicable in this section and is not used for evaluation in subsequent chapters.

Obviously, the better the alignment quality, the better the extracted grammar and consequently more target trees receive a full parse and thus the higher the DOT coverage.

Results

Configurations	**BLEU**	**NIST**	**METEOR**	Coverage
manual	0.5345	**6.9590**	0.7274	70.4167%
skip1_score1	0.5155	6.8706	0.7217	74.4792%
skip1_score2	0.5342	6.9008	0.7300	75.2084%
skip2_score1	0.5167	6.8893	0.7256	74.5834%
skip2_score2	**0.5346**	6.9007	0.7309	75.2084%
skip1_score1_span1	0.5256	6.8751	0.7280	**75.4167%**
skip1_score2_span1	0.5337	6.9198	**0.7314**	74.7917%
skip2_score1_span1	0.5257	6.8893	0.7295	**75.4167%**
skip2_score2_span1	0.5336	6.9201	0.7305	74.7917%

Table 3.2: Translation scores for DOT systems trained using various alignment configurations.

Table 3.2 presents the translation scores for the 9 DOT systems we trained using different parallel treebanks. Firstly, comparing the automatically derived treebanks to the manual alignments, we see that the majority of the automatic configurations lead to comparable or improved translation performance. We also see that translation coverage improves by up to 7.1% absolute improvement (5% relative improvement) when using automatic alignments.

Comparing the automatically generated parallel treebanks to one another, no one particular configuration consistently outperformed the others. However, we do obtain some insight as to the relative performance of the various alignment configurations. When we observe *score2* and *span1* in isolation,[5] they consistently lead to improvements across all metrics. For instance, when *score2* is used instead of *score1* we see improvements, e.g. *skip1_score2* > *skip1_score1* in Table 3.2, and when

[5] *score2* and *span1* were effectively introduced to remedy the same problem: high-scoring, low-quality lexical alignments. When observed in isolation they consistently lead to improvements. However, when applied together e.g. *skip1_score2_span1*, the results produced are erratic. We attribute this behaviour to an apparent conflict between the two options. We leave investigations into the cause of this conflict for further research.

the *span1* option is turned on we again see improvements, e.g. *skip1_score1_span1* > *skip1_score_1* in Table 3.2. Furthermore, configurations in which *skip2* is employed in place of *skip1* tend to have higher translation accuracy and coverage, e.g. *skip2_score1_span1* > *skip1_score1_span1* in Table 3.2. We attribute these improvements in DOT accuracy to the better hypothesis selections being made given the more intuitive selection processes applied when using these configurations. In the following section, we present examples of aligned parallel trees produced by the best-performing configurations from this evaluation.

3.3.4 Manual Evaluation

In this section, we present an evaluation of our alignment algorithm in which we manually observed the quality of the sub-tree alignments in terms of the extent to which they captured certain translational divergences between the languages in the parallel treebank (Hearne et al., 2007). The phenomena we evaluated against were all to be found in our HomeCentre data set which, as noted by (Frank, 1999), provides a rich source of both linguistic and translational complexity. The specific phenomena we observed were:

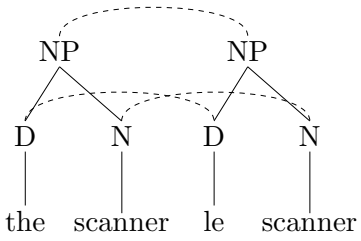
- nominalisation;
- stylistic divergence;
- head-switching;
- lexical divergence.

For the purposes of this evaluation, we used two configurations of the aligner: *skip2_score1_span1* and *skip2_score2*. This choice was based on the evaluations of the previous two sections in which we found *skip2* to outperform *skip1*, and *span1* and *score2* to perform best when not used in the same configuration. The evaluation carried out here is admittedly not as systematic as it might have been. Rather, it was designed to give us a greater overall insight into the strengths and weaknesses

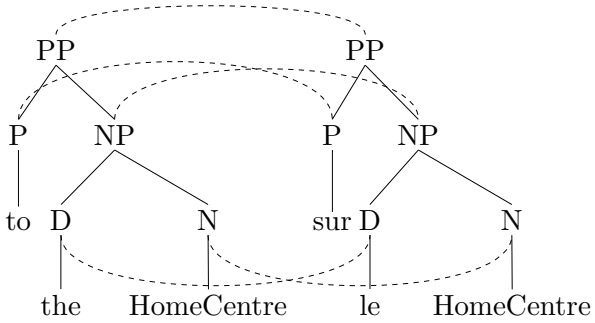
of the algorithm as well as helping us better understand the automatic evaluation scores (cf. (Hearne et al., 2007)).

Before looking at divergent cases, we first observed that the alignment algorithm generally produced accurate output for simple translation cases with relatively isomorphic tree structures. Examples (3.7) and (3.8) illustrate cases where aligner configurations correctly identified equivalent constituents where length, word order and tree structure all match exactly. For short phrases, such examples are common.

reattach the scanner to the HomeCentre ⟶ *replacez le scanner sur le HomeCentre*



(3.7)



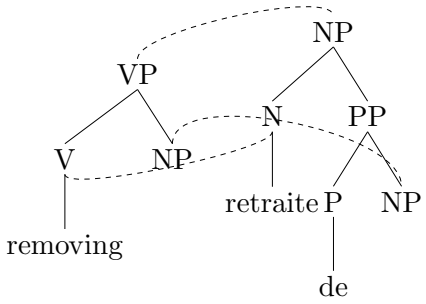
(3.8)

Nominalisation

Instances of nominalisation are commonly presented to the aligner in the HomeCentre data. Consider, for example, the alignments as given by both configurations in (3.9) where the English verb phrase *removing the print head* is realised as the French noun phrase *retraite de la tête d'impression.* As the algorithm does not take into consideration the labels on the tree, but rather the likelihood that the surface strings are translations of each other, there is no impediment to the linking of the English `VP` to the French `NP`. Furthermore, the lower `NP` alignment is straightforward. Note, however, the (probably incorrect) alignment between the `VP` *removing* and the

N *retraite*. This alignment did not appear in the manual alignment as the annotator considered the meaning equivalence to be between *removing* and *retraite de*.

removing the print head ⟶ *retraite de la tête d'impression*

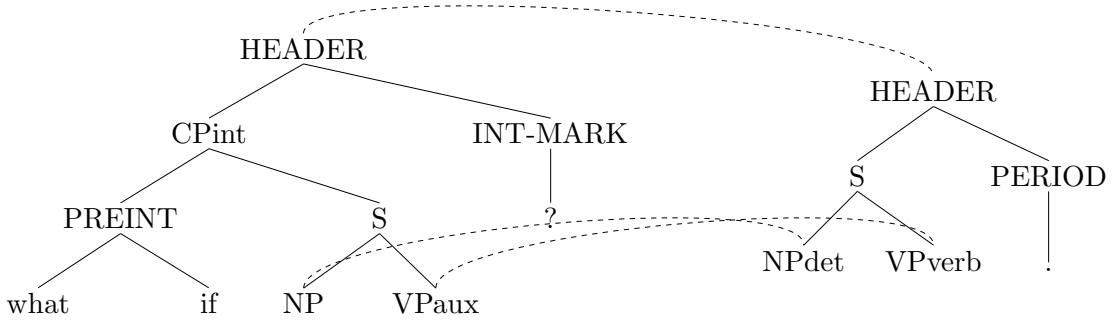


(3.9)

Stylistic Divergences

It is also common for sentences expressing the same concept to have different surface representations for simply stylistic reasons. We see an example of this in (3.10) where the English section header is phrased as a question, whereas in French the correspondant is a declarative statement. The tree pair in (3.10) also exemplifies the correct alignments as output by both aligner configurations.

What if the scanner does not work? ⟶ *Le scanner ne fonctionne pas.*

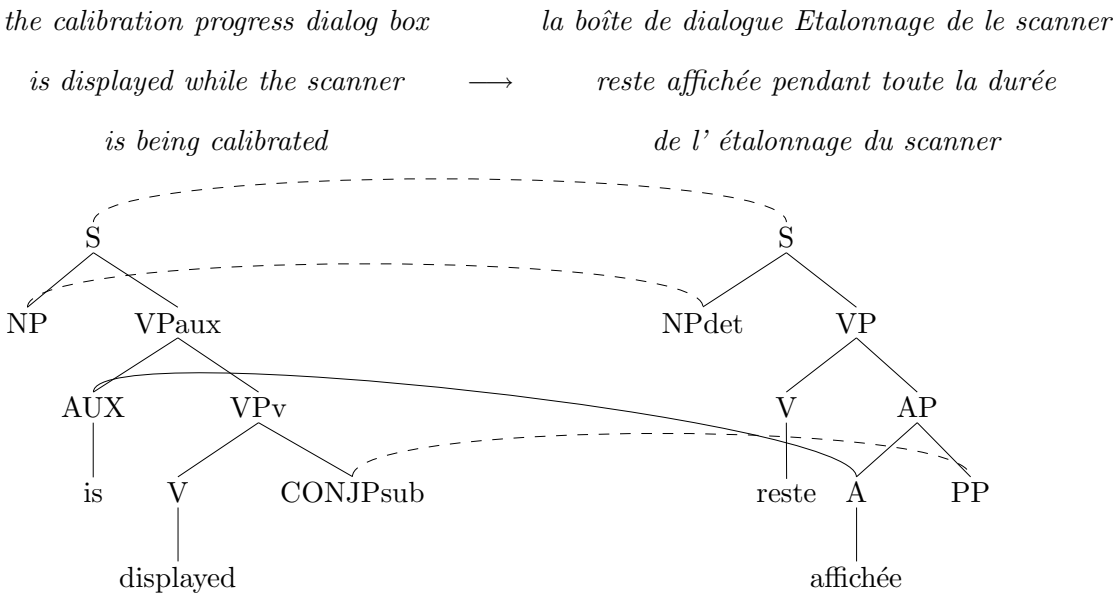


(3.10)

Head-switching

Another complex translation case presented to the aligner is that of head-switching where the head word in the source language sentence corresponds to a non-head word in the target language realisation. An example of head-switching is given in

(3.11), where the dashed alignments represent the manual alignments and the solid link (between `AUX` *is* and `A` *affichée*) represents an erroneous alignment introduced by both aligner configurations. Obviously we attribute this error to poor lexical choice on the part of the algorithm where we find it tends to have difficulty aligning frequently occurring lexical items, such as *is*, which may have many possible translational equivalents given the available statistics.



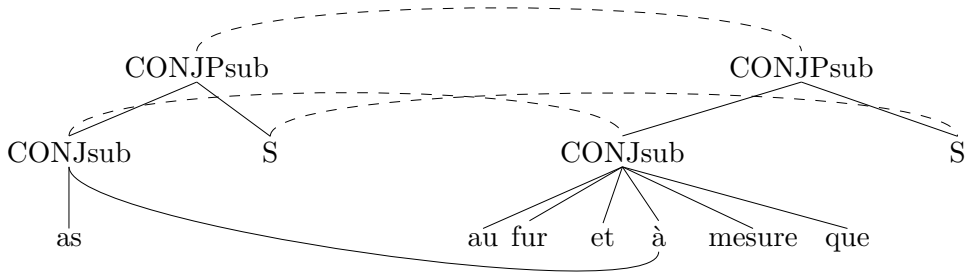
(3.11)

Lexical Divergences

Lexical divergences, where a single word in the source language can correspond to many words in the target language and vice versa, occur frequently in the data and the algorithm captures them with regularity. For instance, *skip2_score2* produced the output shown in example (3.12) by the dashed links, which exactly matches the manual alignment produced for that tree pair. This outcome is very desirable because, as we described in Section 3.2.4, when calculating the score for a particular alignment hypothesis, we not only consider the translational equivalence of the dominated substrings, but also the translational equivalence of the remainder of the source and target sentences. In this way, links can be inferred even when constituent substrings are lexically divergent. Furthermore, *skip2_score2* normalises for length

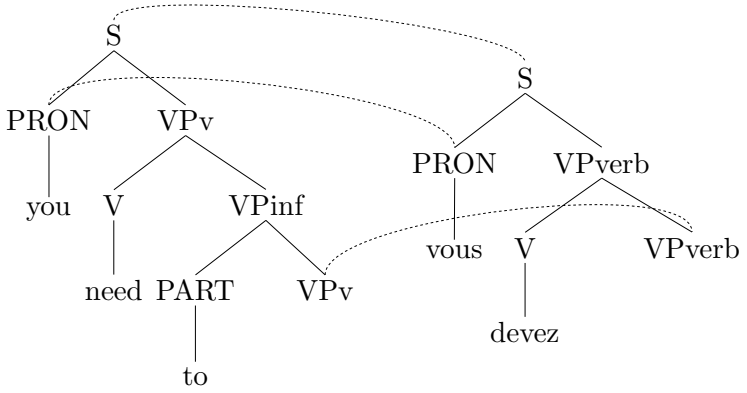
specifically when scoring, which aids in capturing this alignment. *skip2_score1_span1* errs by introducing a 1-to-1 alignment as illustrated by the solid link.

the scanner will move across ⟶ *le scanner se déplace le long de la page au fur*
the page as it scans *et à mesure que il effectue la numérisation*

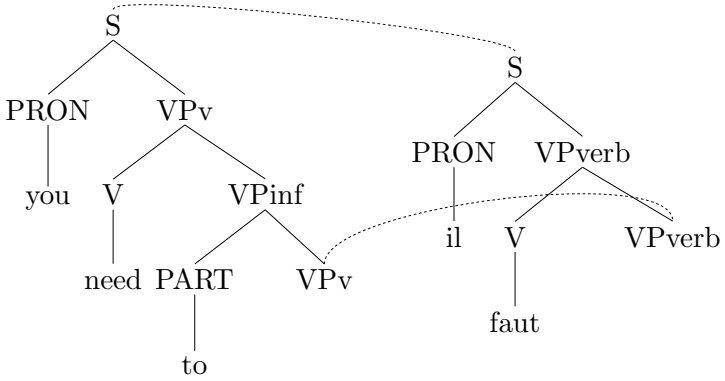


(3.12)

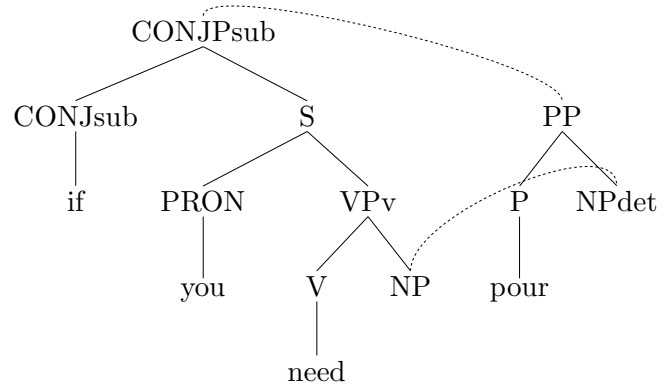
There are many other instances in the data of how frequently occurring words can vary greatly in terms of how they are translated. This phenomenon is illustrated for the English verb *to need* in examples (3.13) – (3.16).



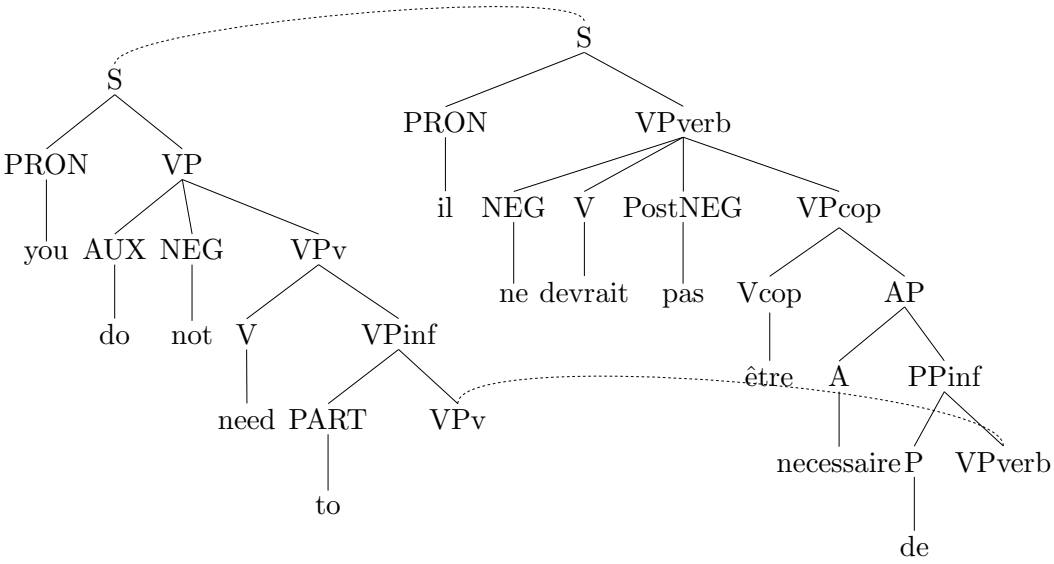
(3.13)



(3.14)



(3.15)



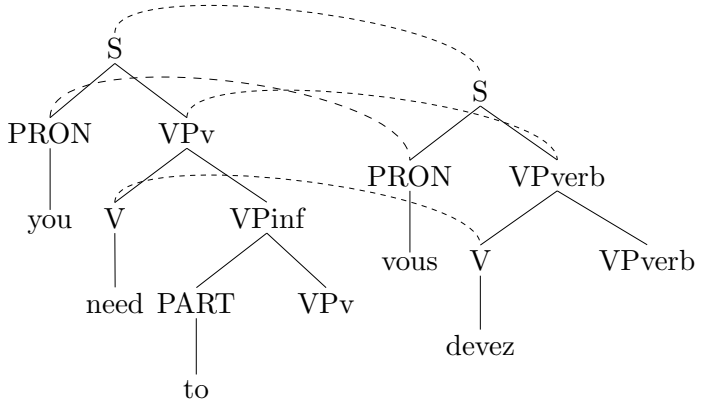
(3.16)

you need to `X` can be realised as both *vous devez* `X` and *il faut* `X` in French, as shown in examples (3.13) and (3.14). This differs, however, when the object is nominal rather than sentential: *if you need* `X` is shown in (3.15) to translate as *pour* `X`. Finally, we show in example (3.16) that the negative *you do not need to* `X` can translate as *il ne devrait pas être necessaire de* `X`, which literally means 'it should not be necessary to X' in English.[6]

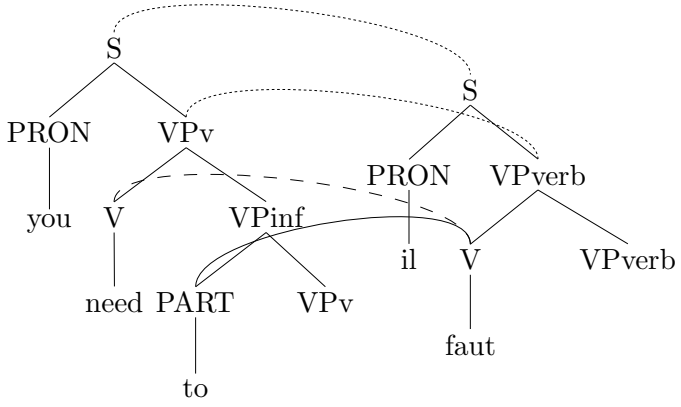
These examples are handled reasonably well by both configurations of the alignment algorithm, again due to the strength of the equivalence relation between the object constituents. For example, in (3.17) and (3.18) we show the automatically aligned versions of the tree pairs shown in (3.13) and (3.14). Again we see lexi-

[6]We note that this is just a subset of the French realisations for the verb *to need* which occur in the HomeCentre corpus.

cal alignments in the automatic output not present in the manual alignments; the annotator considered the equivalences to be (*need to*, *devez*) and (*you need to*, *il faut*). While the case for linking *need* with *devez* is arguable, the link between *need* and *faut* is incorrect. The alignments in (3.17) were produced by both automatic configurations. The tree in (3.18) show misalignments produced by both *skip2_score1_span1* (the solid link) and *skip2_score2* (the dashed link). The dotted links are those in common between the two configurations. The misalignment produced by *skip2_score2* is attributed simply to poor lexical choice, while the lexical misalignment in *skip2_score1_span1* is due to the induction of an erroneous link at a higher level in the tree pair which consequently caused the poor lexical selection.



(3.17)



(3.18)

3.3.5 Discussion and Conclusions

Given all evaluation scenarios it is clear we have developed a viable alternative to manual alignment when it comes to the contruction of parallel treebanks. As we discussed in section 2.1.1, although the goals of manual alignment may not

ultimately be the same as those of automatic alignment, they still serve as a solid baseline. To this effect, we saw a good balance between precision and recall in section 3.3.2 when comparing the automatically induced alignments against the gold standard. This performance was also reflected in the translation task in section 3.3.3, where translation scores for the automatically induced alignments were very competitive and DOT translation coverage increased over the manual alignments.

One aspect of the alignments that was not reflected between the experiments was the quality of the lexical alignments. We noticed in early experiments that poor lexical choice was an issue, hence our introduction of the features *span1* and *score2*. Despite this, in Table 3.1 we saw a huge increase in precision when measured only in terms of non-lexical links which told us that our word alignments were not so accurate. However, this did not necessarily carry over to the translation experiments as the evaluation scores for the automatic configurations often improved over the manual configuration. The explanation for this may lie in how the MT system we used works; because DOT displays a preference for using larger fragments when building translations wherever possible, the impact of inconsistencies amongst smaller fragments (i.e. word-level alignments) is minimised. The issue of poor-quality lexical alignments was again highlighted in the manual analysis where we saw that the majority of errors, when capturing translational divergences, were due to poor lexical choice.

Regarding the possible configurations of the aligner, while no single configuration consistently outperformed the rest, it was clear that a number of features introduced consistently lead to better performance. For example, in both the intrinsic and extrinsic evaluations, *skip2* outperformed *skip1*. Furthermore, *score2* outperformed *score1* (when used without *span1*), while turning on *span1* also lead to improvements.

It is clear that further improvements lie in improving word-alignment quality. There are a number of possible avenues to explore to this effect, such as inferring word-alignment probabilities from alternative alignments techniques to GIZA++, e.g. (DeNero and Klein, 2007; Deng and Byrne, 2008; Lardilleux and Lepage, 2008;

Lambert, 2008), or identifying particularly troublesome alignments, such as those between function words and punctuation, and dealing with them as a pre-processing step. However, this is beyond the scope of this thesis, and remains for further research (cf. section 6.1).

3.4 Summary

In this chapter, we presented the development and evaluation of a novel algorithm for automatically inducing sub-sentential alignments between context-free phrase-structure trees in order to produce parallel treebanks. The algorithm, presented as an alternative to the time-consuming and error-prone process of manual alignment, induces links regardless of the constituent labelling scheme of the trees and on a language pair-independent basis. We have shown the algorithm to have a high correlation with manual alignments in terms of precision and recall, while allowing enough leeway for it to build parallel treebanks which can outperform manually aligned treebanks when used as training data for a DOT system. We have also illustrated the algorithm's capability to capture complex translational divergences between English and French.

In the next two chapters, we depart from further development and evaluation of the alignment algorithm. Rather, we use it as a tool for building parallel treebanks and subsequently investigate how we can exploit them across other paradigms of MT. However, we do see the alignment algorithm being used successfully to align larger volumes of data across a number of different language pairs, thus consolidating our claims and evaluations presented here.

Extensions, optimisations and additional evaluation of the alignment algorithm can be found in the work and dissertation of Ventsislav Zhechev (Zhechev and Way, 2008; Zhechev, 2009) who pursued this line of research over the course of his Ph.D studies.

Chapter 4

Exploiting Parallel Treebanks in Phrase-based SMT

In the previous chapter, we described a sub-tree alignment algorithm which provides us with a means for building large-scale parallel treebanks which can be exploited in MT. As we discussed in Section 2.2, translation models in PB-SMT systems are estimated from statistical word alignments induced across sententially aligned parallel corpora. They do not rely on linguistically motivated information in order to extract phrase pair correspondences. It has been shown that restricting the set of phrase pairs, extracted in this way, to those that correspond to syntactic constituents is harmful to translation accuracy (Koehn et al., 2003). However, these experiments also demonstrated that there is a gap in the space of phrase pairs extracted by PB-SMT systems that could potentially be filled by constituent-based phrase pairs. In our case, these constituent-based phrase pairs are extracted from parallel treebanks built automatically using statistical parsers and the sub-tree alignment algorithm of Chapter 3.

We hypothesise that adding linguistically motivated constituent-based phrase pairs, extracted from a parallel treebank, to the translation model of a PB-SMT system (where the parallel treebank was built over the same parallel corpus from which the phrase-based translation model was originally derived) may help to im-

prove translation accuracy in two instances:

1. by introducing new phrase correspondences that were not extracted by the PB-SMT system, and

2. adding probability mass in the model to those potentially more reliable phrase pairs that were extracted via both methods.

The second case occurs as those phrase pairs which have been seen in both the parallel treebank and the baseline model will have increased frequency and will consequently be assigned higher probability by maximum-likelihood estimation (cf. section 2.2.3).

In the remainder of this chapter we investigate the extent to which phrase pair correspondences derived from automatically built parallel treebanks can be exploited within the PB-SMT framework. In contrast to those approaches which aim to induce phrase translation models exclusively from tree-based data, we will *supplement* existing phrase-based models with the parallel treebank phrase pairs. Following this we explore a number of alternative methods for harnessing the information encoded in parallel treebanks, such as word alignments, in this paradigm of MT.

4.1 Supplementing PB-SMT with Syntax-Based Phrases: pilot experiments

In this section we describe some small-scale pilot experiments we carried out (Tinsley et al., 2007a) in order to test our hypothesis: that supplementing phrase-based translation models with syntactically motivated phrase pairs extracted from parallel treebanks, automatically generated over the same training data, can lead to improvements in translation accuracy. In order to do this, we carried out four translation tasks: English-to-German, German-to-English, English-to-Spanish and Spanish-to-English. For each task, a number of PB-SMT systems were built using

various combinations of baseline SMT phrase pairs and syntax-based phrase pairs extracted from parallel treebanks in the translation models.

4.1.1 Data Resources

Parallel Corpora

In the experiments we present here, two distinct data sets were used. For the English–Spanish language pair we randomly extracted a set of 4,911 sentence pairs from version 2 of the Europarl parallel corpus (Koehn, 2005). Extraction was restricted such that the English sentences were required to be between 5 and 30 words in length. This was done in order to reduce the required parsing time as well as increase the precision of the word alignment. The data set was then randomly split for training and testing with an approximate ratio of 10:1, leaving 4,411 sentence pairs in the training set and 500 test sentences.

For the English–German language pair, the data set consisted of 10,000 sentences pairs extracted randomly from version 2 of the Europarl parallel corpus. The restriction applied here again required English sentences to be between 5 and 30 words. Finally, this set was randomly split for training and testing with a ratio of 10:1, giving us a training set comprising 9,000 sentence pairs and a test set of 1,000 sentences.

These data sets, while relatively small in terms of MT, constituted a significant increase in the size of the alignment task for our algorithm when building the parallel treebanks. Based on the quality of alignment and translation output in the pilot experiments presented in this chapter, we were confident we could proceed with much larger-scale tests as described later in this thesis.

Parallel Treebanks

In these experiments, and all subsequent experiments presented in this chapter, the parallel treebanks we exploit for MT training and syntax-based phrase extraction

are derived from the original parallel corpora used to train the baseline PB-SMT systems. Similarly, the word translation probabilities used to calculate the hypothesis scores for the sub-tree aligner (cf. section 3.2.4) are also calculated from the original parallel corpora in all cases.

The process of generating parallel treebanks from the parallel corpora described previously was completely automated. Firstly, each monolingual corpus was parsed using an off-the-shelf parser. The English corpus in both data sets was parsed using the parser of Bikel (2002) trained on the Penn II treebank (Marcus et al., 1994). The Spanish corpus was also parsed using Bikel's parser, this time trained for Spanish on the Cast3LB Treebank (Civit and Martí, 2004) as described in (Chrupała and van Genabith, 2006). Finally, the German corpus was parsed using the BitPar parser (Schmid, 2004) which was trained on the German TiGer treebank (Brants et al., 2002). The final step in the annotation process was to align the newly parsed parallel corpora at sub-sentential level using the alignment algorithm of Chapter 3. We did this using the algorithm configuration *skip2_score2_span1*.[1]

Given that our parallel treebanks here were automatically generated — as they were in all subsequent experiments presented in this thesis — the question arises as to their accuracy given the potential for error propagation due to the various automatic processes employed. Of course, parsing errors can be found in the trees and alignment errors do occur, but we are satisfied that the accuracy of the automatic tools we employ is sufficient to demonstrate our hypothesis. For instance, the three parsers we use for these experiments have high reported accuracy: 88.88% labelled f-score for English (Bikel, 2002), 83.96% labelled f-score for Spanish (Chrupała and van Genabith, 2006) and 81.13% f-score for German (Schmid, 2004). Investigating the impact of parse errors on alignment and subsequent translation tasks, while be-

[1]Using this configuration is slightly counter-intuitive given the findings in Chapter 3. However, this decision was taken following discussions with my colleague, Ventsislav Zhechev, who continued research on the alignment tool whilst working towards his Ph.D. thesis (Zhechev, 2009). He confirmed (personal communication, July 2007) that, based on empirical evidence given further development on, and improvements to the alignment algorithm, this configuration consistently performed most accurately. This was later reported in Zhechev and Way (2008).

yond the scope of this work, is certainly an avenue for future research. Furthermore, we have adopted a philosophy whereby we make use of whatever resources in terms of parsers and corpora are available in order to produce parallel treebanks. The alternative to this is to manually craft parallel treebanks, which is wholly impractical on the scale with which we are working in MT.

4.1.2 Phrase Extraction

In order to investigate our hypothesis, we must extract two sets of phrase pairs: word alignment-based phrase pairs[2] as used in PB-SMT systems, and syntax-based phrase pairs as extracted from our parallel treebanks.

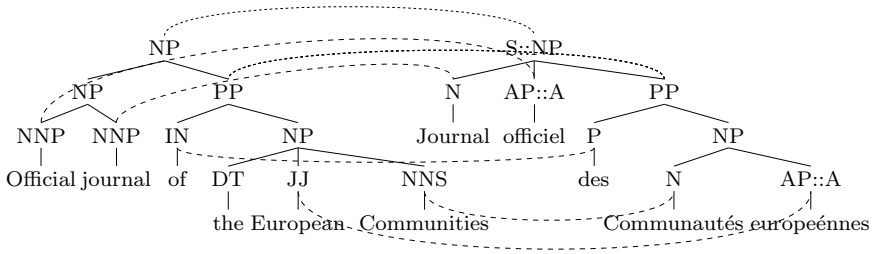
Baseline phrase pairs are extracted using the open source Moses toolkit (Koehn et al., 2007). During this process, the intersection of bidirectional GIZA++ word alignments are refined using the *grow-diag-final* heuristic and extracted by Moses as described in section 2.2.2. Syntax-based phrase pairs are extracted from the parallel treebanks according to the automatically induced sub-tree alignments. These phrase pairs correspond to the yields of all linked constituent pairs in a given tree pair.

We will illustrate this process with an example. In Figure 4.1 we see an example sentence pair from an English–French parallel corpus. Figure 4.1(a) shows the parallel treebank entry for this pair, while Figure 4.1(b) shows its refined word alignment according to the PB-SMT system. The combined set of extracted phrase pairs to be added to the translation model is given in Figure 4.1(c). We can see that while there is overlap between the two sets of phrase pairs ($*$), there are also a certain number of phrase pairs unique to the parallel treebank ($\diamond$). These phrase pairs represent the gap in the baseline phrase pairs we referred to at the beginning of this chapter. Supplementing the baseline model with the syntax-based phrase pairs allows the gap to be somewhat filled, thus increasing the translation coverage of the model. Additionally, the resulting modified combined probability model will have a higher likelihood attached to these hypothetically more reliable phrase pairs

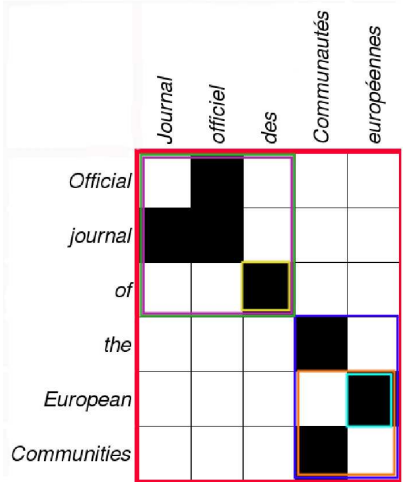
[2]We will henceforth refer to these phrase pairs as *baseline phrase pairs*.

Training Sentence Pair

"Official journal of the European Communities" ↔ "Journal officiel des Communautés européennes"



(a)



(b)

† Official journal	↔	Journal officiel	
† Official journal of	↔	Journal officiel des	
∗ Official journal of the/ European Communities	↔	Journal officiel des/ Communautés européennes	
∗ of	↔	des	(c)
∗ of the European Communities	↔	des Communautés européennes	
∗ the European Communities	↔	Communautés européennes	
∗ European	↔	européennes	
◇ Communities	↔	Communautés	
◇ Official	↔	officiel	
◇ journal	↔	Journal	

Figure 4.1: Example of phrase extraction for the given sentence pair depicting: (a) the aligned parallel tree pair; (b) the word alignment matrix (the rectangled areas represent extracted phrase pairs); (c) the combined set of extracted phrase pairs where: ◇ = only extracted from (a); † = only extracted from (b); ∗ = extracted from both (a) and (b).

occurring in the intersection of the two sets.

4.1.3 MT System Setup

For each translation task, we created three translation models comprising:

- only baseline phrase pairs (Baseline);
- only syntax-based phrase pairs (Syntax_only);
- a direct combination of both sets of phrase pairs (Baseline+Syntax).

Our PB-SMT systems were built using Moses for word alignment, baseline phrase extraction, model estimation and decoding. In the direction combination model (Baseline+Syntax), probabilities were calculated via relative frequency — as described in section 2.2.3 — using the combined phrase pair counts from the baseline and syntax-based sets. Trigram language modelling was carried out, on the target side of the parallel corpora, using the SRI language modelling toolkit (Stolcke, 2002). We did not carry out any parameter tuning in these experiments given the small amount of training data. All translations were again evaluated using the metrics BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Lavie and Agarwal, 2007).

4.1.4 Results

The results for the four translation tasks are presented in Tables 4.1–4.4. Looking firstly at the smaller data set, the results for the English–Spanish language pair are given in Tables 4.1 and 4.2. Adding the syntax-based phrase pairs (Baseline+Syntax) leads to significant improvements over the baseline across all three evaluation metrics. For example, we see a 1.02% absolute (5.78% relative) increase in BLEU score from English–Spanish,[3] and a 1.26% absolute (7.18% relative) increase from Spanish–English.[4] Using syntax-based phrase pairs only in the transla-

[3]En–Es: 4.36% relative NIST increase; 4.12% relative METEOR increase.
[4]Es–En: 4.92% relative NIST increase; 2.77% relative METEOR increase.

tion model does not improve over the baseline according to BLEU score, but results for the other metrics vary.

Looking now at the results for the English–German language pair presented in Tables 4.3 and 4.4, we see less pronounced, but nonetheless significant, improvements across all three metrics when supplementing the baseline model with syntax-based phrase pairs. From English–German we observe a 0.73% absolute (6.16% relative) increase in BLEU score,[5] while from German–English we see a 0.65% absolute (4% relative) increase.[6] Again, using only syntax-based phrase pairs performs slightly worse than the baseline in terms of BLEU score while varying across the other metrics.

4.1.5 Discussion

The principal aim of these experiments was to investigate whether phrase pairs extracted from our parallel treebanks could impact positively on translation accuracy in PB-SMT. The findings here suggest that this is indeed a viable hypothesis. If we examine the sets of extracted phrase pairs further, we obtain an indication as to where the improvements are coming from. Looking at the frequency information for the English–German phrase pairs presented in Table 4.5, we see that approximately 77.6% of the syntax-based phrase pairs were not extracted in the baseline model. In the combined model (Baseline+Syntax), this constituted 26.42% of the total number of phrase pairs. A further 7.63% of the phrase pairs were found in the intersection of the two sets, with the remaining 65.9% extracted by the baseline model only.

A similar situation is seen when we look at the English–Spanish data in Table 4.5. Again, a large proportion — approximately 68% — of the syntax-based phrase pairs were not found in the baseline model, and these constituted 20.65% of the total number of phrase pairs in the combined model. Just 9.58% of the phrase pairs occurred in the intersection of the two sets. As we discussed previously, it

[5]En–De: 4.56% relative NIST increase; 2.55% relative METEOR increase.
[6]De–En: 4.81% relative NIST increase; 3.41% relative METEOR increase.

English–Spanish			
Configuration	**BLEU**	**NIST**	**METEOR**
Baseline	0.1765	4.8857	0.4515
+Syntax	**0.1867**	**5.0898**	**0.4701**
Syntax_only	0.1689	4.8662	0.4560

Table 4.1: English–Spanish translation scores.

Spanish–English			
Configuration	**BLEU**	**NIST**	**METEOR**
Baseline	0.1754	4.7582	0.4802
+Syntax	**0.1880**	**4.9923**	**0.4935**
Syntax_only	0.1708	4.8664	0.4659

Table 4.2: Spanish–English translation scores.

English–German			
Configuration	**BLEU**	**NIST**	**METEOR**
Baseline	0.1186	4.1168	0.3840
+Syntax	**0.1259**	**4.3044**	**0.3938**
Syntax_only	0.1055	4.1153	0.3796

Table 4.3: English–German translation scores.

German–English			
Configuration	**BLEU**	**NIST**	**METEOR**
Baseline	0.1622	4.9949	0.4344
+Syntax	**0.1687**	**5.2474**	**0.4492**
Syntax_only	0.1498	5.1720	0.4327

Table 4.4: German–English translation scores.

is the combined effect of these two elements — the proportion of novel phrases being introduced into the model and the increased likelihood being placed on the phrase pairs in the intersection — that yields improved translation accuracy over the baseline. In Section 4.2.1, we describe experiments which give us further information as to the role played by these two factors in the combined model.

Language	**Phrase Type**	**#Phrases**	∩
En-De	Baseline	104,839	10,879
	Syntax	48,537	
En-Es	Baseline	77,639	9,9374
	Syntax	29,575	

Table 4.5: Frequency information regarding the number of phrase pairs extracted from the baseline system and from the parallel treebank for the English–German and English–Spanish data sets. ∩ is the number of phrase pairs in the intersection of the two sets.

Figure 4.2 presents some examples of the type of phrase pairs that were unique to the syntax-based set from the English–Spanish task. We note that many of these phrase pairs contain the possessive ending *'s* on the English side[7] which is frequently misaligned during statistical word alignment. We highlighted this particular issue in our discussion on tree alignment vs. word alignment in section 2.1.1 (cf. example (2.2) on page 11). Additionally, we see instances of longer phrase pairs which are relatively easy to capture as constituents in the parallel treebank, but which require a more precise word alignment for baseline phrase pair extraction.

the union 's	↔	de la unión
the council 's	↔	del consejo
yesterday 's	↔	de ayer
the european union 's/ recommendations	↔	las recomendaciones/ de la unión europea
the joint debate on/ the following reports	↔	el debate conjunto de/ los siguientes informes

Figure 4.2: Phrase pairs unique to the syntax-based set.

Examples of how this gave rise to improvements in translation accuracy can be

[7]This possessive is analysed as a separate token during parsing, alignment and translation.

seen below, where we show output for the Baseline (Base) and Baseline+Syntax (B+S) models for English-to-Spanish translation (where the vertical bars indicate the segments used to build the translation during decoding). In (4.1), we see the possessive phrase *de la sra schroedter* captured as a single constituent given the addition of the syntax-based phrase pairs to the B+S model, while the *'s* is translated out of context in the Baseline model. Turning to example (4.2), we again see the possessive phrase captured as a single unit in the B+S model. Similar examples were found throughout the system output whereby both the Baseline and Baseline+Syntax models arrived at the same translation, but the Baseline+Syntax model did so by using a single segment while the Baseline model pieced together smaller segments to form the final translation. This is a desirable property of the model as we are more likely to achieve fluent output if we exploit longer, previously seen exemplars. Recalling the discussion in Section 2.7, this was the motivation for the introduction of the phrase penalty feature in the log-linear model.

Src:	Mrs Schroedter 's report...
Ref:	El trabajo de la Sra Schroedter...
Base:	Señora Schroedter | del | informe...
B+S:	El informe | de la Sra Schroedter...

(4.1)

Src:	The commission 's proposals
Ref:	Las propuestas de la comisión
Base:	La comisión | propuestas de
B+S:	Las propuestas de la comisión

(4.2)

The 'Syntax_only' Models

The experiments of Koehn et al. (2003) demonstrated that restricting baseline phrase pairs to those corresponding to syntactic constituents in parallel trees is harmful to translation quality by as much as 0.04 BLEU points (a ∼17% relative decrease). These results were attributed to the fact that many legitimate translation pairs ex-

tracted in PB-SMT models which may be non-intuitive or non-syntax-based phrase pairs, such as *house the* and *there are*, do not correspond to syntactic constituents and are consequently filtered out to the detriment of translation performance. Thus, if we were to employ only our syntax-based phrase pairs in a translation model, we would expect to see similar results to the restricted model of Koehn et al. (2003).[8] Looking at the translation performance of our Syntax_only model, we can analyse the performance of a syntax-only model in terms of our experiments and compare our findings to those of Koehn et al. (2003).

In Table 4.5, we see that there are significantly fewer syntax-based phrase pairs than baseline phrase pairs: ∼54% fewer for English–German and ∼63% fewer for English–Spanish. Looking back at Tables 4.1 to 4.4, we see how this relates to translation quality. There are 12.4% and 8.27% relative drops in BLEU score from the baseline, for English-to-German and German-to-English respectively, when using only syntax-based phrase pairs (Syntax_only). However, this is not reflected in the NIST or METEOR metrics, where scores range from insignificant differences compared to the baseline to statistically significant improvements over the baseline, e.g. 3.54% relative increase in NIST for German-to-English. For English-to-Spanish and Spanish-to-English we observe a 4.5% and 2.7% drop in the respective BLEU scores, but again the NIST and METEOR scores vary. Even if we ignore the inconclusive results across the metrics, the decrease in translation performance according to BLEU across the tasks is relatively small given the size of the Syntax_only phrase table compared to the baseline. Furthermore, although the results are not directly comparable, they are a lot less pronounced than the deterioration in performance presented in the experiments of Koehn et al. (2003) — who measured their translation using only BLEU score — while we do not see consistent drops across all of our evaluation metrics.

From these results we would be inclined to believe that the syntax-based phrase

[8]Similar, but not exactly the same, as in the experiments of Koehn et al. (2003) the syntax-based phrase pairs are a subset of the baseline phrase pairs. Table 4.5 shows us that this is not the case here.

pairs extracted from parallel treebanks are more reliable than those baseline phrase pairs learned without syntax. Despite there being considerably fewer phrase pairs in the Syntax_only model, translation performance is competitive with the Baseline model. Further analysis of the set of syntax-based phrase pairs reveals a large proportion of them to be word alignments:[9] for English–German, 37.67% and for English–Spanish, 38.12%. We attribute this to the structure of the parse trees in our parallel treebanks. Of all the constituent nodes available as alignment hypotheses during the construction of the parallel treebanks, 63.69% on average were part-of-speech tags which ultimately gives rise to a large number of word alignments in the set of syntax-based phrase pairs. As we discussed in section 3.3.5, word alignment is a source of difficulty for the sub-tree aligner, specifically alignments between pairs of function words and between punctuation marks. It is possible that the presence of these high-frequency, potentially unreliable alignments in the model could be hindering the potential of the syntax-based phrase pairs to further improve translation quality. We will address this issue later in Section 4.2.

4.1.6 Summary

In this section, we presented a set of proof-of-concept experiments designed to test our hypothesis that baseline PB-SMT quality can be improved by supplementing the translation model with syntax-based phrase pairs. Our findings show that this is a viable hypothesis. The introduction of novel phrase pairs into the baseline model, along with increased likelihood attached to 'reliable' phrase pairs extracted by both methods, gives rise to significantly improved translation accuracy. We also suggest that syntax-based phrase pairs are more reliable than baseline phrase pairs based on the performance of a Syntax_only model. Finally, we suggest further improvements may be obtained if we can deal with the problem of erroneous word alignments between the parallel trees.

In section 4.2, we scale these experiments up by almost two orders of magnitude

[9] A word alignment in this case is a 1-n or n-1 alignment, where $n \geq 1$.

to determine whether the hypothesis holds. In addition, we carry out a further series of experiments in order to investigate alternative ways to exploit the information encoded in parallel treebanks within the PB-SMT framework.

4.2 Supplementing PB-SMT with Syntax-Based Phrases: scaling up

In the experiments presented in this section, we focus our efforts on a single translation task, namely English-to-Spanish. The experimental methodology employed in the previous section is replicated here while increasing the size of the training set by approximately two orders of magnitude. Following this, we carry out a series of further tests in order to investigate alternative ways of exploiting parallel treebanks in the PB-SMT framework (Tinsley et al., 2009).

4.2.1 Experimental Setup

For all translation experiments carried out in the remainder of this chapter, we used version 2 of the English–Spanish Europarl corpus.[10] After cleaning the corpus — which involved removal of erroneous sentential alignments, blank lines, sentences of over 100 tokens in length and sentence pairs with length ratio greater than 9:1 — there remained 729,891 aligned sentence pairs. These were then split into a development set of 1,000 sentence pairs and a test set of 2,000 sentence pairs, all selected at random. Test sentences were restricted in length to between 5 and 30 tokens on the English side. This resulted in an average test sentence length of 12.3 words. When building the parallel treebank from this data set, we used the same parser for the Spanish corpus as in section 4.1.1, namely Bikel (2002). For the English corpus, we used the more accurate[11] Berkeley parser (Petrov and Klein, 2007), again trained on the Penn II treebank. To the best of our knowledge, at

[10]Downloaded from http://www.statmt.org/europarl/

[11]The reported accuracy of the Berkeley parser is 90.05% labelled f-score as opposed to Bikel's 88.88%. The Berkeley parser also runs significantly faster.

the time these experiments were originally carried out, this was the largest reported parallel treebank exploited for MT training.

The baseline MT system setup was again similar to that of section 4.1.2. We used the Moses (Koehn et al., 2007) toolkit for phrase extraction, scoring and decoding. All translation systems were tuned to the BLEU metric on the development set using minimum error-rate training (Och, 2003), as implemented in Moses. 5-gram language modelling was carried out on the target side of the parallel corpus using the SRI language modelling toolkit (Stolcke, 2002). All translations were performed from English into Spanish and were, again, evaluated using the metrics BLEU, NIST and METEOR. Statistical significance was tested using bootstrap resampling, with a confidence value of p=0.05 unless otherwise stated.

4.2.2 Direct Phrase Combination

The first set of experiments we carried out replicated those in section 4.1.4. Again, we built three models using only baseline phrase pairs (Baseline), only syntax-based phrase pairs (Syntax_only) and a direct combination of the two sets of phrase pairs (Baseline+Syntax). The results of these translation experiments are presented in Table 4.6.

Config.	**BLEU**	**NIST**	**METEOR**
Baseline	0.3341	7.0765	0.5739
+Syntax	**0.3397**	**7.0891**	**0.5782**
Syntax_only	0.3153	6.8187	0.5598

Table 4.6: Results of large-scale direct combination translation experiments.

Our findings here are similar to those of section 4.1.4. We see that adding the syntax-based phrase pairs to the baseline model leads to smaller, but statistically significantly improved translation accuracy across *all* metrics (0.56% absolute increase in BLEU score, 1.56% relative increase[12]). As before, we attribute this to a

[12]We quote these improvements for BLEU score as system parameters were optimised over this metric and thus it is the most appropriate for analysis.

combination of two factors: the introduction of novel phrase pairs into the translation model, and the increased probability mass given to more reliable phrase pairs found in the intersection of the two sets. Both of these elements can be seen to good effect when we examine the sets of phrase pairs further. In the combined model, 16.79% of the entries are unique phrase pairs introduced from the parallel treebank, while a further 4.87% obtain increased likelihood having been introduced by both the baseline and syntax-based sets of phrase pairs. The exact figures are provided in Table 4.7.

Resource	**#Phrase Tokens**	**#Phrase Types**	$\cap$
Baseline	72,940,465	24,708,527	1,447,505
Syntax	21,123,732	6,432,771	

Table 4.7: Frequency information regarding the number of phrase pairs extracted from the baseline system and from the parallel treebank for the English–Spanish Europarl data set.

These findings raise a further interesting question. Although the hypothesis that supplementing the baseline model with syntax-based phrase pairs still holds, the improvements are not as pronounced as those seen in section 4.1.4, when smaller training sets were used. This may be attributable to the decreased presence of the syntax-based phrase pairs in the combined model. For example, if we look at Table 4.8, we see that the percentage of syntax-based phrase pairs found overall is considerably smaller given the larger data set. These figures are not directly comparable given the different training corpora used. However, in section 4.2.6 we describe an experiment whereby we increase the size of the training set incrementally and analyse the effect on translation performance (Tinsley and Way, 2009).

Looking back at Table 4.6, we again see that using the syntax-based phrase pairs alone (Syntax_only) does not lead to any improvements over the baseline, this time across all three evaluation metrics. Once more, however, we could interpret this drop in accuracy (5.96% relative BLEU score) as being disproportionate with the considerably fewer number of phrase pairs in the Syntax_only model compared to

Data	Unique Syntax	∩ Syntax
∼5k	20.65%	9.58%
∼10k	26.42%	7.63%
∼730k	16.79%	4.87%

Table 4.8: Statistics of the prominence of syntax-based phrase pairs in combined models given training set size. Data = sentence pairs in training sets. Unique Syntax = % of novel phrase pairs introduced from the parallel treebank. ∩ Syntax = % of syntax-based phrase pairs also extracted in baseline model.

the Baseline model — there are almost 4 times fewer phrase pairs — thus lending further credence to our suggestion in Section 4.1.5 that the syntax-based phrase pairs are of higher quality than the baseline phrase pairs. However, as the overall space of extractable phrase pairs is restricted by both syntactic constituents and sub-sentential alignments (as is the case in parallel treebanks), the Syntax_only model simply lacks sufficient coverage to improve upon the baseline.

In section 4.1.1, we discussed the issue of parser quality and how we were satisfied that their accuracy was sufficient to demonstrate our hypothesis. We note at this stage that improvements have been made on a large-scale by exploiting parallel treebanks despite some level of parser (and alignment) noise. Given this, we suggest that as parsing and alignment quality continue to improve, translation accuracy will follow suit, and so we can consider our results here to be a lower bound on improvements achievable using these automatic techniques.

Further Experiments

As we suggested at the end of section 4.1.5, it is possible that high-frequency, low-quality word alignments found in the set of syntax-based phrase pairs could be adversely affecting the quality of the combined translation model. In order to investigate this further, we carried out an additional experiment whereby we restricted in two ways the manner in which the syntax-based phrase pairs were introduced into the combined model in two ways. Firstly, we added only "strict phrase pairs" to the baseline model. We define a strict phrase pair here as an m-to-n alignment where

both m and n are greater than 1. In doing this, all word alignments are removed from the set of syntax-based phrase pairs and the only contribution to the combined model is a set of reliable strict phrase pairs. This would give us an indication as to whether, in general, the word alignments were harming translation performance.

Our second method of restricting the syntax-based phrase pairs aims at refining the previous method. Rather than removing all word alignments, we only remove those which do not reach a certain threshold τ. This threshold is based on the lexical translation probability table produced by GIZA++.[13]

Algorithm 5 Filtering Word Alignments

```
for all syntax-based word alignments do
  if word alignment is found in the t-table then
    if it occurs above assigned threshold τ then
      keep in the set of syntax-based phrase pairs
    else
      remove from the set of syntax-based phrase pairs
    end if
  else
    keep in the set of syntax-based phrase pairs
  end if
end for
```

Using this method, presented in Algorithm 5, a syntax-based word alignment which occurs in the t-table is removed if it falls below the threshold. For the purposes of this experiment, we arbitrarily set the threshold as the 50th percentile of entries in the t-table. The intended effect here is to retain the novel syntax-based word alignments while filtering out those "poor" alignments — even though they may be frequently occurring in the set of syntax-based phrase pairs — according to GIZA++ and our threshold.

The results of these experiments are presented in Table 4.9. We see even further significant improvements over the baseline for all three metrics (0.73% absolute; 2.18% relative increase in BLEU) when using only strict syntax-based phrase pairs. In this configuration, the translation model was reduced by 3% compared to the combined model having removed 1,308,577 entries in total. By removing the influ-

[13]These are the same lexical translation probabilities used to calculate the translational equivalence scores for the sub-tree alignment algorithm of Chapter 3.

ence of the unreliable word alignments, the overall probability model was improved while removing some redundancy and the potential for further search errors during decoding. When filtering the syntax-based phrase pairs using the threshold (Filter Threshold), we still see a significant improvement over the baseline. However, the difference relative to the combined model (Baseline+Syntax), while an improvement across all three metrics, is not statistically significant. In total, only 10.55% of the syntax-based word alignments were removed.

Config.	**BLEU**	**NIST**	**METEOR**
Baseline	0.3341	7.0765	0.5739
+Syntax	0.3397	7.0891	0.5782
Strict phrases	**0.3414**	**7.1283**	**0.5798**
Filter Threshold	0.3400	7.1093	0.5792

Table 4.9: Effect of restricting the set of syntax-based phrase pairs.

From these results, it is clear that unreliable word alignments are still affecting translation as leaving them out gives rise to further improvements in translation performance. In terms of ultimately overcoming this issue, we could potentially investigate the use of an improved threshold, rather than the arbitrary value chosen here, to find the optimal set of syntax-based word alignments to use. However, we believe that this avenue of work has limited potential and that future efforts in this area would best served improving the word alignments within the sub-tree alignment algorithm.

4.2.3 Prioritised Phrase Combination

In all previous experiments, we directly combined the counts of the observed baseline and syntax-based phrase pairs in the translation model, producing modified probabilities with higher likelihood assigned to those phrase pairs in the intersection of the two sets, as well as introducing novel phrase pairs. In this section, we examine an alternative approach to phrase combination — prioritised combination — originally presented by Hanneman and Lavie (2009) in terms of incorporating

non-syntax-based phrase pairs into a syntax-based MT system.

Following this method, given two sets of phrase pairs, for example A and B, we prioritise one set over the other. Assuming we have prioritised set A, when combining the two sets, we only add phrase pairs from set B if their source-side phrases are not already covered by some entries in A. For example, if the English source phrase *in the corner* existed in the syntax-based set with the target side *en el rincón* and in the baseline set with the target side *en la esquina*, assuming we were prioritising the syntax-based set, we would only add *in the corner* $\leftrightarrow$ *en el rincón* to the combined set (where in direct combination we would add both).

The motivation behind this approach is that we may believe one set of phrase pairs to be more reliable than the other: the prioritised set. Thus, when the prioritised set provides a translation for a particular source phrase, we opt to trust it and not introduce further ambiguity from the other set of phrase pairs.

In the experiments we present here, we build a model in which we prioritise the syntax-based phrase pairs over the baseline phrase pairs. Our idea here is that, given our findings in section 4.1.5, we believe the syntax-based phrase pairs to be more reliable, and so by prioritising them, the overall effect is a syntax-based model supplemented with non-constituent-based phrase pairs from the baseline set. For completeness, we also build a model in which the baseline phrase pairs are prioritised. The results of these experiments are presented in Table 4.10.

Config.	**BLEU**	**NIST**	**METEOR**
Baseline+Syntax	**0.3397**	**7.0891**	0.5782
Syntax Prioritised	0.3339	6.9887	0.5723
Baseline Prioritised	0.3381	7.0835	**0.5789**

Table 4.10: Translation results using a prioritised combination of phrase pairs.

Prioritising the syntax-based phrase pairs leads to a significant drop in translation accuracy compared to the direct combination model (Baseline+Syntax). The resulting translation model has 7.79% fewer entries than the direct combination. By prioritising the syntax-based phrase pairs, we no longer have an overlap between the

two sets of phrase pairs, and so we do not see the benefit of the increased likelihood on those phrases in the intersection. It is the absence of this factor that leads to the drop in performance. These findings are congruent with those of Hanneman and Lavie (2009), who also saw a drop in accuracy when employing syntax prioritisation over direct combination in the context of their statistical transfer-based MT system (cf. Section 2.3.1).

Turning to the baseline-prioritised model, while we may have expected similar results to the syntax-prioritised model due to the absence of the phrase pairs in the overlap, we see no significant difference compared to the direct combination. This lack of overlap phrases is compensated for by a reduction in the number of syntax-based word alignments in the model. In the direct combination model, 20.41% of the syntax-based entries are word alignments. In the baseline-prioritised model, only 1.93% of the syntax-based entries are word-alignments. This can be attributed to the baseline model containing many of the source sides of the ill-formed syntax-based word alignments and, consequently, those alignments are not added to the model. Some examples of these syntax-based word alignments that were not included are given in Figure 4.3.

I	↔	mi
am	↔	me
.	↔	y
to	↔	que
was	↔	que
—	↔	de
to	↔	”

Figure 4.3: Ill-formed syntax-based word alignments not included in the baseline prioritised model.

Given these findings, we believe the direct combination approach to be the most advantageous method for combining the two sets of phrase pairs and that its benefits will be further exemplified when the syntax-based word alignments are improved.

4.2.4 Weighting Syntax-Based Phrases

In section 4.2.2, we showed that we can improve baseline translation quality by directly adding syntax-based phrase pairs into the model. Given this, our next set of experiments investigates whether giving more weight to the syntax-based phrase pairs in the translation model will yield further improvements. Based on our previous suggestions that the syntax-based phrase pairs appear to be more reliable, our motivation here is that if we further increase the probability mass assigned to them, they are more likely to be selected at decoding time which would consequently result in more accurate translations. In order carry this out, we built three translation models — with a direct combination of baseline and syntax-based phrase pairs — in which we counted the syntax-based phrase pairs twice, three times and five times when estimating phrase translation probabilities. The results of these experiments are show in Table 4.11.

Configuration	**BLEU**	**NIST**	**METEOR**
Baseline+Syntax	**0.3397**	**7.0891**	**0.5782**
+Syntax x2	0.3386	7.0813	0.5776
+Syntax x3	0.3361	7.0584	0.5756
+Syntax x5	0.3377	7.0829	0.5771

Table 4.11: Effect of increasing relative frequency of syntax-based phrase pairs in the direct combination model.

The findings here are slightly erratic. Doubling the presence of the parallel treebank phrase pairs (+Syntax x2) lead to statistically insignificant differences (albeit lower) compared to the baseline across all metrics, while counting them three times (+Syntax x3) lead to a significant drop ($p=0.02$) in translation accuracy. Counting them five times (+Syntax x5) again lead to insignificant (yet lower) differences. We suspect these results are due to the fact that, while increasing the likelihood of the reliable phrase pairs, we are also increasing the influence of the unreliable translation pairs, such as the word alignments discussed previously.

Given these negative results for weighting the syntax-based phrase pairs more

heavily, a natural follow-up experiment was to build a model in which we weighted them less heavily. More specifically, we built a direct combination model in which we counted each syntax-based phrase pair 0.5 times when estimating phrase translation probabilities. The results of this experiment, presented below in Table 4.12, show a small, but not statistically significant, improvement over the direct combination model.

Configuration	**BLEU**	**NIST**	**METEOR**
Baseline+Syntax	0.3397	7.0891	0.5782
Half-weights	**0.3404**	**7.1050**	**0.5792**

Table 4.12: Effect of weighting syntax-based phrase pairs less heavily in the direct combination model.

Intuitively, this model is similar to the baseline prioritised model in that it will most likely choose a baseline phrase pair where it exists, and default to syntax-based phrase pairs when no baseline phrase pair exists. However, this model has the additional advantage of increasing still further the likelihood of phrase pairs in the intersection as we are not discarding anything. It is this combination of factors that ultimately results in improved translation accuracy over the baseline prioritised model.

To conclude our analysis of alternative weighting strategies for the syntax-based phrase pairs, we carried out one final experiment in which we exploit the Moses decoder's (Koehn et al., 2007) ability to employ two[14] independently scored phrase tables. Rather than combining the counts of the baseline and syntax-based phrase pairs, phrase translation probabilities are calculated for each set of phrase pairs individually and, in theory, the minimum error-rate training selects the optimal weights for the features in each model given the development set. The decoder then chooses the most likely target language translation by selecting phrases from both phrase tables. Table 4.13 shows the performance of this system relative to the Baseline+Syntax configuration.

[14]In our case we are dealing with two sets of phrase pairs. The decoder can, in fact, employ

Configuration	BLEU	NIST	METEOR
Baseline+Syntax	**0.3397**	**7.0891**	**0.5782**
Two Tables	0.3365	7.0812	0.5750

Table 4.13: Effect of using two separate phrase tables in the translation model.

We obtain no improvement over our baseline using this approach. Although this method would appear to be the most intuitive way to combine the two sets of phrase pairs, we suspect that by scoring them individually, we again lose the increased probability mass on those phrase pairs in the intersection. As we have previously demonstrated this to be and important factor in achieving improvements using the two sets of phrase pairs, the results here are not surprising.

4.2.5 Filtering Treebank Data

Koehn et al. (2003) demonstrated that using longer phrase pairs does not yield much improvement when translating, and they occasionally lead to worse results. For these reasons, a default setting in Moses when extracting baseline phrase pairs is to restrict their length to 7 tokens. We used this setting in all experiments carried out thus far in this thesis, yet no restriction was placed on the length of the syntax-based phrase pairs. Therefore, it is possible that some of the longer phrase pairs in the syntax-based set were harming translation performace. In order to investigate this, we built a direct combination model in which we filtered out all syntax-based phrase pairs with more than 7 tokens.

The effect of this filtering is shown in Table 4.14, where we see inconsistent fluctuation in scores across the metrics. This indicates that the longer syntax-based phrase pairs were originally used only sparsely for translation in the Baseline+Syntax model. We confirm this when analysing how the translation hypotheses were constructed. In the Baseline+Syntax model, only 18 phrases of length greater than 7 tokens were used, which constituted 0.183% of the total number of phrases used.

more than two phrase tables, e.g (Srivastava et al., 2009).

Thus, removing the 38.22% of syntax-based phrase pairs over 7 tokens in length had negligible ramifications on translation. From this we can conclude that when combining the syntax-based phrase pairs with the baseline phrase pairs, they may be restricted in length similar to the baseline phrase pairs, resulting in a smaller phrase table without loss of translation accuracy.

Config.	BLEU	NIST	METEOR
Baseline+Syntax	**0.3397**	7.0891	**0.5782**
-Filtered	0.3387	**7.0926**	0.5767

Table 4.14: Effect of filtering longer syntax-based phrase pairs.

4.2.6 Training Set Size: Effect on Influence of Syntax-Based Phrase Pairs

From our findings in Sections 4.1.4 and 4.2.2, it would appear that the influence of the syntax-based phrase pairs in direct combination with baseline phrase pairs is reduced as the size of the training set increases. However, we cannot be certain of this as the experimental conditions were different for the two sets of results. In order to investigate this further, we designed an experiment, using the English–Spanish parallel corpus and treebank of section 4.2.2, in which we increased the size of the training corpus incrementally and evaluated translation performace on a common test set (Tinsley and Way, 2009). Starting off with a subset of 10,000 training sentence pairs, we built four MT systems with the following combinations of phrase pairs: Baseline, Syntax_only, Baseline+Syntax and Strict Phrases. We then repeated this process, doubling the size of the training corpus until we had used the entire corpus. All other experimental conditions are the same as those experiments presented in section 4.2.2, including the development and test sets. Having completed translation for these 28 system configurations, we evaluated the results and analysed the trends as the training corpus size increased. Figure 4.4 summarises the outcome of these experiments.

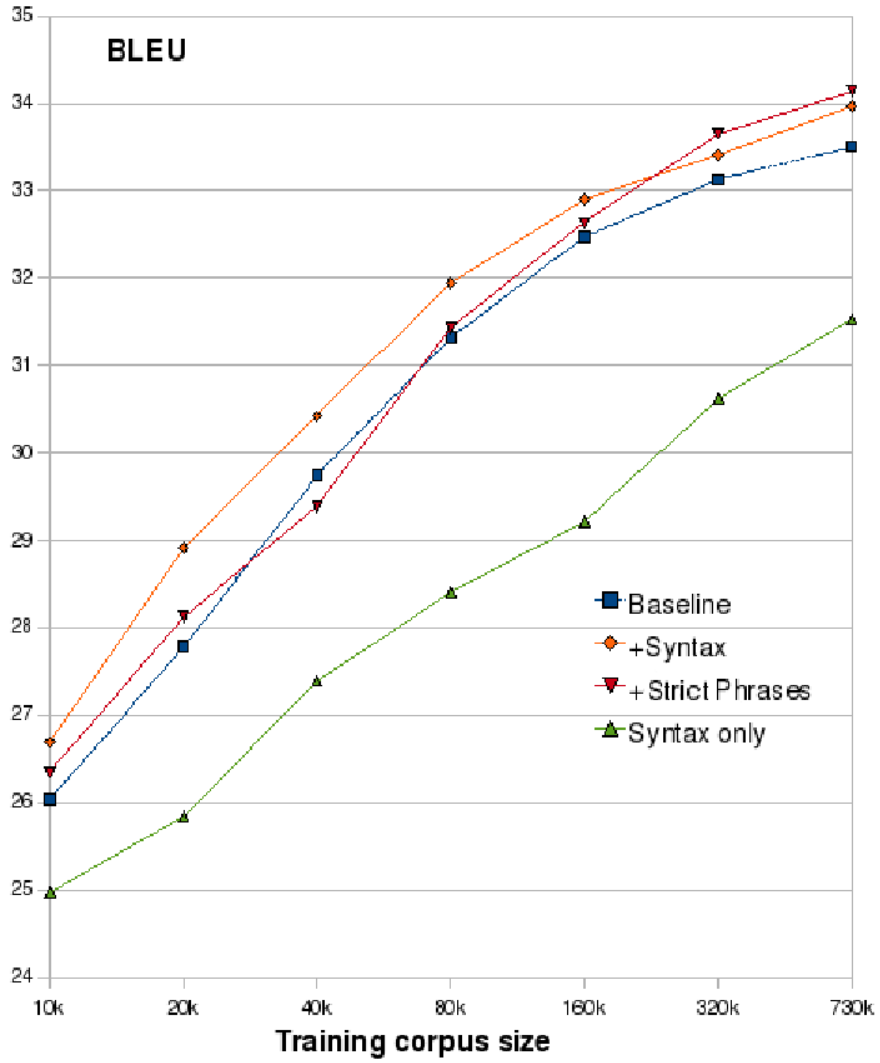


Figure 4.4: Effect of increasing training corpus size on influence of syntax-based phrase pairs.

The first and most obvious point to note is that, in general, as expected, translation performance increases as the training set size increases. Aside from that, we see that the gains achieved over the baseline by adding the syntax-based phrase pairs (+Syntax) steadily diminish as the training corpus size grows, with the greatest improvement being seen at 20,000 training pairs. We obtain further insight into this if we examine the graph in Figure 4.5. As the training set grows, many of the phrase pairs that were unique to the syntax-based set are also extracted by the baseline method. As a consequence, each time we increment the number of sentence pairs in the training set, the percentage of phrase pairs in the direct combination (Baseline+Syntax) unique to the syntax-based model decreases. Conversely, the number of phrase pairs unique to the baseline model increases by approximately 3% at each increment, while the number of phrase pairs seen in the intersection of the two sets steadily drops by approximately 2%. This tells us that the baseline model is simultaneously introducing more novel phrase pairs into the combined model as well as learning phrase pairs that may have previously been unique to the syntax-based set. It is a combination of these factors that ultimately diminishes the complementary effect of the syntax-based phrase pairs in the combined model as the training corpus increases.

Another potential contribution to the decreasing influence of the syntax-based phrase pairs as the training set grows may be the increased likelihood of the aforementioned unreliable word alignments. Looking back at the strict phrase model (+Phrases) in Figure 4.4, in which we remove syntax-based word alignments, we see that translation performance converges with, and eventually outperforms, the Baseline+Syntax model as the training set approaches 730,000 sentence pairs. This indicates to us that with larger training sets, we introduce more unreliable word alignments into the translation model and subsequently, it is preferable to leave them out.

Such a suggestion is corroborated by the work of Way and Groves (2005) and Groves (2007), who discovered that when building hybrid translation models using

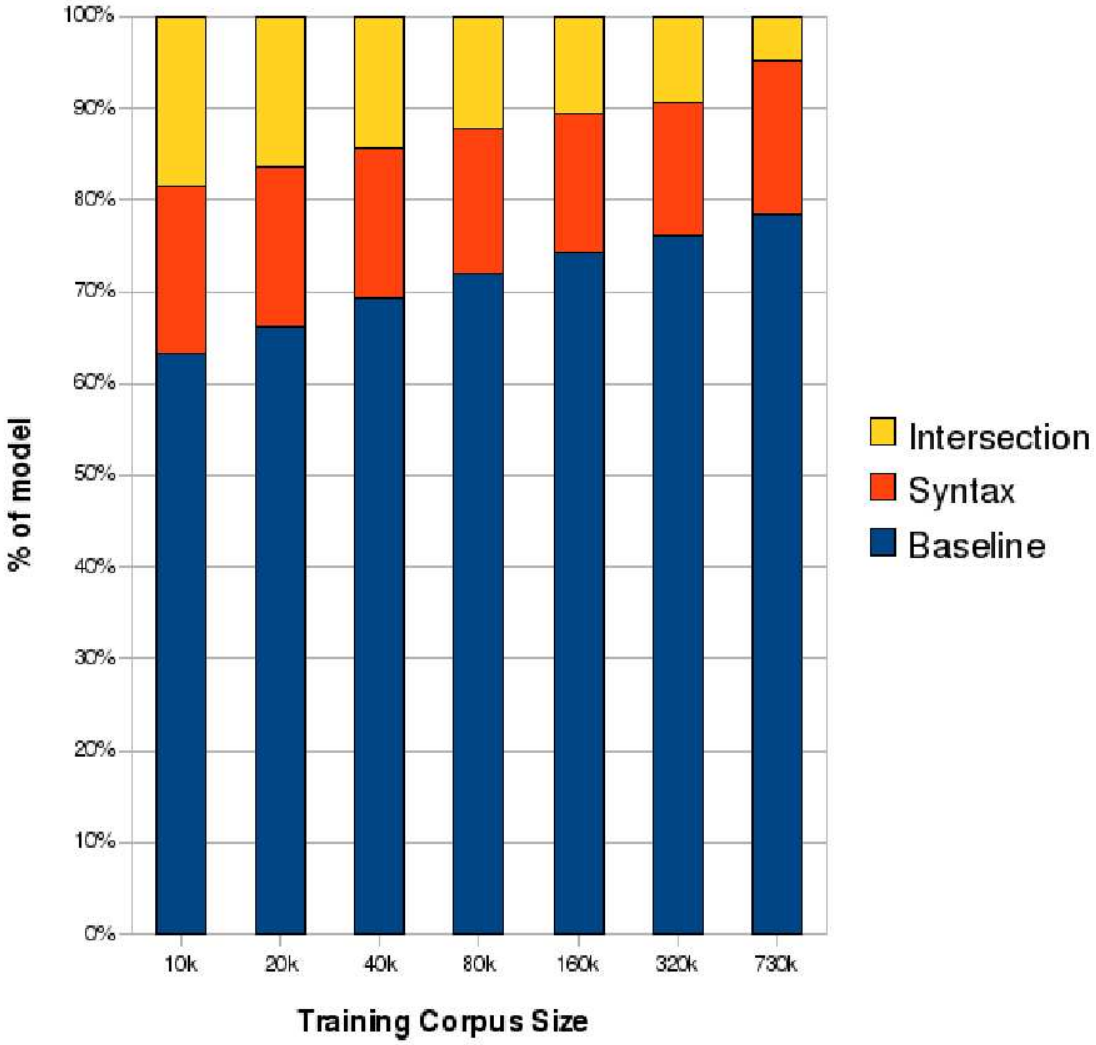


Figure 4.5: Proportions of data in the Baseline+Syntax model from the baseline and syntax-based sets given the increasing training corpus size.

EBMT chunks and baseline phrase pairs, low precision EBMT word alignments were harming translation performance and ultimately it was better to omit them from the hybrid model as the training set grew.

Given these findings, we would expect this trend to continue upwards. That is, if we were to double the size of the training set once more we might assume that no gains will be achieved by supplementing the baseline model with syntax-based phrase pairs. We carry out some experiments in Chapter 5 which give us further insight into this. It may also be the case that this approach is best suited to MT systems with smaller training sets, for example in scenarios in which limited data resources, or disc space, are available. Under such conditions, more benefit from the use of the syntax-based phrase pairs could be seen. We investigate this suggestion in further in section 4.4.

4.3 Exploring Further Uses of Parallel Treebanks in PB-SMT

The experiments described so far in this chapter have focussed on investigating whether supplementing baseline models with syntax-based phrase pairs can improve translation accuracy. In this section, we consider alternative ways in which the information encoded in parallel treebanks can be incorporated into the PB-SMT framework.

4.3.1 Treebank-Driven Phrase Extraction

One oft-cited reason for the inability of syntax-based MT systems to improve upon the state-of-the-art is that using only constituent-based translation units is too restrictive and leads to a reduction in the overall coverage of the system (Koehn et al., 2003; Chiang, 2005). Translation units such as the English–German pair *there is* $\leftrightarrow$ *es gibt* will never be extracted as a stand-alone constituent phrase pair despite

being a perfectly acceptable translation pair as it will never be parsed as a single constituent. In an attempt to overcome this problem, we sought some ways in which we could exploit the linguistic information encoded in our automatically generated parallel treebanks to extract a set of non-constituent-based phrase pairs for use in a PB-SMT system. The motivation behind this is that instead of only having a set of restrictive syntax-based phrase pairs, or a set of statistically learned baseline phrase pairs, we would have a set of "linguistically informed" phrase pairs that would potentially be more reliable than either of the alternatives.

In all of our previous experiments, baseline phrase pairs were extracted using the method described in section 2.2.2. As we mentioned in section 4.1.2, the intersection of birectional GIZA++ alignments is refined using the *grow-diag-final* heuristic and then used to seed the extraction of phrase pairs with Moses. Instead of doing this, we use the word alignments encoded in the parallel treebank to seed the Moses phrase extraction process and build a translation model. Additionally, we take the union of the parallel treebank word alignments and the refined GIZA++ word alignments and again use this to seed Moses' phrase extraction process. This gives us two translation models in which the phrases have been learned with some input from the "linguistically-aware" parallel treebank. Given these two models, we build a further two models in which we supplement them with the actual syntax-based phrase pairs themselves. Using these four translation models (summarised in Table 4.15), we carry out translation experiments using the exact same data set and experimental configuration as the previous English–Spanish experiments of section 4.2.

Table 4.16 gives the results of these experiments. The first two rows in the table, showing the results from section 4.2.2, represent our baseline here. In the third row (Treebank_ex), we see that seeding the phrase extraction with treebank word alignments leads to a large drop in translation accuracy compared to the baseline. Supplementing this model with the syntax-based phrase pairs (Treebank_ex+Syntax) significantly improves performance, as we would expect given our previous findings, yet it still does not approach the accuracy of the baseline

Treebank_ex	Moses phrase extraction seeded with the word alignments encoded in the parallel treebank.
Treebank_ex+Syntax	Direct combination of the model produced by Treebank_ex and the syntax-based phrase pairs from the parallel treebank.
Union_ex	Moses phrase extraction seeded with the union of the word alignments encoded in the parallel treebank and the refined GIZA++ word alignments.
Union_ex+Syntax	Direct combination of the model produced by Union_ex and the syntax-based phrase pairs from the parallel treebank.

Table 4.15: Description of the 4 translation models produced using treebank-driven phrase extraction.

Seeding the phrase extraction using the parallel treebank word alignments leads to an unwieldy amount of phrase pairs in the translation model — approximately 86.6 million (92.9 when including the syntax-based phrase pairs) — many of which are completely useless, e.g. *framework for olaf , in order that ↔ marco.* This is due to the fact that the parallel treebank word alignments have quite low recall and thus the phrase extraction algorithm is free to extract a large number of phrase pairs anchored by a single alignment.[15] This situation does not occur with the baseline as the word alignment refinements are designed to increase the recall of the word alignments,[16] and the phrase extraction process is tailored to this. Thus, in their current format, the parallel treebank word alignments are too sparse to be used alone for seeding the PB-SMT phrase extraction process.

The issue of word alignment recall in the parallel treebank was the motivation for the next experiment: using the union of the treebank word alignments and the refined GIZA++ alignments. Our intuition underlying this experiment is that we would simultaneously increase the recall of the statistical word alignments (by introducing novel word alignments) and the precision of the parallel treebank word

[15]In the example *framework for olaf , in order that ↔ marco*, the only word alignment anchoring the phrases was between *framework* and *marco*.

[16]The relates to creating a more densely populated word alignment matrix as we saw in Figure 2.7 on page 21.

Config.	BLEU	NIST	METEOR
Baseline	0.3341	7.0765	0.5739
+Syntax	**0.3397**	**7.0891**	0.5782
Treebank_ex	0.3102	6.6990	0.5564
+Syntax	0.3199	6.8517	0.5639
Union_ex	0.3277	6.9587	0.5679
+Syntax	0.3384	7.0508	**0.5788**

Table 4.16: Translation results using different word alignments to seed phrase extraction. alignments.

alignments (by reinforcing them with statistical word alignments), and create a more robust, reliable word alignment for seeding phrase extraction.

Looking again at Table 4.16, we see from the fifth row (Union_ex) that using the union of the two word alignments led to a small, but significant, drop in translation accuracy compared to the baseline across all metrics. More interestingly, we note from row six (Union_ex+Syntax) that when we supplemented this model with the syntax-based phrase pairs we saw comparable performance to the Baseline+Syntax model. This is particularly interesting as the Baseline+Syntax model contains approximately 29.7 million phrase pairs, whereas the Union_ex+Syntax model contains only 13.1 million phrase pairs. This constitutes a 56% decrease in translation model size without any significant loss of translation accuracy. These figures, and those for the other models described in this section, are given in Table 4.17. Analysing these findings further, we note that the phrase pairs in the Union_ex+Syntax model are almost a complete subset of the phrase pairs in the Baseline+Syntax model, in that all bar 170 of the 13.1 million phrase pairs in the Union_ex+Syntax are also found in the Baseline+Syntax model.

Word Alignment	#Phrases	#Phrases+Syntax
Baseline	24,708,527	29,693,793
Treebank_ex	86,629,635	92,889,746
Union_ex	7,476,227	13,105,420

Table 4.17: Comparison of the phrase table size for each model. #Phrase = number of phrases extracted using a given word alignment. #Phrase+Syntax = size of model when syntax-based phrases are included.

This discovery is a very positive and interesting by-product of these experiments. Filtering of PB-SMT translation models has been the focus of substantial research in recent years as evidenced by the number of publications of the topic: Eck et al. (2005); Johnson et al. (2007); Lu et al. (2007); Sánchez-Martínez and Way (2009) to cite but a few. What we do here differs from the conventional approach in that rather than performing filtering as a post-processing step or as a dynamic process during phrase extraction, we produce a reduced model by *a priori* constraining the phrase extraction with a dense, but precise, word alignment. While investigating these findings further is beyond the scope of this thesis, it is certainly an area that warrants more attention. There are also potentially more creative ways in which we could combine the two sets of word alignments for seeding phrase extraction. We will discuss some of these approaches further in section 6.1.

We can conclude from our experiments in this section that it is best to use refined statistical word alignments rather than parallel treebank word alignments for seeding PB-SMT phrase extraction. However, given a parallel corpus and a parallel treebank, we can use all information at our disposal — statistical word alignments, parallel treebank word alignments and syntax-based phrase pairs — to generate concise translation models that achieve comparable translation performance to much larger baseline models.

4.3.2 Treebank-Based Lexical Weighting

In section 2.2.3 we described the lexical weighting feature employed in the log linear model of PB-SMT systems (Koehn et al., 2003). This feature checks how well the words on the source and target sides of a phrase pair translate to one another. This is done by scoring each phrase pair according to the statistical word alignments calculated by GIZA++.

Given the findings of the previous section, we considered the potential for using the parallel treebank word alignments to calculate the lexical weights for the phrase pairs in our translation models. In order to do this, we first calculated a lexical

translation probability distribution $w(s|t)$ over the treebank word alignments, which was estimated via relative frequency according to the formula in (4.3).[17]

$$f(s|t) = \frac{count(s,t)}{\sum_{s'} count(s',t)} \tag{4.3}$$

We then used this distribution to assign two new sets of lexical weights to the Baseline+Syntax model. One set of weights was calculated using the treebank lexical probabilities only. The second set of weights was calculated by combining the counts of the treebank word alignments and the statistical word alignments in order to calculate a combined lexical translation distribution, similar to the union of the word alignments in section 4.3.1. Translation results using the Baseline+Syntax model with these sets of lexical weights are presented in Table 4.18.

Config.	**BLEU**	**NIST**	**METEOR**
Baseline+Syntax	**0.3397**	**7.0891**	**0.5782**
+Treebank_weights	0.3356	7.0355	0.5732
+Combined_weights	0.3355	7.0272	0.5741

Table 4.18: Translation results using parallel treebank-induced lexical translation probabilities to calculate lexical weighting feature.

Translation performance degrades slightly compared to the baseline across all three metrics when using the new lexical weights, while the results are almost identical when comparing the two new approaches. Aside from the potential issue of alignment precision in the treebank word alignments, there are a number of possible explanations for the ineffectiveness of this approach. The majority of the phrase pairs in the combined translation model (i.e. the baseline phrase pairs) were extracted according to the statistical word alignments and would, therefore, have a high word alignment recall between the source and target phrases. To replace these word alignments with the treebank word alignments gives a lower recall which leads to less reliable lexical weights.

[17]We introduced this formula previously when discussing the feature functions of the log-linear model in Section 2.7.

Another potentially significant reason why the treebank-based lexical weights were not successful is that, for a given sentence pair, there exists only a single "hard" alignment for each aligned word. Conversely, the statistical word alignments estimated by EM see some probability mass given to word pairs not included in the final set of most likely alignments for a given sentence pair.

4.4 New Language Pairs: IWSLT Participation

In 2008, we participated in an evaluation task at the International Workshop for Spoken Language Technology (IWSLT) (Ma et al., 2008). This involved building a number of MT systems for different language pairs and, in some cases, translating output produced by automatic speech recognition (ASR) systems. This campaign was of particular interest to us for a number of reasons. Up to this point, all of our experiments concerning the combining of syntax-based phrase pairs in PB-SMT models have used only European language pairs as training data. Furthermore, one of the language pairs has always been English. The IWSLT campaign presented us with an opportunity to use our sub-tree aligner with a non-European language, namely Chinese, while also affording us the chance to train on a language pair not including English, namely Chinese–Spanish.

By using these new languages, we were able to further evaluate the language-independent nature of our sub-tree aligner as well as test the quality of the subsequent syntax-based phrase pairs in new translation tasks. This would also allow us to confirm the cross-lingual applicability of our hypothesis on the use of syntax-based phrase pairs in PB-SMT.

Finally, as we mentioned at the end of section 4.2.6, this hypothesis may be most appropriate in scenarios where only limited training resources are available. This case arises in the IWSLT task where the provided training corpora contain approximately 20,000 sentence pairs, affording us the opportunity to substantiate this claim.

4.4.1 Task Description

We participated in a number of translation tasks for language pairs and translation directions. The main data resource for training and development was the Basic Travel Expression Corpus (BTEC) (Kikui et al., 2003), a multilingual parallel corpus containing tourism-related sentences similar to those usually found in phrasebooks for a tourist going abroad (Kikui et al., 2006). For each translation task, we built a parallel treebank and subsequently created two translation models: Baseline and Baseline+Syntax. All other configurations of the MT system and evaluation setup are the same as for the experiments presented earlier in this chapter (i.e. using Moses to build the PB-SMT system and SRILM for 5-gram language modelling). We describe the data specific to each translation task in the sections below and summarise them in Table 4.19.

Chinese–English

For the Chinese–English task, the parallel training corpus comprised 21,973 sentence pairs. From this, we automatically generated a parallel treebank, parsing both sides of the parallel corpus with the Berkeley parser (Petrov and Klein, 2007) and aligning the tree pairs with our sub-tree aligner (cf. Chapter 3). The development set for each direction comprised 489 sentences, and 6 reference translations were used to evaluate translation quality.

Chinese–Spanish

For the Chinese–Spanish task, the training corpus contained 19,972 sentence pairs. As in section 4.2.1, we used the parser of Bikel (2002) to parse the Spanish side of the parallel corpus, while the Chinese side was again parsed with the Berkeley parser (Petrov and Klein, 2007) and the trees were aligned using our sub-tree aligner. The development sets contained 506 sentences and we made use of 16 reference translations to evaluate translation quality.

Pivot Task: Chinese–English–Spanish

We also took part in a Chinese–Spanish translation task in which English was used as a pivot language. To do this, we built two MT systems, for Chinese–English and English–Spanish. For this task, each system had two distinct training sets comprising 20,000 sentence pairs, and developments sets containing 506 sentences with 16 reference translations to evaluate translation quality. The same monolingual parsers as before, and the sub-tree aligner, were used to build the parallel treebanks. Translation from Chinese into Spanish was then achieved by first translating the Chinese input into English using the first half of the pivot system, and subsequently translating the English output into Spanish using the English–Spanish component.

Language Pair	Training Set	Dev Set	References
Zh–En	21,973	489	6
Zh–Es	19,972	506	16
Zh–En (pivot)	20,000	506	16
En–Es (pivot)	20,000	506	16

Table 4.19: Summary of the training and development corpora used for the IWSLT translation tasks.

4.4.2 Results

Table 4.20 below presents the results of the translation tasks in terms of BLEU score achieved on the development set. We see significant improvements in translation accuracy across all tasks when supplementing the baseline model with syntax-based phrase pairs. For Chinese–English, we see a 1.9% absolute (5.28% relative) increase in BLEU score, while for Chinese–Spanish we see a 2.31% absolute (8.57% relative) increase. Finally, for the Chinese–Spanish–English pivot task, we observe a 4.6% absolute (16.24% relative) increase in scores.

As before, these improvements can be attributed to the complementary value of the syntax-based phrase pairs in the combined model. The combination of novel phrase pairs being introduced and the increased likelihood assigned to those phrase

Config. \ Languages	Zh–En	Zh–Es	Zh–Es–En
Baseline	0.3595	0.2693	0.2832
+Syntax	**0.3785**	**0.2924**	**0.3292**

Table 4.20: Effect of using syntax-based phrase pairs on IWSLT 2008 tasks.

pairs in the intersection of the two sets of phrase pairs lead to improved translation performance. The effect of direct combination for each language pair is summarised in Table 4.21. We demonstrated in section 4.2.6 that the influence of the syntax-based phrase pairs was inversely proportional to the size of the training corpus and, thus suggested that the direct combination method may be best suited to tasks in which limited training resources are available. This is confirmed by our findings here. We see that the increase in the model size — when adding the syntax-based phrase pairs — is greater than in the larger experiments of previous sections. We also see that the percentage of phrase pairs in the intersection is slightly lower confirming, as we suggested, that as the training set grows, the baseline method learns many of the phrase pairs previously seen in the syntax-based set only.

System	Baseline	Syntax	Combo	Coverage	∩
Zh–En	158,807	86,161	213,875	34.67%	14.54%
Zh–Es	101,593	68,870	151,446	49.06%	12.56%
Zh–En (pivot)	84,025	80,431	144,630	72.12%	13.70%
En–Es (pivot)	292,209	65,628	323,884	10.84%	10.48%

Table 4.21: Impact of adding syntax-based phrase pairs to the baseline model across the IWSLT 2008 translation tasks. The *Baseline*, *Syntax* and *Combo* columns present the numbers of phrase pairs in each model for each language pairs, while the *Coverage* column shows the percentage increase in the size of the phrase table from the baseline to the combined model.

4.4.3 Conclusions

Given the findings in this section, it is clear that the sub-tree alignment algorithm is truly language-independent. We have demonstrated its applicability with a non-European language (Chinese) and across a language pair not including English (Chi-

nese and Spanish), neither of which were used during the original development of the algorithm. We have also shown that our hypothesis regarding the use of syntax-based phrase pairs in PB-SMT has multilingual applicability also. Finally, we have confirmed our suggestions — that using syntax-based phrase pairs in direct combination with baseline phrase pairs is most beneficial when only limited training resources are available — by presenting significantly improved translation performance on three independent tasks with a training corpus of 20,000 sentences pairs or fewer.

4.5 Comparing Constituency and Dependency Structures for Syntax-Based Phrase Extraction

All of our previous experiments in this chapter have used constituency parses as the basis for automatic generation of parallel treebanks and the subsequent extraction of syntax-based phrase pairs. However, there may be other techniques for syntactic analysis of sentences that would provide an alternative, potentially improved, phrase segmentation for translation. In this section, we investigate the impact of variation in syntactic analysis type — specifically, constituency parsing *vs.* dependency parsing — on the extraction of syntax-based phrase tables. Our experimental objective is to compare the relative merits of each method of annotation by measuring translation accuracy (Hearne et al., 2008). In order to do this, we automatically derive two parallel treebanks, one constituency-based and one dependency-based, and extract two sets of syntax-based phrase pairs. We then combine these directly with baseline phrase pairs and consider the value of each combined model.

4.5.1 Syntactic Annotations

The data annotation types we consider are constituency parses and dependency parses. In both cases, each sentence is tagged with part-of-speech (POS) informa-

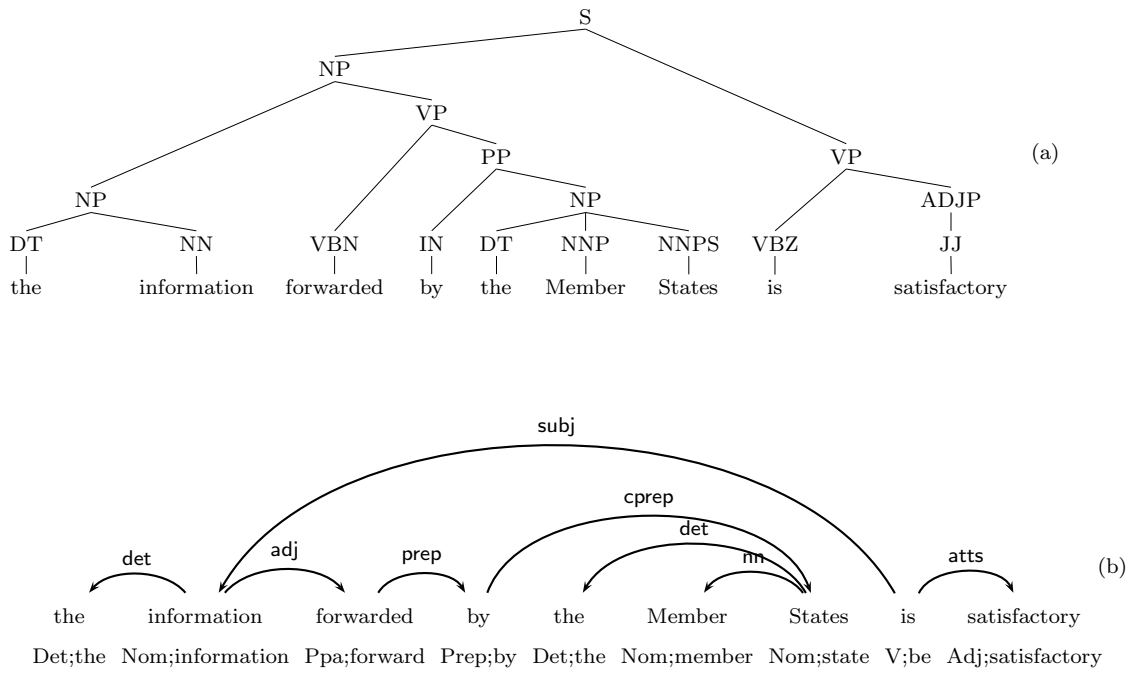


Figure 4.6: Phrase-structure tree (a) and dependency relations (b) for the same English sentence.

tion, and in the case of dependency parses a lemma is also associated with each word. Constituency parses, or context-free phrase-structure tree, make explicit syntactic constituents (such as noun phrases (NP), verb phrases (VP) and prepositional phrases (PP)) identifiable in the sentence. An example of a constituency parse is given in Figure 4.6(a), where we see that the overall sentence comprises an NP followed by a VP, each of which has some internal structure. Dependency parses make explicit the relationships between the words in the sentence in terms of heads and dependents. An example of a dependency parse is given in Figure 4.6(b), where an arc from word w_i to word w_j indicates that w_i is w_j's head and, correspondingly, w_j is w_i's dependent. These arcs are labelled such that the label indicates the nature of the dependency; in the given example, the label on the arc from *is* to *information* is labelled SUBJ indicating that *information* is the subject of *is*.

Our tree aligner of Chapter 3 has not previously been used to align dependency structures. These structures are not directly compatible with the aligner because the tool requires that the input trees be in labelled, bracketed format. While the labels themselves can be arbitrary and the branching-factor and depth of the tree are

irrelevant — for instance, a POS-tagged sentence with a single, arbitrary root label is perfectly acceptable — it must be possible to associate each node in the tree with its corresponding surface string. The output of the dependency parser, as shown in Figure 4.6, does not directly meet this requirement. Therefore, we must convert the dependency-parsed data into a bracketed format recognisable by the aligner. This is done using the method presented in Algorithm 6, by creating a set of constituents in which each constituent comprises a head and its dependents arranged as siblings in the order in which they occurred in the sentence.

Algorithm 6 Formal conversion of dependency parses.

```
Beginning with the head n of the dependency
CreateConstituent(n);
if n has dependents then
   create new constituent node c
   add n as a child of c
   for each dependent d of n do
      CreateConstituent(d)
   end for
   add c as child of previous c
else
   add n as a child of parent of n's head
end if
```

We note at this point that this conversion is purely formal rather than linguistically motivated cf. the approach of Xia and Palmer (2001). As the alignment algorithm is not concerned with the specific constituent labelling schema used, and our translation experiments require only the extraction of string-based phrase pairs for the aligned output, we pack sufficient information into the node labels during the dependency conversion such that the original dependency information is fully recoverable from the bracketed structure.

The bracketed representation for the dependency structure in Figure 4.6 is given in Figure 4.7. In this representation, we see that each node label retains the dependency information, indicating which child is head and the function of each of its dependent children. The label formats for constituents and parts-of-speech are *index;head=index;*$func_1$*=index;...;*$func_n$*=index* and *index;tag;lemma* respectively.

The single feature of dependency parses which cannot be satisfactorily encoded

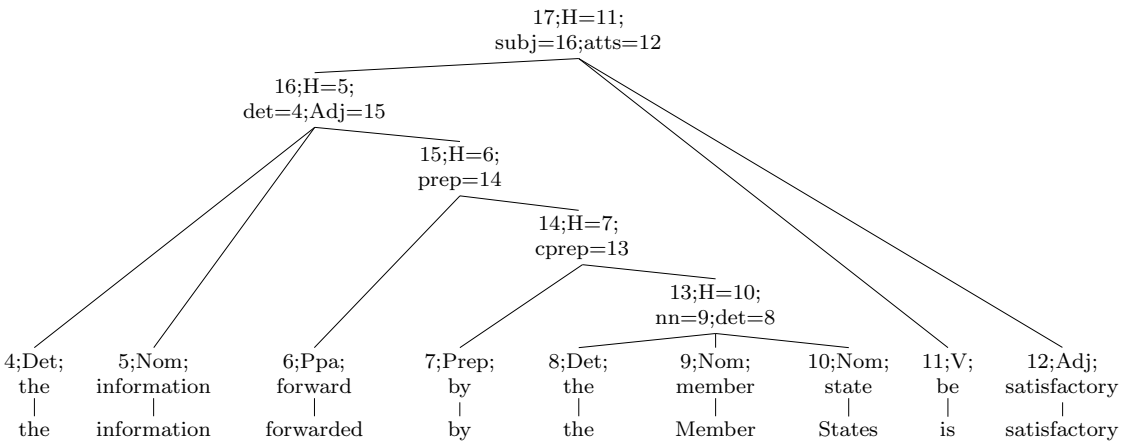


Figure 4.7: Constituency structure derived from a dependency parse.

in our bracketed representation is non-projectivity. An example of a non-projective dependency structure is given in Figure 4.8. In our bracketed representation, each head and its direct dependents are grouped as siblings under a single node according to the surface word order. In Figure 4.8, the relationship between the dependent *not* and its head *has been followed* is correctly represented by the dashed line from the root constituent 15 to constituent 12. However, as this branch crosses the one between 13 and *has*, this structure is not acceptable to the aligner. This forces us to compromise by attaching the non-projective constituent to the lowest non-crossing parent constituent. Thus, the dashed line in Figure 4.8 is dropped and the dotted line linking 12 to 13 is inserted instead. However, the true relationship is encoded in the node labelling: constituent 15's label records the fact that 13 is 12's head.[18]

4.5.2 Data and Experimental Setup

In order to investigate the relative merits of using constituency parses *vs.* dependency parses for syntax-based phrase extraction, we carried out a set of translation experiments, similar to our previous experiments, in which we directly combined the two sets of syntax-based phrase pairs with baseline phrase pairs in a PB-SMT system and evaluate translation accuracy. In the experiments we present, we used the

[18]This analysis arises from the parser's pre- and post-processing procedures, which result in deviations from standard part-of-speech tagging. We discuss which parser we use in the next section.

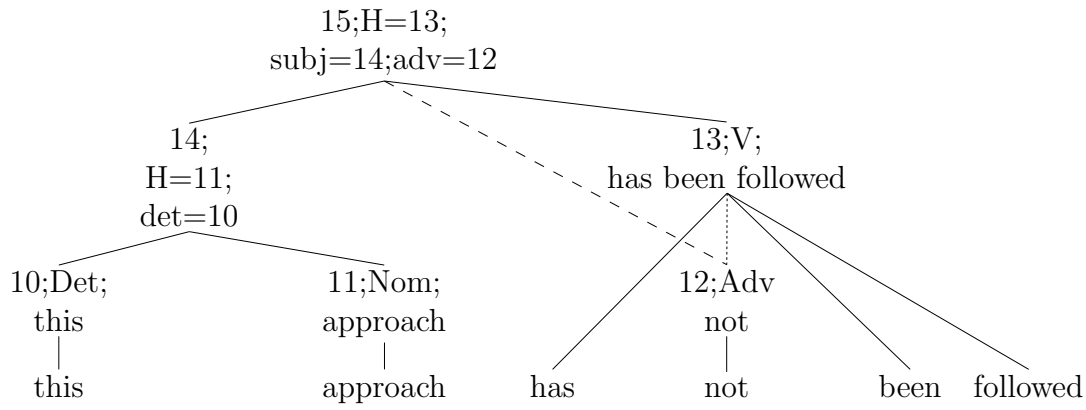


Figure 4.8: A non-projective converted structure.

JOC English–French parallel corpus provided within the framework of the ARCADE sentence alignment evaluation campaign (Véronis and Langlais, 2000; Chiao et al., 2006).[19] The JOC corpus is composed of texts published in 1993 as a section of the C Series of the Official Journal of the European Community. It contains about 400,000 words corresponding to 8,722 aligned sentences with an average sentence length of 23 words for English and 27.2 words for French.

We built our constituency-based parallel treebank using the parser of Bikel (Bikel, 2002) trained on the Penn II Treebank (Marcus et al., 1994) for English and the same parser trained on the Modified French Treebank (Schluter and van Genabith, 2007) for French. For the dependency-based parallel treebank, the corpus was parsed using the English and French versions of the SYNTEX parser (Bourigault et al., 2005). The dependency structures were converted to bracketed format using the method of the previous section and both pairs of trees were aligned using our sub-tree aligner.

In our experimental setup, we split the dataset into 1,000 test/reference pairs and 7,722 training pairs. Our PB-SMT system setup and evaluation framework was exactly the same as that used in section 4.1.3, and all translations were carried out from French into English. We built a number of translation models using baseline phrase pairs, the two sets of syntax-based phrase pairs and various direct combinations of the three.

[19]The JOC corpus is distributed by ELDA (http://www.elda.org).

4.5.3 Results

	Config.	BLEU	NIST	METEOR
1	Baseline	0.3035	6.262	0.6432
2	Con_only	0.2997	6.319	0.6359
3	Dep_only	0.2990	6.332	0.6411
4	Baseline+Con	0.3198	6.516	0.6561
5	Baseline+Dep	**0.3203**	**6.528**	**0.6572**
6	Con+Dep	0.3109	6.466	0.6467
7	Baseline+Con+Dep	0.3190	6.510	0.6556

Table 4.22: Evaluation of translation accuracy using the constituency- and dependency-based phrase pairs.

The results of our experiments are presented in Table 4.22. In analysing our results, we considered the relative merits of using constituency-annotated *vs.* dependency-annotated data, both individually and in combination with baseline phrase pairs. Looking at the first three rows of Table 4.22, we see that using either set of syntax-based phrase pairs (Con_only and Dep_only) in place of the baseline phrase pairs (Baseline) leads to lower translation accuracy according BLEU and METEOR but increased performance according to NIST. These results are akin to our previous findings using similar size data sets (cf. section 4.1.4) as the syntax-based models have considerably less coverage than the baseline model — the baseline model is 2.97 times larger than the constituency-based model and 3.19 times larger than the dependency-based model — yet the phrase pairs are more reliable.

What is interesting to note here is that there is insignificant difference between the Con_only and Dep_only models in terms of translation performance. Examining the models more closely, we see that the constituency-based model is only 7.5% larger than the dependency-based model. Furthermore, 65.35% of the phrase pairs in the constituency-based model are also found in the dependency-based model (this intersection corresponds to 70.28% of the dependency-based phrase pairs). This relative similarity in the make-up of the two models accounts for the comparable translation accuracy. These figures are summarised in Table 4.23.

In order to further compare the two sets, we observed translation accuracy when

Config.	#Phrases	∩
Constituency	79,720	52,104
Dependency	74,137	

Table 4.23: Comparison of standalone constituency- and dependency-based models.

the respective sets of phrase pairs were directly combined with the baseline phrase pairs. These results are shown in rows four and five of Table 4.22. We see that, individually, directly combining constituency- and dependency-based phrase pairs with the baseline phrase pairs (Baseline+Con and Baseline+Dep respectively) leads to statistically significant (p=0.05) improvements over the baseline. For Baseline+Con we obtain a 1.63% absolute (5.37% relative) improvement in BLEU, while for Baseline+Dep we obtain a 1.68% absolute (5.54% relative) improvement. Again, this is in line with our hypothesis of combining syntactic- and non-syntactic phrase pairs to gain improvements. However, again there is an insignificant difference between the Baseline+Con and Baseline+Dep models.

Looking at Table 4.24 comparing the two combined phrase tables, we see similar characteristics across both. In the Baseline+Con model, 7.39% of the phrase pairs were in the intersection of the baseline and constituency-based sets, while a further 19.66% of the phrase pairs were unique to the constituency-based set. In the Baseline+Dep model, 7.63% of the phrase pairs were in the intersection of the two sets of phrase pairs, while 18.03% were unique to the dependency-based set. We attribute these similarities to the insignificant differences in translation performance when comparing the two sets of syntax-based phrase pairs.

Config.	#Baseline	#Syntax	#Combo	∩
Constituency	236,789	79,720	294,728	21,781
Dependency		74,137	288,876	22,050

Table 4.24: Comparison of constituency- and dependency-based models when used in combined models.

Comparing the constituency- and dependency-based phrase pairs further, we obtain additional insight as to the similarity of the two sets of phrase pairs. Firstly,

the average phrase length of the two sets of phrase pairs is quite similar, with dependency phrases being slightly shorter on average (4.92 *vs.* 6.15 tokens). Secondly, we note that 48.96% of the constituency-based phrase pairs correspond to word alignments,[20] while this figure is 52.53% for dependency-based phrase pairs. Of these word alignments for the constituency- and dependency-based sets, 81.95% and 82.15% are respectively are found in the intersection of the two sets of phrase pairs. As PB-SMT systems have a preference for shorter phrase pairs (Koehn et al., 2003), including word alignments, when analysing the phrase pairs used to build the translation hypotheses, we see that for the Baseline+Con model, 71.54% of the phrase pairs corresponded to word alignments, while 71.09% of the Baseline+Dep phrase pairs used were word alignments. It is likely that many of the word alignments actually employed when building these translations were in the intersection of the two sets, and thus the resulting final translations, and subsequent results, are similar. When looking at identical output produced by both models, we see that this is the case. For example, in (4.4) the underlined words were translated as single token segments and were found in both the constituency and dependency set of phrase pairs.[21]

Src:	Ces chiffres doivent être évalués en tenant compte :	
Ref:	These figures must be assessed in the light of :	(4.4)
Con/Dep:	These figures must be assessed bearing in mind :	

To complete this set of experiments, we built two further translation models. Firstly, we directly combined the constituency- and dependency-based phrase pairs in to a single model, the translation result of which can be seen in row 7 (Con+Dep) of Table 4.22. The Con+Dep model improves upon the baseline by 0.74% BLEU score (absolute, 2.44% relative). This result is achieved despite the Con+Dep model

[20]Recall that a word alignment in this sense is any 1-to-n, or n-to-1 alignment where $n \geq 1$.
[21]The remaining words were translated as part of phrasal segments.

containing 57.02% fewer phrase pairs than the Baseline model, thus further highlighting the redundancy in the set of baseline phrase pairs as we originally demonstrated in section 4.3.1 when using parallel treebank data to seed baseline phrase extraction. We also attribute this outcome to the fact that the phrase pairs in the Con+Dep mode are more reliable translation pairs given their syntactic foundation. We also note here that the Con+Dep model does not achieve the same levels of translation accuracy as the Baseline+Con and Baseline+Dep models (rows 4 and 5 of Table 4.22). The higher coverage of these models, which includes complementary combination of precise syntax-based phrase pairs and non-syntactic phrase pairs not found in the Con+Dep model, accounts for the greater translation scores.

The final model we built combined all three sets of phrase pairs: baseline, constituency-based and dependency based. The performance of this model, seen in row 7 of Table 4.22 (Baseline+Con+Dep), while improving over the baseline model as we would expect, shows insignificant differences in translation accuracy when compared to the Baseline+Con and Baseline+Dep. Examining this set of phrase pairs further, we see there are only 0.97% and 3.02% more phrase pairs than in the Baseline+Con and Baseline+Dep models respectively. Very few novel phrases are introduced and so what we are essentially doing is increasing the frequency of the syntax-based phrases which we already showed to be ineffective in Section 4.2.4.

4.5.4 Conclusions

We observe that when incorporating syntax-based data into PB-SMT systems, constituency and dependency representations for syntactic analysis and subsequent phrase extraction perform equally as well (Hearne et al., 2008). We could not distinguish between either set of syntax-based phrase pairs whether they were employed in isolation or in direct combination with baseline phrase pairs. From this we can conclude that when using these representations for phrase extraction, the two representations are interchangeable and one should use whatever tools are most accurate for the language pair in question. For instance, if we were translating between Irish

and Czech, and there were dependency parsers available for those languages that were more accurate than constituency parsers for the same, we suggest it may be most appropriate to use those. Similarly, we have learned that, for a given language, if there is only a dependency parser available, it is adequate to use this in place of a constituency parser for syntax-based phrase extraction without sacrificing any potential improvements over a PB-SMT baseline.

While expanding on this particular line of research is beyond the scope of this thesis, further work has been carried out (Srivastava and Way, 2009) which scales up the experiments presented here and introduces additional techniques for syntactic annotation and phrase extraction.

4.6 Summary

In this chapter, we examined the hypothesis that syntax-based phrase pairs extracted from a parallel treebank can be used to supplement the translation model of a PB-SMT system and give rise to improvements in translation accuracy. We presented the design and execution of a series experiments which confirmed this hypothesis to be true for data sets up to approximately 730,000 sentence pairs. We also discovered that this hypothesis carries most weight with smaller training sets and that its effectiveness descreases somewhat as the training set size increases. However, we suggest that it may eventually become ineffective as the training set continues to grow. Analysing our findings further, we note that low-precision word alignments induced in the parallel treebanks have a negative impact on the contribution of the syntax-based data to the point that, until such a time as their accuracy is improved, it may be desirable to omit them from the set of syntax-based phrase pairs.

In addition to substantiating our hypothesis, a number of further important findings were made throughout the course of this chapter. We confirmed the language-independent nature of our sub-tree aligner, as well as the cross-lingual applicability of our hypothesis, by successfully employing both on previously untested languages

and language pairs. Furthermore, we demonstrated that dependency-based syntactic analyses, along with constituency-based analyses, may be used with our sub-tree aligner to produce parallel treebanks. These dependency-based parallel treebanks can then be exploited to produce comparable sets of syntax-based phrase tables and, consequently, comparable translation performance as constituency-based parallel treebanks.

In exploring alternative applications of our parallel treebanks in PB-SMT, we discovered that they can be used to seed the PB-SMT phrase extraction process to produce translation models up to 56% smaller than baseline models without any significant reduction in translation accuracy.

In the following chapter, we investigate how our automatically generated parallel treebanks can be exploited in a syntax-aware MT system by employing some of the successful techniques for phrase combination presented in this chapter.

Chapter 5

Exploiting Parallel Treebanks in Syntax-Based MT

While PB-SMT systems have achieved state-of-the-art performance in recent years, there is no direct way to incorporate syntactic information into the framework without significantly re-engineering some component(s) of the system. While this has been carried out with some success (Collins et al., 2005; Carpuat and Wu, 2007; Hassan et al., 2007; Koehn and Hoang, 2007; Stroppa et al., 2007; Haque et al., 2009a,b), these modifications still do not accommodate parallel treebanks directly as training data. In the last chapter, we demonstrated a number of ways in which parallel treebanks can be exploited within the PB-SMT framework, for instance by supplementing the translation model and constraining the phrase extraction process. However, in order to fully exploit the linguistic information encoded in our automatically-generated parallel treebanks — namely sub-tree alignments, syntactic structure and node labels — we need to employ them in an MT system that inherently makes use of this form of data. In this chapter, we describe how we exploit our parallel treebanks for use in the syntax-aware Statistical Transfer MT system (Stat-XFER) (Lavie, 2008) described previously in section 2.3.1. We stress at this juncture that the goal of the experiments presented here was not to improve over

a baseline PB-SMT system,[1] but rather to demonstrate that our parallel treebanks are viable as direct training resources and to evaluate the effectiveness of the deeper syntax encoded within the treebanks in a syntax-aware MT framework.

In section 5.1 of this chapter we describe the set-up of the Stat-XFER translation experiments and the new data set from which we build our parallel treebank. In section 5.2 we describe the bilingual phrase extraction process for syntax-based MT and detail the grammars used in the experiments, including a manually-crafted grammar and a grammar extracted automatically from the parallel treebank. Section 5.3 discusses the results of these experiments along with a detailed qualitative analysis of the translation output. Finally, in section 5.4 we replicate some of the PB-SMT experiments of Chapter 4, using a larger data set, for comparative purposes.

5.1 Data and Experimental Setup

The data set we used for the experiments presented in this chapter was the French–English section of the Europarl corpus release 3.[2] This parallel corpus, used for the 2009 Workshop on Machine Translation (WMT'09) (Callison-Burch et al., 2009), comprises 1,261,556 aligned sentence pairs. We automatically generated our parallel treebank from this corpus using the Berkeley parser (Petrov and Klein, 2007) to parse both the English and French sides — the English parser was again trained on the Penn II Treebank (Marcus et al., 1994) while the French parser was trained on the original French Treebank (Abeillé et al., 2000) — and our sub-tree aligner (Tinsley et al., 2007b) (cf. Chapter 3) to induce links between tree pairs.

As all of our experiments perform translation from French into English, we used a suffix-array language model (Zhang and Vogel, 2005, 2006) from a corpus of 430 million words,[3] including the English side of our parallel corpus, the English side

[1] We were aware, based on previously published results (i.e. (Hanneman and Lavie, 2009; Ambati and Lavie, 2008), that the Stat-XFER system was not yet capable of outperforming a PB-SMT baseline, but could nevertheless carry out translation to a sufficient standard as to serve as a useful medium for evaluating the quality of our parallel treebanks.

[2] Downloaded from http://www.statmt.org/europarl/

[3] Thanks to the MT group at the LTI in CMU for providing the language model.

of the Canadian Hansard corpus,[4] and newswire data. All systems were tuned via minimum error-rate training on the BLEU metric, using the news-dev2009a data provided by the WMT'09 as the development set. This comprised 600 sentences with an average length of 32.4 tokens. Finally, we tested the systems on the news-dev2009b set also from the workshop, which comprised 1,500 sentences with an average length of 32.4 token, and used the BLEU, NIST and METEOR metrics for automatic evaluation.

5.2 Stat-XFER: Exploiting Parallel Trees

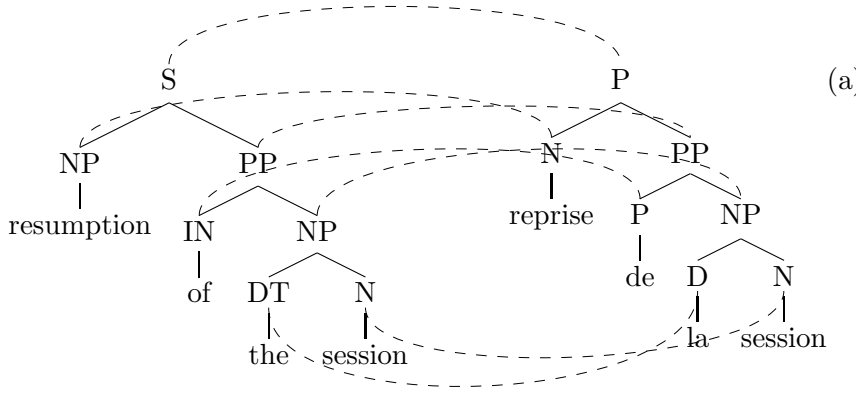
As we described in Section 2.3.1, the Stat-XFER engine exploits two language pair-dependent resources both extracted from parallel treebanks: a probabilistic bilingual lexicon (phrase table) and, optionally, a grammar of weighted synchronous context-free grammar (SCFG) rules.

5.2.1 Phrase Extraction

The difference between a Stat-XFER phrase table and that of a PB-SMT system is that each entry in the table also contains a syntactic category for the source and target phrases. Thus, each entry is a fully lexicalised SCFG expression which can later be used in conjunction with the weighted SCFG rules. This is an immediate example of how the Stat-XFER engine exploits additional information from the parallel treebank that is not exploited in PB-SMT. Looking at Figure 5.1, we see an illustration of how bilingual lexicon entries are extracted from a parallel treebank for use in the Stat-XFER system.

Similar to parallel treebank phrase extraction for PB-SMT, for each linked constituent pair we extract the surface strings dominated by the source and target nodes. The difference in the case of syntax-based MT here is that we also extract

[4] http://www.isi.edu/natural-language/download/hansard/



(a)

NP	::	N	→	[resumption]	::	[reprise]	
IN	::	P	→	[of]	::	[de]	
DT	::	D	→	[the]	::	[la]	
N	::	N	→	[session]	::	[session]	(b)
NP	::	NP	→	[the session]	::	[le session]	
PP	::	PP	→	[of the session]	::	[de la sesion]	
S	::	P	→	[resumption of the session]	::	[reprise de le session]	

Figure 5.1: An aligned English–French parallel tree pair (a) and set of extracted Stat-XFER bilingual lexicon entries (b).

the constituent node labels. Using this method, we extracted 5,461,912 bilingual lexicon entries from the French–English Europarl corpus.

5.2.2 Grammar Extraction

Grammar rules in the Stat-XFER system take a similar form to the bilingual lexicon entries. The difference lies in the fact that the right-hand sides of these SCFG productions can contain both lexicalised items as well as non-terminals and pre-terminals. This allows them to be used in conjunction with the bilingual lexicon to build full translations. For example, in Figure 5.2 we see a subset of the grammar rules extractable from the parallel tree in Figure 5.1 (a).

Constituent alignment information, shown here as co-indices on the non-terminals, indicates the correspondences between source and target language constituents on the right-hand sides of the SCFG rules as encoded in the parallel treebank. In the experiments presented in this chapter, we made use of two grammars: a manually-crafted grammar and a grammar automatically derived one from our parallel tree-

S	::	P	→	[NP^1 PP^2]	::	[N^1 PP^2]
S	::	P	→	["resumption" PP^1]	::	["reprise" PP^1]
PP	::	PP	→	[IN^1 "the" N^2]	::	[P^1 "la" N^2]
NP	::	NP	→	[DT^1 N^2]	::	[D^1 N^2]

Figure 5.2: A subset of the SCFG rules extractable from the parallel treebank entry in Figure 5.1 (a).

bank. We discuss these two grammars in greater detail below.

Manually Crafted Grammar

We make use of a small, manually crafted grammar containing nine SCFG rules. The grammar presented in Figure 5.3 was created during the development of the Stat-XFER system and used by Hanneman and Lavie (2009) in experiments on phrase combination. It defines a number of rules designed to address certain local word ordering phenomena between French and English (particularly within noun phrases). For example, we see that rules (1)–(4) in Figure 5.3 deal with the reordering of adjectives and nouns,[5] while rules (5) and (6) account for the deletion of the French preposition *de* along with further nominal reordering. Finally, rules (7)–(9) were designed to be used in conjunction with rules (2) and (4) for correct ordering of larger adjectival phrases. In section 5.3, we will present many examples of these rules being used in actual translation cases.

Automatically Derived Grammar

The second grammar we employ in these experiments was extracted automatically from our parallel treebank. As an efficient solution has yet to be found for exploiting large-scale grammars in the Stat-XFER system, we make use of a reduced grammar comprising the top-forty most frequent SCFG rules.[6] In order to extract this

[5]The complete tag sets for the English and French parses are given in Appendices A and B respectively.

[6]There were 8,233,480 SCFG rules extracted in total from the data set.

(1)	NP	::	NP	→	[N^1 A^2]	::	[JJ^2 N^1]
(2)	NP	::	NP	→	[N^1 AP^2]	::	[$ADJP^2$ N^1]
(3)	NP	::	NP	→	[D^1 N^2 A^3]	::	[DT^1 JJ^3 N^2]
(4)	NP	::	NP	→	[D^1 N^2 AP^3]	::	[DT^1 $ADJP^3$ N^2]
(5)	NP	::	NP	→	[N^1 "de" N^2]	::	[N^2 N^1]
(6)	NP	::	NP	→	[D^1 N^2 "de" N^3]	::	[DT^1 N^3 N^2]
(7)	AP	::	ADJP	→	[A^1 C^2 A^3]	::	[JJ^1 CC^2 JJ^3]
(8)	AP	::	ADJP	→	[A^1 "," A^2 C^3 A^4]	::	[JJ^1 "," JJ^2 CC^3 JJ^4]
(9)	ADJP	::	ADJP	→	[ADV^1 A^2]	::	[RB^1 JJ^2]

Figure 5.3: The manually crafted nine-rule grammar from French-to-English.

grammar, we used a rule induction toolkit[7] based on the work of Ambati and Lavie (2008). The extraction process makes use of the word alignments in our parallel treebank to infer an alternative phrase-level alignment between the tree pairs and induce an SCFG.

The automatic grammar contains a number of rules which, intuitively, are potentially useful for translation. Some of these are shown in Figure 5.4. For example, rule (1) in Figure 5.4 defines an example of adjective/noun reordering, while rules (2) and (3) allow for deletion of a preposition and article respectively, which can often be necessary. As well as these rules capturing translational divergences, the grammar contains rules such as (4) which accounts for straightforward mapping of prepositional phrases. The full forty-rule grammar is provided in Appendix C. We also demonstrate the application of many of the automatic grammar rules in actual translation cases in section 5.3 in addition to statistics regarding how often each rule was applied during translation.

[7]Downloaded from http://www.cs.cmu.edu/∼vamshi/rulelearner.htm

(1)	NP	::	NP	→	[D^1 N^2 A^3]	::	[DT^1 JJ^3 N^2]
(2)	NP	::	NP	→	["des" N^1]	::	[N^1]
(3)	NP	::	NP	→	["le" N^1]	::	[N^1]
(4)	PP	::	PP	→	["de" NP^1]	::	["of" NP^1]

Figure 5.4: Examples of SCFG rules in the automatic grammar.

5.3 Stat-XFER Results and Discussion

The results of our translation experiments with the Stat-XFER system are given in Table 5.1. The first row of the table — Xfer-no_gra — shows the results for a system configuration in which no grammar was used. In this configuration, only a bilingual lexicon is used, so the translation process of the system replicates that of a monotonic SMT decoder.[8]

Config.	**BLEU**	**NIST**	**METEOR**
Xfer-no_gra	0.2437	6.6295	0.5446
Xfer-man_gra	**0.2483**	**6.6558**	**0.5471**

Table 5.1: Translation results using the Stat-XFER system and our parallel treebank as training data.

Comparing the second row — Xfer-man_gra — we see the effect of using the nine-rule manual grammar on translation; improvements are seen across all three translation metrics (0.46% absolute increase in BLEU score; 1.89% relative increase). This confirms that, even with such a minimal grammar, we can improve translation accuracy by incorporating syntactic information. When translating the 1,500 test sentences, the nine rules in our manual grammar were applied a total of 509 times. A breakdown of how often each individual rule was used is presented in Figure 5.5.

From these numbers, we can see that rules (1)–(4), concerning local noun–adjective reordering, are applied over 62% of the time demonstrating how useful it is to model such translational divergences. There are many examples to be found

[8] A monotonic decoder is one in which no reordering model is included as a feature in the log-linear model (cf. section 2.2.3)

Rule	Freq.	Rule RHS		
(1)	126	[N^1 A^2]	::	[JJ^2 N^1]
(2)	30	[N^1 AP^2]	::	[$ADJP^2$ N^1]
(3)	152	[D^1 N^2 A^3]	::	[DT^1 JJ^3 N^2]
(4)	19	[D^1 N^2 AP^3]	::	[DT^1 $ADJP^3$ N^2]
(5)	56	[N^1 "de" N^2]	::	[N^2 N^1]
(6)	62	[D^1 N^2 "de" N^3]	::	[DT^1 N^3 N^2]
(7)	15	[A^1 C^2 A^3]	::	[JJ^1 CC^2 JJ^3]
(8)	4	[A^1 "," A^2 C^3 A^4]	::	[JJ^1 "," JJ^2 CC^3 JJ^4]
(9)	45	[ADV^1 A^2]	::	[RB^1 JJ^2]

Figure 5.5: Nine rule grammar right-hand sides with frequency information pertaining to how often each rule was applied during translation.

in the output translations of these rules being applied to give improved translations over the Xfer-no_gra configuration. Looking at the translation output in (5.1),[9] we see an example of rule (3) being applied successfully to capture the correct reordering of the French noun–adjective pair in the phrase *une avancée fondamentale* which was not captured by the configuration using no grammar.[10]

Src: une avancée fondamentale pour la protection des droits des citoyens

Ref: a fundamental step forward for the protection of citizen's rights

---

No_gra: a step | fundamental | to | the protection of citizen's rights

Man_gra: a basic step | for | the protection of citizen's rights

(5.1)

Further examples of the usefulness of noun–adjective reordering can be seen in (5.2), where rule (1) applies to correctly reorder the French *événement historique* as

[9]The vertical bars '|' in the examples indicate the boundaries of the segments used from the bilingual lexicon to build the translation hypothesis.

[10]This example is symptomatic of the drawbacks of the automatic evaluation measures that we touched upon previously (cf. Section 2.4.4). In the reference translation we have the phrase "a fundamental step". In the No_gra output, the word order is wrong — "a step fundamental" — but all the words in the reference are matched, so it achieves three unigram matches. In the Man_gra output, a valid translation is produced, but using alternative lexical choice to the reference: "a *basic* step". As a consequence, this translation has only two unigram matches for the translation of this phrase and ultimately it may cause the entire sentence to receive a lower score.

"historic event". The No_gra configuration carries out direct word-for-word translation and consequently gets the word order wrong.

Src:	c'est ce formidable événement historique qui...
Ref:	this fantastic historic event, which...
No_gra:	it is | this | great | event | historic | which...
Man_gra:	it is | this | great | historic event | which...

(5.2)

In addition to this, example (5.3) demonstrates the application of two rules in parallel to capture reordering of a noun and adjectival phrase. Rule (8) applies to capture the comma-separated adjectival phrase "administrative , fiscal and judicial", while rule (2) reorders this with the noun "structures". The No_gra configuration correctly captures a more local noun–adjective reordering with the phrase pair "structures administratives → administrative structures" but it fails to include the other adjectives in the phrase.

Src:	...renforcer ses structures administratives , fiscales et juridictionnelles
Ref:	...tighten up its administrative , fiscal and legal systems
No_gra:	...strengthen | its | administrative structures | , | tax | and | judicial
Man_gra:	...strengthen | its | administrative , fiscal and judicial structures

(5.3)

Finally, in examples (5.4) and (5.5) we see rules being applied which delete the French preposition *de* from the English translation and reorder the nouns in a noun phrase. We see rule (5) being applied twice in example (5.4) to translate *dispositifs de filtrage* and *systèmes de guide*, while example (5.5) shows rule (6) capturing a noun phrase including an article — "the crisis situation" — where the No_gra configuration carried out translation using two phrase pairs which split the French phrase *la situation de crise* and subsequently failed to capture the translational divergence as a result.

Src: . . .dispositifs de filtrage et aux systèmes de guide
Ref: . . .filtering systems and programme classification systems
No_gra: . . .devices | of | filtering | and to | systems | of | guide
Man_gra: . . .filter systems | and the | guidance systems (5.4)

Src: . . .la situation de crise au Pérou
Ref: . . .the crisis in Peru
No_gra: . . .the situation | of crisis | in Peru
Man_gra: . . .the crisis situation | in Peru (5.5)

Using even just this small grammar, we have demonstrated that improvements in translation quality can be made by employing SCFG rules in the system. In the following section we describe results from the experiments in which we used the automatic grammar extracted from our parallel treebank.

5.3.1 Automatically Derived Grammar: Results

Table 5.2 presents the results of the translation experiments in which we employed the automatically extracted grammar. We see from the third row of the table — Xfer-auto_gra — that we achieve even further improvements over the "no grammar" baseline using the automatically extracted grammar across all evaluation metrics (0.65% absolute increase in BLEU score; 2.67% relative increase). Even though they are not directly comparable due to the different rule set sizes, by comparing rows 2 and 3 we see that the automatic grammar performs slightly better than the manual grammar. While these improvements are not statistically significant, they are encouraging insofar as we have yet to determine the most appropriate way to automatically extract grammars. Despite this, using the technique described here, we have achieved comparable results to a manual grammar crafted specifically for the language pair and tagset in question, which is a time-consuming task.

Config.	BLEU	NIST	METEOR
Xfer-no_gra	0.2437	6.6295	0.5446
Xfer-man_gra	0.2483	6.6558	0.5471
Xfer-auto_gra	0.2502	**6.7087**	0.5506
Xfer-man+40_gra	**0.2510**	6.6804	**0.5606**

Table 5.2: Translation results using including the automatically extracted grammar.

In translating the 1,500 test sentences, rules from our automatic grammar were applied a total of 1,450 times, i.e. almost once per sentence. Of the 40 automatic rules, 2 were also found in the manual grammar. They correspond to the two most frequently used rules in Figure 5.5: rules (1) and (3). These rules were also among the most frequently applied rules from the automatic grammar, a summary of which is given in Figure 5.6.[11] Examples of these rules ((4) and (7) in Figure 5.5) being applied correctly — and exactly as they were applied in the same examples using the manual grammar — are shown below in (5.6) and (5.7).

Rule	Freq.	Rule RHS
(1)	257	["des" N^1] :: [N^1]
(2)	231	["les" N^1] :: [N^1]
(3)	175	["à" NP^1] :: ["to" NP^1]
(4)	173	[DT^1 N^2 JJ^3] :: [DT^1 JJ^3 N^2]
(5)	127	["l"' N^1] :: [N^1]
(6)	126	["la" N^1] :: [N^1]
(7)	110	[N^1 JJ^2] :: [JJ^2 N^1]

Figure 5.6: Most frequently applied rules from the automatic grammar.

[11] Only those rules from the automatic grammar which were applied more than 100 times during translation of the test sentences are shown.

Src: La seule instance européenne directement et démocratiquement élue

Ref: The sole european body to be directly and democratically elected

---

No_gra: | The only body | union | directly and democratically elected

Auto_gra: The only | european institution | directly and democratically elected

(5.6)

Src: C' est une avancée particulièrement importante pour les femmes

Ref: This is a particularly important advance for women

No_gra: It is | a step | particularly important | for women

Auto_gra: It is | a vital step | for women

(5.7)

Numerous examples of the rules in Figure 5.6 being applied to produce accurate translations can be found in the output translations. In example (5.8) we see rule (1) applied to correctly delete the French preposition *des* from the translation.[12]

Src: Il est inadmissible. . .que des personnes soient exclues de la vie sociale

Ref: We cannot accept. . .people being excluded from society

No_gra: It is unacceptable. . .that | of | people | are excluded from society

Auto_gra: It is unacceptable. . .that | people | are excluded from society

(5.8)

Similarly, rule (2) captures the deletion of the French definite article *les*. Such articles are commonly used in French and when translating into English, it is often acceptable to translate them in some cases and delete them in others. This presents a challenge for rules such as (2) which may over-apply. For instance, in example

[12]We will not compare the output of the Man_gra and Auto_gra configurations because it is generally the case that if a rule was applied using the Auto_gra configuration to produce a correct translation, and that rule did not exist in the Man_gra configuration, then the Man_gra configuration would produce the same output as the No_gra configuration and vice versa. We are simply highlighting here that when a rule exists and is applied, it helps to produce improved translation output over cases where it is not available.

(5.9) we see the rule correctly applying to remove the articles before "weapons" and "conflicts". Conversely, in example (5.10), we see the rule applying three times with contrasting effects. It first applies to incorrectly remove the clause initial article before "discrimination" but then applies twice more to correctly remove the unnecessary articles before "difficulties" and "women". When using no grammar in these examples, the article is always translated directly, in some cases word-for-word and in others as part of a larger phrase pair.

Src:	Les armes alimentent les conflits de par le monde .
Ref:	Arms fuel conflicts all over the world .
No_gra:	The | weapons | fuel | the | conflict | in the world .
Auto_gra:	Weapons | fuel | conflicts | in the world .

(5.9)

Src:	Les discriminations et les difficults auxquelles sont confrontes les femmes perdurent .
Ref:	The discrimination and difficulties women face unfortunately persist .
No_gra:	The discrimination | and |the difficulties | which | face the | women | continue .
Auto_gra:	Discrimination | and | difficulties | which | face | women | continue .

(5.10)

Following on from this, rules (5) and (6) behave similarly to rule (2) in that they model the deletion of morphological variants of the definite article. Finally, rule (3) describes a direct mapping between prepositional phrases.

The remaining rules in our automatic grammar were applied less frequently during translation. In fact, 12 of the remaining 33 rules[13] extracted were not used at all during translation. Those rules which did apply tended to model direct mappings with no translational divergences, or broader, more general relations. It was often the case that these rules produced similar translations to the No_gra configuration as they did not produce output that could not be modelled by direct word-for-word translation. For example, in (5.11) below, we see the application of a very general

[13]Excluding the 7 most frequently applied rules of Table 5.6.

rule mapping a source NP VP pair to a target NP VP pair. This rule was applied just a single time during the translation of the test set. Additionally, examples (5.12)[14] and (5.13) show the application of rules which mapped directly between different variants of a prepositional phrase. These rules were applied 34 and 9 times respectively.

Rule:	SENT :: S → [NP^1 VP^2] :: [NP^1 VP^2]
Src:	cette procédure n' a pas encore été entamée
Ref:	no such proceedings have been initiated as yet
No_gra:	this procedure | has not yet started
Auto_gra:	| the process has not yet started |

(5.11)

Rule:	PP :: PP → ["en" NP^1] :: ["in" NP^1]
Src:	. . .d' être le plus rapidement possible <u>en mesure</u> d' apporter les modifications nécessaires
Ref:	. . .as quickly as possible , start making the necessary changes
No_gra:	. . .be | as soon as possible | <u>can provide</u> | the necessary changes
Auto_gra:	. . .be | quickly as possible | <u>in the position</u> | to | make the necessary changes

(5.12)

Rule:	PP :: PP → ["de" NP^1] :: ["of" NP^1]
Src:	Les propositions <u>de M.</u> Gil-Robles sur la coopération renforcée. . .
Ref:	<u>Mr Gil-Robles '</u> proposals on reinforced cooperation. . .
No_gra:	The proposals | <u>of | Mr</u> | Gil-Robles on close cooperation. . .
Auto_gra:	The proposals | <u>of Mr</u> | Gil-Robles on close cooperation. . .

(5.13)

Looking back at Table 5.2 (p.131), in row 4 we see the translation results for a Stat-XFER system in which we combined the manual and automatic grammars.

[14]This example also demonstrates the rule VP::VP → [V NP] :: [VB NP] being applied to translate the French VP *apporter les modifications nécessaires* and "make the necessary changes".

This configuration amounted to the addition of 7 new SCFG rules from the manual grammar to the automatic grammar (2 of the manual rules were also found in the automatic grammar). As expected given our previous results, this configuration improved significantly over the No_gra baseline. However, when compared with the Auto_gra configuration, we see an insignificant improvement in BLEU, an improvement in METEOR but a drop in NIST score. These results suggest that the 7 new rules did not provide much benefit over what was already present in the manual grammar. This is confirmed upon finding that the rules from this combined grammar were applied 1,410 times when translating the 1,500 test sentences, as opposed to 1,450 times for the Auto_gra configuration. Furthermore, we noted earlier than there were two rules in common between the manual and automatic grammars. These are the two most frequent rules applied from the manual grammar in the Man_gra MT system configuration — rules (1) and (3) in Figure 5.5 — and account for 54.62% of all rules applied during translation with the Man_gra configuration. Thus, the novel SCFG rules we introduced when combining the manual rules with the automatic grammar were less useful and ultimately did not enhance the grammar significantly enough to lead to substantial improvements in translation accuracy.

5.4 Phrase-Based Translation Experiments

In the interest of completeness, we carried out a set of PB-SMT experiments using the same data and experimental setup as the other experiments in this chapter. As in Chapter 4, we built our phrase-based systems using Moses for phrase extraction and decoding. A 5-gram language model was built using the SRI language modelling toolkit. Minimum Error-Rate Training was carried out, optimising parameters on the BLEU metric and, finally, translations were evaluated automatically using BLEU, NIST and METEOR. Three system configurations were evaluated in total using the direct combination approach described in sections 4.2.2 and 4.2.3 respectively. They were: Baseline phrase pairs only; Syntax-based phrase pairs only and

Baseline and Syntax, direct combination. The results of these experiments are given in Table 5.3.

Config.	BLEU	NIST	METEOR
Baseline	0.3115	7.4816	**0.6087**
Baseline+Syntax	**0.3116**	**7.4985**	0.6076
Syntax_only	0.2793	6.9982	0.5733

Table 5.3: Results of PB-SMT experiments using the larger English–French data set.

Our findings here differ from those of Chapter 4 in that we do not see a significant improvement in translation performance when supplementing the baseline model with syntax-based phrase pairs from the parallel treebank. In section 4.2.6, we demonstrated that the influence of the syntax-based phrase pairs in the combined model decreased as the size of the training set grew (to a maximum training set size of approximately 730,000 sentence pairs). Furthermore, we suggested that if the size of the training set were to continue to increase, we might see the influence of the syntax-based phrase pairs diminish completely. As we are using more than 1,250,000 sentence pairs in these experiments — almost twice as many as the largest experiments conducted previously — our aforementioned assumptions are confirmed given these findings.

However, upon examining the extracted phrase tables further we discovered that the size of the training set is not the only factor at play in reducing the influence of the syntax-based phrase pairs. We observed that there are 9.03 times more baseline phrase pairs than syntax-based phrase pairs for this data set. This is interesting as there were only 3.84 times as many baseline phrase pairs given the data set in section 4.2. In fact, there were fewer syntax-based phrase pairs extracted from the larger data set in this chapter than there were from the data set in section 4.2, which was almost half the size (the number of baseline phrase pairs increased proportionally). The exact figures are shown in Table 5.4.

Investigating this further, we found the French parses in these experiments to

Language	Resource	#Phrases	#Training Pairs
En-Es (Sec.4.2)	Baseline	24,708,527	729,891
	Syntax	6,432,771	
En-Fr	Baseline	47,169,818	1,261,556
	Syntax	5,218,370	

Table 5.4: A comparison of the number of syntax-based phrase pairs extracted from differing data sets.

be relatively flat compared to the Spanish parses of section 4.2.[15] Looking at Table 5.5, we can analyse the parallel treebanks further.

	French	Spanish
Ave. Sentence Length	30.13	28.99
Ave. Nodes per Tree	44.50	48.25
Ave. %Linked nodes	59.3%	67.88%

Table 5.5: Comparing the French and Spanish sides of their respective parallel treebanks.

We see the average length of the French sentences is 30.1 tokens which gives rise to an average of 44.5 constituent nodes per tree when parsed. Of these 44.5 nodes, approximately 59.3% are aligned on average during parallel treebank generation. Comparing this to the data set of section 4.2, there are 8.42% more nodes in the Spanish trees than the French tree and of these nodes a further 8.57% are aligned in the English–Spanish parallel treebank.[16] This ultimately results in flatter French trees, reducing the number of available sub-tree alignments and subsequently the number of extractable phrase pairs from the parallel trees.

We illustrate this further with an example from our data. Comparing the English–Spanish and English–French tree pair fragments[17] — Figures 5.7 and 5.8 respectively — we can see the aforementioned differences more clearly. The English sides of each tree pair, which are identical as they come from the same portion of

[15]The two data sets are roughly comparable as both were derived from the Europarl corpus. The English side of the English–Spanish parallel corpus is a subset of the English side of the larger English–French corpus. For the most part, the English parses produced in the respective parallel treebanks are identical.

[16]To summarise, that is 59.5% of 44.5 French nodes aligned compared to 67.9% of 48.3 Spanish nodes aligned.

[17]The full trees are provided in Appendix D.

the parallel data, have 5 nodes. For the English–Spanish tree pair in Figure 5.7, all 5 English nodes are aligned given 15 Spanish nodes. The Spanish tree is also quite hierarchical and right-branching in nature, with a node depth of seven. Conversely, for the English–French tree pair in Figure 5.8, only 3 of the English nodes are aligned to the French tree which has 9 nodes in total: 40% fewer nodes than in the Spanish tree. We can also see that the French tree is relatively flat, with a node depth of two. There are essentially two non-terminal nodes with the remainder being pre-terminals descending from them. These factors have impacted on the number of alignment options available to the sub-tree aligner and consequently on the number of extractable phrase pairs. It is a combination of this and the larger training set that has contributed to the diminished influence of the syntax-based phrase pairs in the combined models.

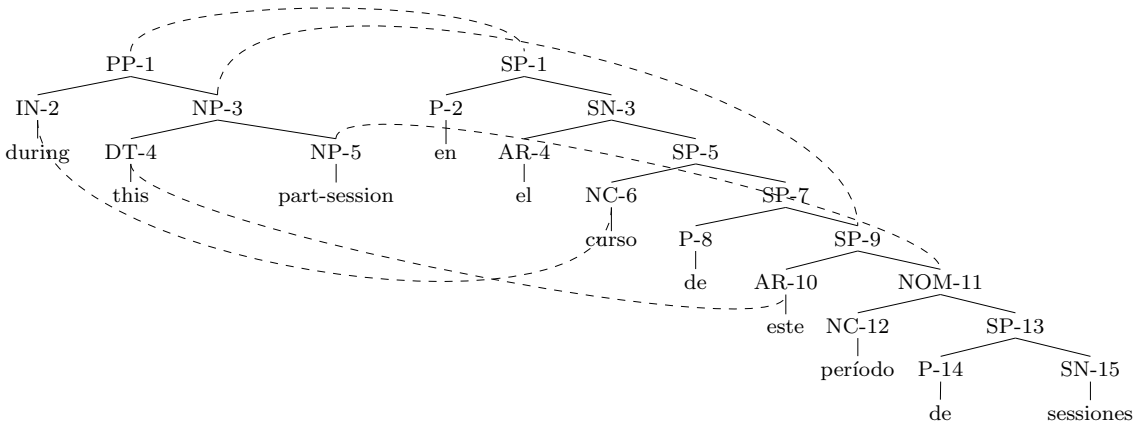


Figure 5.7: Example English–Spanish tree pair and alignments: All English nodes are aligned to the hierarchical Spanish tree.

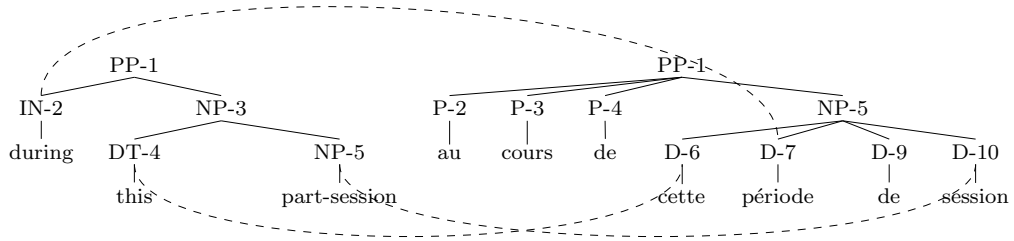


Figure 5.8: Example English–French tree pair and alignments: French tree is quite flat and not all English nodes are aligned.

This is further reflected in the translation performance of the remaining con-

figuration in Table 5.3. In row 3, we see the Syntax_only configuration acheives scores significantly lower than the baseline across all metrics (3.23% absolute drop in BLEU; 11.56% relative). This drop in translation performance is considerably larger than in the experiments of section 4.2.2 where there was only a 5.96% relative difference in BLEU score.

5.5 Summary

As discussed, the potential of automatically generated parallel treebanks extends beyond the extraction of string-based translation pairs. Annotated phrase tables and transfer rules — combined as a synchronous context-free grammar and extracted from parallel treebanks — can be exploited to improve the translation accuracy of a syntax-based MT system. We have shown competitive translation performance when using an automatically extracted set of SCFG rules in place of a manually crafted grammar. This is particularly encouraging as one of the challenges of syntax-based MT is deciding how to refine unwieldy grammars, remove redundancy and ultimately improve efficiency. Thus, what has been demonstrated here serves as a solid foundation for further investigation into the exploitation of our parallel-treebanks in syntax-based MT

In terms of PB-SMT, as we suggested may be the case in section 4.6 of the previous chapter, supplementing the baseline model with syntax-based phrase pairs was ineffective given the much larger data set used here. However, the structure of the parallel trees we used also had a significant impact on this. Our French parses were much flatter than in previous parallel treebanks and this had a substantial impact on the number of extractable phrase pairs. While it is beyond the scope of this thesis, we believe there is value in investigating whether the use of monolingual parsers which produce more hierarchical structures is preferable, or whether pairs of monolingual parsers should be chosen whose resulting structures are more closely related.

Chapter 6

Conclusions

Phrase-based SMT, while the state-of-the-art in MT, is driven solely by statistics and makes no use of linguistic information during the translation process. Syntactic-information has been shown to be useful when incorporated into PB-SMT, and this suggests there is potential in pursuing fully syntax-based models. However, the development of such models has been inhibited by the lack of available syntactically annotated training resources. In this thesis, we addressed four main research questions, outlined in Chapter 1, relating to these issues:

> **RQ1:** Can we develop a method to facilitate the automatic generation of large-scale high-quality parallel treebanks for use in MT?
>
> **RQ2:** Can syntactically motivated phrase pairs extracted from a parallel treebank be exploited to improve phrase-based SMT?
>
> **RQ3:** What other features of the phrase-based model can be enhanced by exploiting the information encoded in parallel treebanks?
>
> **RQ4:** To what extent are our automatically generated parallel treebanks useful in syntax-based MT?

In terms of **RQ1**, in Chapter 3 we presented a novel algorithm for the induction of sub-sentential alignments between tree pairs, thus giving ourselves the ability to fully

automate the process of building parallel treebanks. We described the algorithm in detail and performed intrinsic, extrinsic and manual analysis of the quality of the resulting treebanks. From this evaluation we have drawn the following conclusions:

- we have developed a viable solution to the challenge of sub-tree alignment and, consequently, the automatic generation of large-scale parallel treebanks;
- the algorithm is language pair-independent and has demonstrated its effectiveness across several language pairs, including non-European languages.

Following this, in Chapter 4 we investigated our hypothesis that parallel treebanks have use beyond syntax-based MT by addressing **RQ2** and **RQ3**. Regarding **RQ2**, we exploited parallel treebanks directly by using them to supplement the translation models of a large number of PB-SMT systems. This was done by extracting a set of syntax-based phrase pairs directly from parallel treebanks and using various techniques to combine them with baseline PB-SMT phrase pairs. Moving on to **RQ3**, we carried out further experiments aimed at discovering alternative ways in which the information encoded in parallel treebanks could be exploited to enhance the PB-SMT pipeline. In addition to these experiments, we investigated the possibility of using our sub-tree alignment algorithm to align dependency structures for phrase extraction. Our principal findings from this body of work were as follows:

- significant improvements were achieved in the translation performance of a baseline PB-SMT system by supplementing the translation model with syntax-based phrase pairs extracted from a parallel treebank; the parallel treebank was automatically generated over the same parallel data on which the baseline system was trained. The direct approach to phrase combination performs optimally;
- while this hypothesis holds across various data sets and language pairs, we note that the complementary effect of the parallel treebank data diminishes as the training set size increases to the point where supplementing the model becomes ineffective;

- this approach to supplementing PB-SMT models may be best employed in cases where limited training data is available or where resources dictate the necessity for smaller phrase tables;

- the quality of the word alignments encoded in the parallel treebanks is somewhat inhibiting their ability to improve translation accuracy still further. Improvements to these alignments is key to future gains;

- the parsing formalism used to build the parallel trees has a significant effect on the quality of the resulting treebank and set of phrase pairs. The more hierarchical the parse, the more nodes in the trees, the more sub-tree alignments. We found this to be a desirable property;

- it is quite difficult to improve upon the PB-SMT pipeline by making minor adjustments to certain features, such as lexical weighting;

- it is best to use refined statistical word alignments rather than parallel treebank word alignments to seed PB-SMT phrase extraction. However, given a parallel corpus and a parallel treebank, we can use all information at our disposal — statistical word alignments, parallel treebank word alignments and syntax-based phrase pairs — to generate concise translation models (up to 56% smaller than pure baseline models) that achieve comparable translation performance to much larger baseline models;

- we can successfully align bracketed structures produced by a formal conversion of dependency representations and extract phrase pairs for PB-SMT.

Finally, in terms of **RQ4**, we deployed a parallel treebank as a training resource for a syntax-based, Stat-XFER system in Chapter 5. We extracted a bilingual lexicon directly from the treebank and used encoded word alignments to seed the extraction of a synchronous context-free grammar. Comparing a number of MT systems, we drew the following conclusions:

- translation quality of a syntax-based MT system can indeed be improved by adding deeper syntactic knowledge into the process as demonstrated by the use of a manually-crafted grammar;
- using a very small percentage of transfer rules extracted automatically from a parallel treebank gives rise to comparable translation performance when compared to a manually-crafted grammar;
- the main challenge facing syntax-based MT going forwards is how to extract an efficient, refined grammar from a parallel treebank given the millions of extractable rules.

A final trend we observed in the majority of translation experiments carried out in this thesis was the inconsistency in scores across the automatic evaluation metrics. It was often the case that one metric would report a significant improvement over the baseline, while another would report an insignificant drop in performance. As a consequence of these findings, we believe that despite their utility, the automatic metrics do not necessarily facilitate a definitive analysis of translation quality and some degree of human judgement is still required. This was especially the case in this thesis, where many of the observed differences between systems were small and, consequently, the automatic metrics were unable to tease them apart. Until such a time as research into automatic evaluation of translation quality can demonstrate *consistent* correlation with manual assessment, MT research such as that presented in this thesis will not be able to flourish.

6.1 Future Work

Drawing from the open research questions that have arisen based on our experiments throughout the course of this dissertation, we now present some potential avenues for future research which we believe warrant exploration.

In terms of sub-tree alignments, in section 3.3.3 we saw a conflict between the

score2 and *span1* options. Identifying the source of this conflict may provide useful information which could be applied in the development of a single, optimal configuration for the alignment algorithm.

We noted throughout this thesis that the weakest facet of the alignment algorithm was its induction of word-level alignments, which had an adverse effect on many of the MT tasks we carried out. There are a number of ways in which this issue could be addressed, for example, by using specific anchor alignments between certain troublesome tokens such as function words and punctuation. This would prevent misalignment between these types of words and act as a guide for the selection process by *a priori* ruling out a number of ill-formed hypotheses.

In section 5.4, we saw that the structure of the parse trees in the parallel treebank had a significant effect on sub-tree alignment and subsequent tasks in which the treebanks were exploited. Examining this further — for instance, between language pairs with rich syntactic-annotation resources, such as treebanks and parsers — could provide us with deeper insight as to the type of trees (and tree pairs) most suited to alignment and subsequent tasks. Furthermore, a qualitative analysis of the effect of parser errors on alignment would be useful in indicating whether it would be worthwhile (in terms of resulting quality) spending time to resolve such errors prior to sub-tree alignment. Without such an analysis, the extent to which the propagation of parsing and alignment errors carries over to MT is unclear.

We discovered in section 4.3.1 that using treebank-based word alignments to create a refined word alignment for phrase extraction can lead to a significantly reduced phrase table without any loss of translation accuracy. While this was only observed under a single experimental condition in this thesis, we believe further exploration as to the extent to which this process can be applied is merited. Additionally, more creative ways of combining the various word alignments (e.g. statistical source–target and treebank-based alignments) at our disposal could also be investigated for phrase extraction.

Finally, the exploitation of our automatically generated parallel treebanks in

syntax-based MT was discussed in Chapter 5. We used the word alignments from a parallel treebank to seed the grammar extraction process of the Stat-XFER system. The next logical step following these experiments, is to extract a grammar directly from our parallel treebanks using both the word- and phrase-level alignments. However, the question still remains for syntax-based MT in general as to how we can efficiently employ large-scale automatically extracted grammars to improve overall translation quality.

Appendices

Appendix A

English Parser Tag Set

Tables A.1 and A.2 show the part-of-speech (POS) tags and phrase labels respectively for the Berkeley parser (Petrov and Klein, 2007) trained on the Penn-II Treebank (Marcus et al., 1994) for English, as used in Chapter 5.

POS Tag	**Tag Description**
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
N	Noun, singular
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to

Continued on next page

POS Tag	Tag Description
UH	Interjection
VB	Verb, base form
VBD	Verb, preterite
VBG	Verb, present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person present singular
VBZ	Verb, 3rd person present singular
WDT	WH-determiner
WP	WH-pronoun
WP$	possessive WH-pronoun
WRB	WH-adverb
-LRB-	Left bracket
-RRB-	Right bracket
“	Open quotation
”	Close quotation
,	Comma
.	Full stop
:	Colon

Table A.1: Tag labels in the grammar of the English parser.

Phrase Label	**Phrase Description**
ADJP	Adjectival phrase
ADVP	Adverbial phrase
CONJP	Conjunction phrase
FRAG	Fragment
INTJ	Interjection (∼POS tag UH)
LST	List item marker, including surrounding punctuation
NAC	Not a constituent
NP	Noun phrase
NX	Noun phrase head (N-bar)
PP	Prepositional phrase
PRN	Parenthetical
QP	Quantifier phrase
RRC	Reduced relative clause
S	Declarative clause (sentence)
SBAR	Subordinate clause
SBARQ	Direct question
SINV	Inverted declarative sentence
SQ	Inverted yes/no question
UCP	Unlike coordinated phrase
VP	Verb phrase
WHADJP	WH-adjectival phrase
WHADVP	WH-adverbial phrase
WHNP	WH-noun phrase
WHPP	WH-prepositional phrase
X	Unknown, uncertain or unbracketable

Table A.2: Phrase labels in the grammar of the English parser.

Appendix B

French Parser Tag Set

Tables B.1 and B.2 show the POS tags and phrase labels respectively for the Berkeley parser trained on the Modified French Treebank (Schluter and van Genabith, 2007) for French, as used in Chapter 5.

POS Tag	**Tag Description**
A	Ajective
ADV	Adverb
C	Coordinating conjunction
CL	Clitic pronoun (weak)
D	Determiner
ET	Foreign word
I	Interjection
N	Noun
P	Preposition
PC	Prepositional clitic
PREF	Prefix
PRO	Pronoun (strong)
V	Verb
X	Unknown
-LRB-	Left bracket
-RRB-	Right bracket
,	Comma
.	Full stop
:	Colon
”	Quotation

Table B.1: Tag labels in the grammar of the French parser.

Phrase Label	**Phrase Description**
AP	Adjectival phrase
AdP	Adverbial phrase
NP	Noun phrase
PP	Prepositional phrase
SENT	Sentential clause
Sint	Internal clause
Srel	Relative clause
Ssub	Subordinate clause
VN	Verb nucleus
VPinf	Verb phrase, infinitive
VPpart	Verb phrase, participle
X	Unknown

Table B.2: Phrase labels in the grammar of the French parser.

Appendix C

40-Rule Automatic Grammar

The full 40 rule automatic grammar used in the syntax-based MT experiments of section 5.2.2 is given below in Table C.1.

Rule	**Src. Tag**		**Tgt. Tag**		**Src. RHS**		**Tgt. RHS**
(1)	NP	::	NP	→	[D^1 NP^2]	::	[DT^1 NP^2]
(2)	VP	::	VP	→	[V^1 NP^2]	::	[VB^1 NP^2]
(3)	NP	::	NP	→	["l"' N^1]	::	["the" N^1]
(4)	NP	::	NP	→	["l"' N^1]	::	[N^1]
(5)	PP	::	PP	→	["à" NP^1]	::	["to" NP^1]
(6)	NP	::	NP	→	["des" N^1]	::	[N^1]
(7)	PP	::	PP	→	["de" NP^1]	::	["of" NP^1]
(8)	NP	::	NP	→	[A^1 N^2]	::	[JJ^1 N^2]
(9)	NP	::	NP	→	["les" N^1]	::	[N^1]
(10)	NP	::	WHNP	→	[PRO^1]	::	[WP^1]
(11)	NP	::	NP	→	["le" N^1]	::	["the" N^1]
(12)	NP	::	NP	→	[D^1 N^2 A^3]	::	[DT^1 JJ^3 N^2]
(13)	VP	::	VP	→	[V^1]	::	[VBN^1]
(14)	PP	::	PP	→	["to" NP^1]	::	["to" NP^1]
(15)	NP	::	NP	→	[PRO^1]	::	[PRP^1]
(16)	NP	::	NP	→	[PRO^1 N^2]	::	[PRP^1 N^2]
(17)	NP	::	NP	→	[D^1]	::	[DT^1]

Continued on next page

Rule	Src. Tag		Tgt. Tag		Src. RHS		Tgt. RHS
(18)	NP	::	NP	→	["la" N^1]	::	["the" N^1]
(19)	NP	::	NP	→	["la" N^1]	::	[N^1]
(20)	NP	::	NP	→	[D^1 NP^2]	::	[DT^1 NX^2]
(21)	NP	::	WHNP	→	[D^1]	::	[WDT^1]
(22)	NP	::	NP	→	[N^1 JJ^2]	::	[JJ^2 N^1]
(23)	S	::	S	→	[NP^1 VP^2]	::	[NP^1 VP^2]
(24)	NP	::	NP	→	["la" NP^1]	::	["the" NP^1]
(25)	AP	::	ADJP	→	[A^1]	::	[JJ^1]
(26)	VP	::	VP	→	[V^1]	::	[VB^1]
(27)	NP	::	NP	→	[D^1 N^2]	::	[DT^1 N^2]
(28)	PP	::	PP	→	["dans" NP^1]	::	["in" NP^1]
(29)	AdP	::	ADVP	→	[ADV^1]	::	[RB^1]
(30)	PP	::	PP	→	["du" NP^1]	::	["of" NP^1]
(31)	AP	::	ADJP	→	[ADV^1 A^2]	::	[RB^1 JJ^2]
(32)	PP	::	PP	→	[P^1 NP^2]	::	[IN^1 NP^2]
(33)	NP	::	NP	→	[N^1]	::	[CD^1]
(34)	NP	::	NP	→	[D^1 N^2 PP^3]	::	[DT^1 N^2 PP^3]
(35)	NP	::	NP	→	[N^1]	::	["the" N^1]
(36)	NP	::	NP	→	[N^1]	::	[N^1]
(37)	NP	::	NP	→	["ce" N^1]	::	["this" N^1]
(38)	PP	::	PP	→	["en" N^1]	::	["in" N^1]
(39)	AdP	::	WHADVP	→	[ADV^1]	::	[WRB^1]
(40)	PP	::	PP	→	["des" N^1]	::	["of" N^1]

Table C.1: Full 40 rule grammar for French–English

Appendix D

Full Parse Trees

The following three figures illustrate the full trees for the tree fragments in the examples of Figures 5.7 and 5.8 on 138. The trees in Figures D.1 and D.2 (on pages 155 and 156 respectively) represent the full English–French parallel treebank entry, while the trees in Figures D.1 and D.3 (on pages 155 and 157 respectively) represent the English–Spanish parallel treebank entry. As we mentioned previously, the same English parse tree is found in both parallel treebanks.

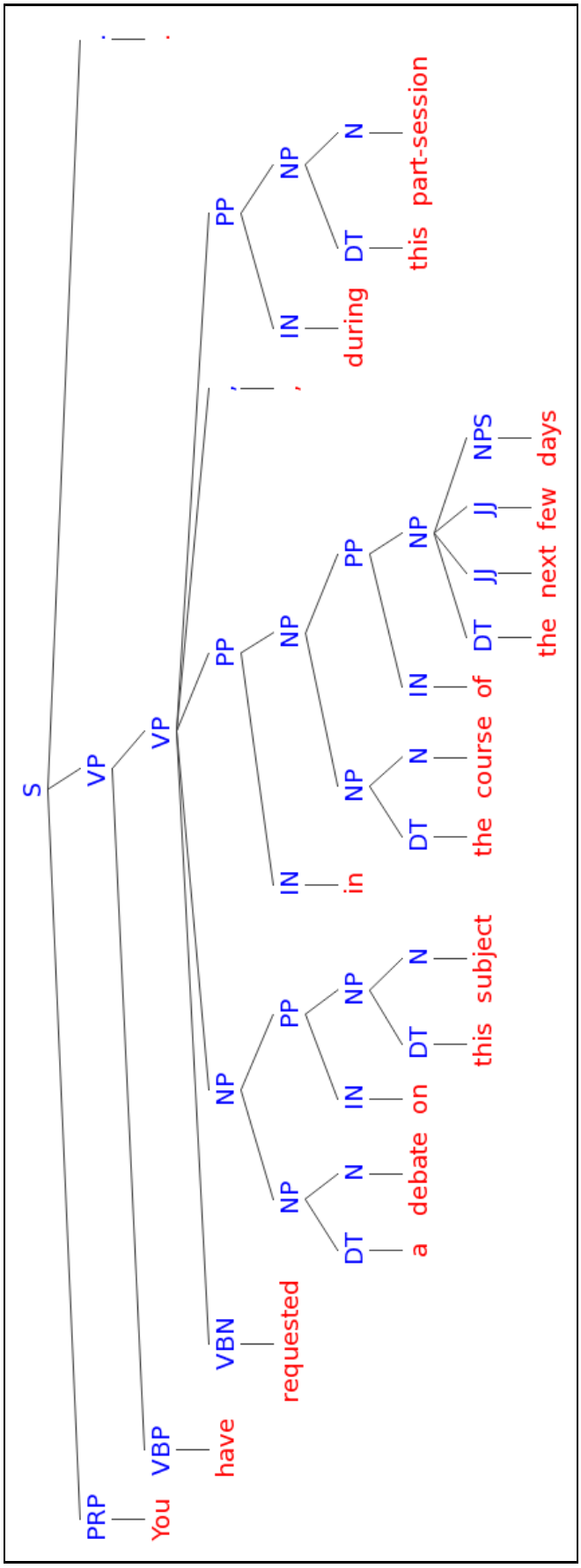


Figure D.1: Full English parse tree from Figures 5.7 and 5.8.

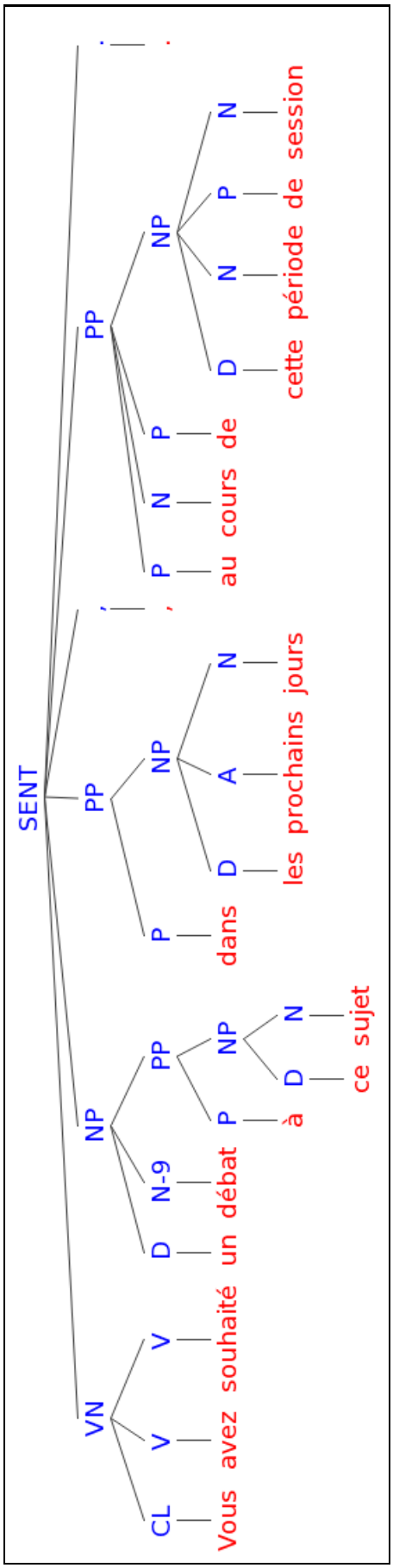


Figure D.2: Full French parse tree from Figure 5.8.

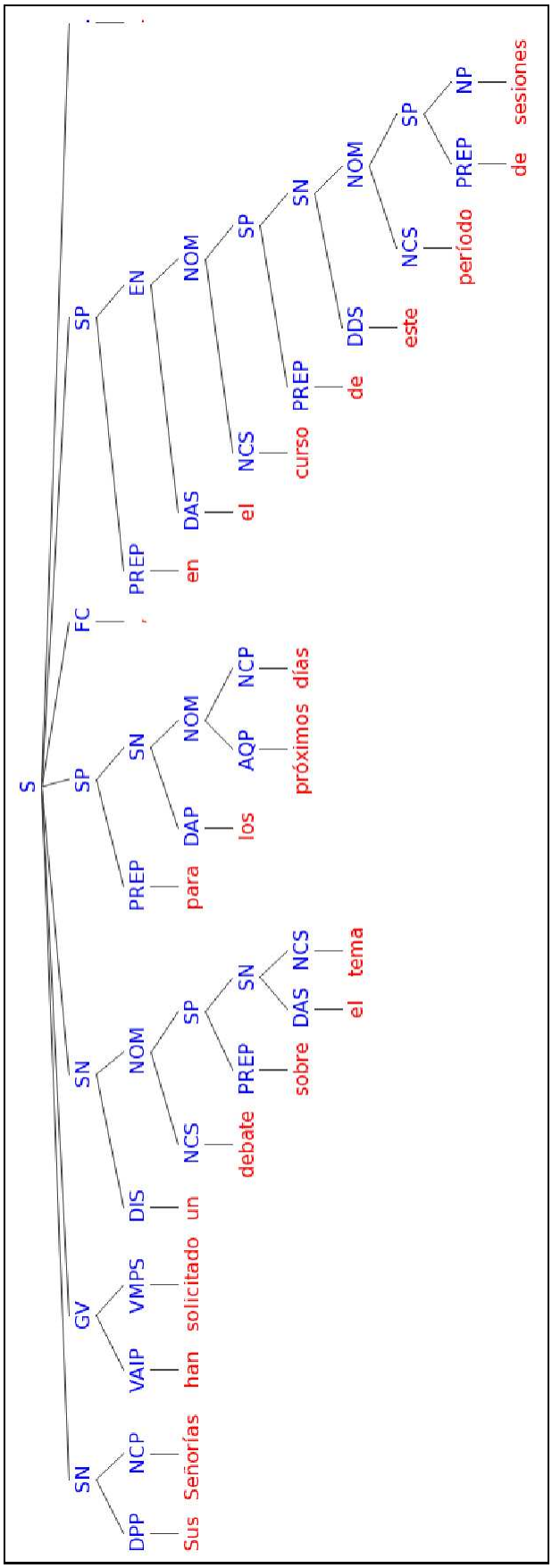


Figure D.3: Full Spanish parse tree from Figure 5.7.

Bibliography

Abeillé, A., Clement, L., and Kinyon, A. 2000. Building a treebank for French. *In* In Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC), Athens, Greece.

Ahrenberg, L. 2007. LinES: An English–Swedish Parallel Treebank. *In* Proceedings of the 16th Nordic Conference of Computational Linguistics (NOLADIA'07), pp. 270–274, Tartu, Estonia.

Ambati, V. and Lavie, A. 2008. Improving Syntax-Driven Translation Models by Re-structuring Divergent and Non-isomorphic Parse Tree Structures. *In* Proceedings of the 8th Conference of the Association for Machine Translation in the Americas (AMTA), pp. 235–244, Waikiki, HI.

Ambati, V., Lavie, A., and Carbonell, J. 2009. Extraction of Syntactic Translation Models from Parallel Data using Syntax from Source and Target Languages. *In* Proceedings of Machine Translation Summit XII, pp. 190–197, Ottawa, Canada.

Banerjee, S. and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *In* Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-05), pp. 65–72, Ann Arbor, MI.

Bikel, D. 2002. Design of a Multi-lingual, parallel-processing statistical parsing engine. *In* Proceedings of the Human Language Technology Conference (HLT), pp. 24–27, San Diego, CA.

R. Bod, R. Scha, and K. Sima'an (eds.) 2003. Data-Oriented Parsing. Stanford CA: CSLI Publications.

Bojar, O. and Hajič, J. 2008. Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. *In* Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, OH.

Bojar, O., Mareček, D., Novák, V., Popel, M., Ptáček, J., Rouš, J., and Žabokrtský, Z. 2009. English-Czech MT in 2008. *In* Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 125–129, Athens, Greece.

Bonnema, R. and Scha, R. 2003. Reconsidering the Probability Model for DOP, pp. 25–42. *In* R. Bod, R. Scha, and K. Sima'an (eds.), Data-Oriented Parsing. Stanford CA: CSLI Publications.

Bourigault, D., Fabre, C., Fréot, C., Jacques, M.-P., and Ozdowska, S. 2005. Syntex, analyseur syntaxique de corpus. *In* Actes des 12èmes journes sur le Traitement Automatique des Langues Naturelles, Dourdan, France.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. 2002. The TIGER Treebank. *In* Proceedings of the Workshop on Treebanks and Linguistic Theories, pp. 27–41, Sozopol, Bulgaria.

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

Brown, P. F., Della-Pietra, V. J., Della-Pietra, S. A., and Mercer,

R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19:263–311.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. *In* Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 1–28, Athens, Greece.

Callison-Burch, C., Osborne, M., and Koehn, P. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. *In* Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 249–256, Trento, Italy.

Carpuat, M. and Wu, D. 2007. How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation. *In* Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07), pp. 43–52, Skövde, Sweden.

Chen, S. F. and Goodman, J. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. *In* 34th Annual Meeting of the Association for Computational Linguistics (ACL'96), pp. 310–318.

Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. *In* 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 263–270, Ann Arbor, MI.

Chiang, D. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics* 33:201–228.

Chiang, D., DeNeefe, S., Chan, Y. S., , and Ng, H. T. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. *In* Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP), pp. 610–619, Waikiki, HI.

Chiang, D., Knight, K., and Wang, W. 2009. 11,001 New Features for Statistical Machine Translation. *In* Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 218–226, Boulder, CO.

Chiao, Y.-C., Kraif, O., Laurent, D., Nguyen, T. M. H., Semmar, N., Stuck, F., Véronis, J., and Zaghouani, W. 2006. Evaluation of multilingual text alignment systems: the ARCADE II project. *In* Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06), pp. 1975–1978, Genoa, Italy.

Chrupała, G. and van Genabith, J. 2006. Using Machine-Learning to Assign Function Labels to Parser Output for Spanish. *In* 44th Annual Meeting of the Association for Computational Linguistics (ACL'06), pp. 136–143, Sydney, Australia.

Civit, M. and Martí, M. A. 2004. Building Cast3LB: A Spanish Treebank. *Research on Language and Computation* 2(4):549–574.

Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., and Kuboň, V. 2004. Prague Czech-English Dependency Treebank. Syntactically Annotated Resources for Machine Translation. *In* Proceedings of the 4th International Conference on Language Resources and Evaluation, pp. 1597–1600, Lisbon, Portugal.

Collins, M., Koehn, P., and Kučerová, I. 2005. Clause Restructuring for Statistical Machine Translation. *In* 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 531–540, Ann Arbor, MI.

Dempster, A., Laird, N., and Rubin, D. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B* 39:1–38.

DeNero, J. and Klein, D. 2007. Tailoring Word Alignments to Syntactic Ma-

chine Translation. *In* 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), pp. 17–24, Prague, Czech Republic.

Deng, Y. and Byrne, W. 2008. HMM Word and Phrase Alignment for Statistical Machine Translation. *IEEE Transactions on Audio, Speech, and Language Processing* 16:494–507.

Doddington, G. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. *In* Human Language Technology: Notebook Proceedings, pp. 128–132, San Diego, CA.

Eck, M., Vogel, S., and Waibel, A. 2005. Low Cost Portability for Statistical Machine Translation Based on *n*-gram Coverage. *In* Machine Translation Summit X, pp. 227–234, Phuket, Thailand.

Eisner, J. 2003. Learning Non-Isomorphic Tree Mappings for Machine Translation. *In* 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), Companion Volume, pp. 205–208, Sapporo, Japan.

Frank, A. 1999. LFG-based syntactic transfer from English to French with the Xerox Translation Environment. *In* ESSLLI'99 Summer School, Utrecht, The Netherlands.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. *In* 44th Annual Meeting of the Association for Computational Linguistics, pp. 961–968, Sydney, Australia.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. 2004. What's in a Translation Rule? *In* HLT-NAACL 2004: Main Proceedings, pp. 273–280, Boston, MA.

Galron, D., Penkale, S., Way, A., and Melamed, I. D. 2009. Accuracy-based scoring for DOT: Towards direct error minimization for Data-Oriented

Translation. *In* Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 371–380, Singapore.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. 2001. Fast Decoding and Optimal Decoding for Machine Translation. *In* Proceedings of the Joint Meeting of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01) and the 10th Meeting of the European Association for Computational Linguistics, pp. 228–235, Toulouse, France.

Groves, D. 2007. Hybrid Data-Driven Models of Machine Translation. PhD thesis, Dublin City University, Dublin, Ireland.

Groves, D., Hearne, M., and Way, A. 2004. Robust Sub-Sentential Alignment of Phrase-Structure Trees. *In* Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), pp. 1072–1078, Geneva, Switzerland.

Gustafson-Čapková, S., Samuelsson, Y., and Volk, M. 2007. SMULTRON - The Stockholm MULtilingual parallel TReebank. www.ling.su.se/dali/research/smultron/index.

Han, C., Han, N.-R., Ko, E.-S., and Palmer, M. 2002. Development and Evaluation of a Korean Treebank and its Application to NLP. *In* Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC '02), pp. 1635–1642, Canary Islands, Spain.

Hanneman, G., Ambati, V., Clark, J. H., Parlikar, A., and Lavie, A. 2009. An improved statistical transfer system for French-English machine translation. *In* Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 140–144, Athens, Greece.

Hanneman, G., Huber, E., Agarwal, A., Ambati, V., Parlikar, A., Peterson, E., and Lavie, A. 2008. Statistical transfer systems for French-English and German-English machine translation. *In* Proceedings of the Third Workshop on Statistical Machine Translation, pp. 163–166, Columbus, OH.

Hanneman, G. and Lavie, A. 2009. Decoding with Syntactic and Non-Syntactic Phrases in a Syntax-Based Machine Translation System. *In* Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3), pp. 1–9, Boulder, CO.

Hansen-Schirra, S., Neumann, S., and Vela, M. 2006. Multi-dimensional Annotation and Alignment in an English-German Translation Corpus. *In* Proceedings of the Workshop on Multi-dimensional Markup in Natural Language Processing (NLPXML-2006) at EACL, pp. 35–42, Trento, Italy.

Haque, R., Naskar, S., Ma, Y., and Way, A. 2009a. Using Supertags as Source Language Context in SMT. *In* Proceedings of the 13th Annual Meeting of the European Association for Machine Translation (EAMT-09), pp. 234–241, Barcelona, Spain.

Haque, R., Naskar, S. K., van den Bosch, A., and Way, A. 2009b. Dependency Relations as Source Context in Phrase-Based SMT. *In* Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC), p. (to appear).

Hassan, H., Sima'an, K., and Way, A. 2007. Supertagged Phrase-based Statistical Machine Translation. *In* 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), pp. 288–295, Prague, Czech Republic.

Hassan, H., Sima'an, K., and Way, A. 2009. A syntactified direct translation model with linear-time decoding. *In* Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 1182–1191, Singapore.

He, Y. and Way, A. 2009. Improving the Objective Function in Minimum Error Rate Training. *In* Proceedings of Machine Translation Summit XII, pp. 238–245, Ottawa, Canada.

Hearne, M. 2005. Data-Oriented Models of Parsing and Translation. PhD thesis, Dublin City University, Dublin, Ireland.

Hearne, M., Ozdowska, S., and Tinsley, J. 2008. Comparing Constituency and Depedency Representations for SMT Phrase-Extraction. *In* Actes des 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN '08), Avignon, France.

Hearne, M., Tinsley, J., Zhechev, V., and Way, A. 2007. Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner. *In* Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07), pp. 83–94, Skövde, Sweden.

Hearne, M. and Way, A. 2003. Seeing the Wood for the Trees: Data-Oriented Translation. *In* Machine Translation Summit IX, pp. 165–172, New Orleans, LA.

Hearne, M. and Way, A. 2006. Disambiguation Strategies for Data-Oriented Translation. *In* Proceedings of the 11th Conference of the European Association for Machine Translation (EAMT-06), pp. 59–68, Oslo, Norway.

Imamura, K. 2001. Hierarchical Phrase Alignment Harmonized with Parsing. *In* Proceedings of Sixth Natural Language Processing Pacific Rim Symposium, pp. 206–214, Tokyo, Japan.

Johnson, H., Joel, M., Foster, G., and Kuhn, R. 2007. Improving Translation Quality by Discarding Most of the Phrasetable. *In* Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), pp. 967–975, Prague, Czech Republic.

Johnson, M. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics* **28**:71–76.

Kaji, H., Kida, Y., and Morimoto, Y. 1992. Learning Translation Templates from Bilingual Text. *In* Proceedings of the 15th Conference on Computational linguistics, pp. 672–678, Nantes, France.

Kikui, G., Sumita, E., Takezawa, T., and Yamamoto, S. 2003. Creating Corpora for Speech-to-Speech Translation. *In* Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech-08), pp. 381–384, Geneva, Switzerland.

Kikui, G., Yamamoto, S., Takezawa, T., and Sumita, E. 2006. Comparative Study on Corpora for Speech Translation. *IEEE Transactions on Audio, Speech and Language Processing* 14:1674–1682.

Kneser, R. and Ney, H. 1995. Improved Backing-off for M-gram Language Modeling. *IEEE International Conference on Acoustics, Speech, and Signal Processing* 1:181–184.

Knight, K. 1999. Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics* 25:607–615.

Koehn, P. 2004. Statistical Significance Tests for Machine Translation Evaluation. *In* Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 388–395, Barcelona, Spain.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *In* Machine Translation Summit X, pp. 79–86, Phuket, Thailand.

Koehn, P. November 2009. Statistical Machine Translation. Cambridge University Press.

Koehn, P. and Hoang, H. 2007. Factored Translation Models. *In* Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 868–876, Prague, Czech Republic.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. 2007. Moses: Open Source

Toolkit for Statistical Machine Translation. *In* 45th Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, pp. 177–180, Prague, Czech Republic.

Koehn, P., Och, F. J., and Marcu, D. 2003. Statistical Phrase-Based Translation. *In* Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03), pp. 48–54, Edmonton, Canada.

Lambert, P. 2008. Exploiting Lexical Information and Discriminative Alignment Training in Statistical Machine Translation. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain.

Lardilleux, A. and Lepage, Y. 2008. A Truly Multilingual, High Coverage, Accurate, yet Simple, Subsentential Alignment Method. *In* Proceedings of the 8th Conference of the Association for Machine Translation in the Americas (AMTA), pp. 125–132, Waikiki, HI.

Lavie, A. 2008. Stat-XFER: A General Search-based Syntax-driven Framework for Machine Translation. *In* Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-08) - Invited Paper, pp. 362–375, Haifa, Israel.

Lavie, A. and Agarwal, A. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. *In* Proceedings of the Second ACL Workshop on Statistical Machine Translation (SSST-2), pp. 228–231, Prague, Czech Republic.

Lavie, A., Parlikar, A., and Ambati, V. 2008. Syntax-driven Learning of Subsentential Translation Equivalents and Translation Rules from Parsed Parallel Corpora. *In* Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation (SSST-2), pp. 87–95, Columbus, OH.

Liu, Y., Liu, Q., and Lin, S. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. *In* Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 609–616, Sydney, Australia.

Lu, Y., Huang, J., and Liu, Q. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. *In* Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), pp. 343–350, Prague, Czech Republic.

Ma, Y., Tinsley, J., Hassan, H., Du, J., and Way, A. 2008. Exploiting Alignment Techniques in MaTrEx: the DCU Machine Translation System for IWSLT08. *In* Proc. of the International Workshop on Spoken Language Translation, pp. 26–33, Waikiki, HI.

Manning, C. and Schütze, H. 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.

Marcu, D. and Wong, W. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *In* Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), pp. 133–139, Philadelphia, PA.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. 1994. The Penn Treebank: Annotating Predicate Argument Structure. *In* Proceedings of the ARPA Workshop on Human Language Technology, pp. 110–115, Princeton, NJ.

Marton, Y. and Resnik, P. 2008. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. *In* 46th Annual Meeting of the Association for Computational Linguistics (ACL'08), pp. 1003–1011, Columbus, OH.

Megyesi, B., Dahlqvist, B., Pettersson, E., and Nivre, J. 2008. Swedish-Turkish Parallel Treebank. *In* Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-2008), Marrakech, Morocco.

Melamed, I. D. 1998. Annotation Style Guide for the Blinker Project. Technical Report 98-06, University of Pennsylvania, Philadelphia, PA.

Menezes, A. and Richardson, S. D. 2003. A Best-First Alignment Algorithm for Extraction of Transfer Mappings, pp. 421–442. *In* M. Carl and A. Way (eds.), Recent Advances in Example-Based Machine Translation. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38:39–41.

Monson, C., Llitjós, A. F., Ambati, V., Levin, L., Lavie, A., Alvarez, A., Aranovich, R., Carbonell, J., Frederking, R., Peterson, E., and Probst, K. 2008. Linguistic Structure and Bilingual Informants Help Induce Machine Translation of Lesser-Resourced Languages. *In* Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-2008), pp. 26–30.

Nesson, R., Shieber, S. M., and Rush, A. 2006. Induction of Probabilistic Synchronous Tree-Insertion Grammars for Machine Translation. *In* Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA 2006), pp. 138–127, Cambridge, MA.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13:95–135.

Och, F. J. 2003. Minimum Error Rate Training in Statistical Machine Transla-

tion. *In* 41st Annual Meeting of the Association for Computational Linguistics (ACL'03), pp. 160–167, Sapporo, Japan.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., and Radev, D. 2004. A Smorgasbord of Features for Statistical Machine Translation. *In* D. M. Susan Dumais and S. Roukos (eds.), HLT-NAACL 2004: Main Proceedings, pp. 161–168, Boston, Massachusetts, USA.

Och, F. J. and Ney, H. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *In* 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), pp. 295–302, Philadelphia, PA.

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

Och, F. J. and Ney, H. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics* 30:417–449.

Och, F. J., Tillmann, C., and Ney, H. 1999. Improved Alignment Models for Statistical Machine Translation. *In* Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 20–28, College Park, MD.

Owczarzak, K. 2008. A Novel Dependency-Based Evaluation Metric for Machine Translation. PhD thesis, Dublin City University, Dublin, Ireland.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical report, IBM T.J. Watson Research Center, Yorktown Heights, NY.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *In* 40th Annual Meeting of the

Association for Computational Linguistics (ACL-02), pp. 311–318, Philadelphia, PA.

Peterson, E. 2002. Adapting a Transfer Engine for Rapid Machine Translation Development. Master's thesis, Georgetown University, Washington, D.C.

Petrov, S. and Klein, D. 2007. Improved Inference for Unlexicalized Parsing. *In* Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, pp. 404–411, Rochester, NY.

Porter, M. F. 1980. An Algorithm for Suffix Stripping. *Program* **14**:130–137.

Poutsma, A. 2000. Data-Oriented Translation: Using the Data-Oriented Parsing framework for Machine Translation. Master's thesis, University of Amsterdam, The Netherlands.

Poutsma, A. 2003. Machine Translation with Tree-DOP, pp. 63–81. *In* R. Bod, R. Scha, and K. Sima'an (eds.), Data-Oriented Parsing. Stanford CA: CSLI Publications.

Samuelsson, Y. and Volk, M. 2006. Phrase Alignment in Parallel Treebanks. *In* Proceedings of 5th Workshop on Treebanks and Linguistic Theories (TLT-06), pp. 93–96, Prague, Czech Republic.

Sánchez-Martínez, F. and Way, A. 2009. Marker-based Filtering of Bilingual Phrase Pairs for SMT. *In* 13th Annual Meeting of the European Association for Machine Translation (EAMT-09), pp. 144–151, Barcelona, Spain.

Schluter, N. and van Genabith, J. 2007. Preparing, Restructuring and Augmenting a French Treebank: Lexicalised Parsing or Coherent Treebanks? *In* Proceedings of the 10th Conference of the Pacific Association of Computational Linguistics (PACLING 2007), Melbourne, Australia.

Schmid, H. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. *In* Proceedings of the 20th International Conference on Computational Linguistics (COLING 04), pp. 162–168, Geneva, Switzerland.

Sima'an, K. and Buratto, L. 2003. Backoff Parameter Estimation for the DOP Model. *In* Proceedings of the 14th European Conference on Machine Learning (ECML'03), pp. 373–384, Cavtat-Dubrovnik, Croatia.

Srivastava, A., Penkale, S., Tinsley, J., and Groves, D. 2009. Evaluating Syntax-Driven Approaches to Phrase Extraction for MT. *In* Proceedings of the 3rd Workshop on Example-Based Machine Translation, p. (to appear), Dublin, Ireland.

Srivastava, A. and Way, A. 2009. Using Percolated Dependencies for Phrase Extraction in SMT. *In* Proceedings of Machine Translation Summit XII, pp. 316–323, Ottawa, Canada.

Stolcke, A. 2002. SRILM - An Extensible Language Modeling Toolkit. *In* Proceedings of the International Conference Spoken Language Processing, pp. 901–904, Denver, CO.

Stroppa, N., van den Bosch, A., and Way, A. 2007. Exploiting Source Similarity for SMT using Context-Informed Features. *In* Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07), pp. 231–240, Skövde, Sweden.

Telljohann, H., Hinrichs, E., and Kübler, S. 2004. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. *In* Proceedings of the Fourth International Conference on Language Resources and Evaluation, Porto, Portugal.

Tinsley, J., Hearne, M., and Way, A. 2007a. Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. *In* Proceedings of the

Sixth International Workshop on Treebanks and Linguistic Theories (TLT-07), pp. 175–187, Bergen, Norway.

Tinsley, J., Hearne, M., and Way, A. 2009. Parallel Treebanks in Phrase-Based Statistical Machine Translation. *In* Proceedings of the Tenth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), pp. 318–331, Mexico City, Mexico.

Tinsley, J. and Way, A. 2009. Automatically-Generated Parallel Treebanks and their Exploitability in Phrase-Based Statistical Machine Translation. *Machine Translation* p. (in press).

Tinsley, J., Zhechev, V., Hearne, M., and Way, A. 2007b. Robust Language-Pair Independent Sub-Tree Alignment. *In* Machine Translation Summit XI, pp. 467–474, Copenhagen, Denmark.

Véronis, J. and Langlais, P. 2000. Evaluation of Parallel Text Alignment Systems. the ARCADE Project, pp. 369–388. *In* J. Véronis (ed.), Parallel Text Processing: Alignment and Use of Translation Corpora, chapter 19. Kluwer Academic Publishers, Dordrecht.

Vilar, D., Stein, D., and Ney, H. 2008. Analysing Soft Syntax Features and Heuristics for Hierarchical Phrase Based Machine Translation. *In* Proceedings of the International Workshop on Spoken Language Translation, pp. 190–197, Waikiki, HI.

Volk, M. and Samuelsson, Y. 2004. Bootstrapping Parallel Treebanks. *In* Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora (COLING2004), pp. 63–69, Geneva, Switzerland.

Way, A. and Groves, D. 2005. Hybrid Data-Driven Models of Machine Translation. *Machine Translation: Special Issue on Example-Based Machine Translation* **19**:301–323.

Wellington, B., Waxmonsky, S., and Melamed, I. D. 2006. Empirical Lower Bounds on the Complexity of Translational Equivalence. *In* Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 977–984, Sydney, Australia.

Wu, D. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics* 23:377–404.

Xia, F. and Palmer, M. 2001. Converting dependency structures to phrase structures. *In* HLT '01: Proceedings of the First International Conference on Human Language Technology Research, pp. 1–5.

Yamada, K. and Knight, K. 2001. A Syntax-Based Statistical Translation Model. *In* Proceedings of the Joint Meeting of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01) and the 10th Meeting of the European Association for Computational Linguistics, pp. 523–530, Toulouse, France.

Yamada, K. and Knight, K. 2002. A Decoder for Syntax-based Statistical MT. *In* ACL, pp. 303–310, Philadelphia, PA.

Zhang, Y. and Vogel, S. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. *In* Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05), pp. 294–301, Budapest, Hungary. The European Association for Machine Translation.

Zhang, Y. and Vogel, S. 2006. Suffix Array and its Applications in Empirical Natural Language Processing. *In* Technical Report CMU-LTI-06-010, Pittsburgh, PA.

Zhang, Y., Vogel, S., and Waibel, A. 2004. Interpreting Bleu–NIST scores: How much improvement do we need to have a better system? *In* Proceedings of

the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal.

Zhechev, V. 2009. Automatic Generation of Parallel Treebanks: An Efficient Unsupervised System. PhD thesis, Dublin City University, Dublin, Ireland.

Zhechev, V. and Way, A. 2008. Automatic Generation of Parallel Treebanks. *In* Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08), pp. 1105–1112, Manchester, UK.

Zollmann, A. and Venugopal, A. 2006. Syntax-Augmented Machine Translation via Chart Parsing. *In* Proceedings of the Workshop on Statistical Machine Translation: HLT–NAACL 2006, pp. 138–141, New York, NY.

Zollmann, A., Venugopal, A., Och, F., and Ponte, J. 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. *In* Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08), pp. 1145–1152, Manchester, England.