PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Linguistically Motivated Reordering Modeling for Phrase-Based Statistical Machine Translation

Arianna Bisazza

Advisor:

Marcello Federico	Fondazione Bruno Kessler

Thesis Committee:

Alexander Fraser	University of Stuttgart
Philipp Koehn	University of Edinburgh
Christof Monz	University of Amsterdam

April 2013

Abstract

Word reordering is one of the most difficult aspects of Statistical Machine Translation (SMT), and an important factor of its quality and efficiency. While short and medium-range reordering is reasonably handled by the phrase-based approach (PSMT), long-range reordering still represents a challenge for state-of-the-art PSMT systems. As a major cause of this problem, we point out the inadequacy of existing reordering constraints and models to cope with the reordering phenomena occurring between distant languages. On one hand, the reordering constraints used to control translation complexity appear to be too coarse-grained. On the other hand, the reordering models used to score different reordering decisions during translation are not discriminative enough to effectively guide the search over very large sets of hypotheses. In this thesis we propose several techniques to improve the definition of the reordering search space in PSMT by exploiting prior linguistic knowledge, so that long-range reordering may be adequately handled without sacrificing efficiency. In particular, we focus on Arabic-English and German-English: two language pairs characterized by uneven distributions of reordering phenomena, with long-range movements concentrating on few patterns. Through extensive experiments, we show that our techniques can significantly advance the state of the art in PSMT for these challenging language pairs. When compared with a popoular tree-based SMT approach, our best PSMT systems achieve comparable or higher reordering accuracies while being considerably faster.

Keywords
[natural language processing; statistical machine translation; word reordering]

Aknowledgements

My first encounter with machine translation dates back to almost five years ago. I had always been fascinated by the study of foreign languages, but only recently by the possibility of modeling them through computers. Since then it didn't take me long to realize that this was the very research field I wanted to work in. Still, it is thanks to many helpful and inspiring people that my PhD became such an amazing journey.

Without a doubt, I owe most of what I have learned in these years to my advisor Marcello Federico, whom I could never thank enough for accepting me in his group despite my atypical background. His enthusiasm for research and constant attention to detail have greatly helped me to make my PhD experience successful. I cannot easily remember a meeting with him that did not result in a precious piece of advice.

I would also like to express my gratitude for being part of such a diverse research team. It would have been much harder to carry out my work without the chance of sharing and discussing problems of all kinds with other students and researchers on a daily basis. For this reason I wish to thank all the present and past members of the HLT unit at Fondazione Bruno Kessler, and in particular (...in order of appearance):

- Sara Tonelli for the simple but fundamental reason that she informed me about the existence of a machine translation group at FBK;
- Roberto Gretter for initiating me into the field of speech recognition;
- Nicola Bertoldi and Roldano Cattoni for helping me through the intricacies of SMT technology and cluster computing;
- Mauro Cettolo, Matteo Negri and Marco Turchi for fostering discussion and providing useful feedback at our weekly round table;
- Christian Hardmeier for sharing with me his eclectic knowledge of linguistics and computer science, and especially for helping me analyze the word reordering patterns of German-English;
- Daniele Pighin for bringing his machine learning expertise to my earlier work;
- my fellow PhD students Silvana Bernaola, Gözde Özbal, Nick Ruiz, Prashant Mathur,

> *Yashar Mehdad, José Camargo De Souza, M. Amin Farajian and more... for being always available to brainstorm research and life issues, and for adding to every working day precious moments of cultural exchange and friendship.*

Outside of Trento, I had the chance to spend three unforgettable months at Microsoft Redmond. I am exceptionally grateful to Chris Quirk and to the whole NLP group at MS for letting me be a part of their unique research environment and get behind the wheel of one of the world largest SMT engines.

Thanks to Alexandra Birch for helpful discussions on word order evaluation, and to Hieu Hoang for his great technical support on the Moses engine.

Thanks to John Tinsley and Alexandru Ceausu for our valuable collaboration at Dublin City University.

I am also grateful to the members of my dissertation committee – Alexander Fraser, Philipp Koehn and Christof Monz – for their encouragement and helpful feedback.

Finally, thanks to my family, my parents and my brothers, for their constant and loving support. And thanks to Fatih, my greatest listener and future husband, for accompanying me through this adventure and making these last five years the happiest of my life. None of this would have been possible without them.

Part of the research presented in this thesis was supported by the European Union Seventh Framework Programme under grant agreement 287658 (EU-BRIDGE).

To my grandfather, nonno Franco, who knew how to talk to computers and to people.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Statistical machine translation (SMT) is a data-driven approach to the translation of text from a natural language into another. The core SMT methods [Brown et al., 1990, 1993, Berger et al., 1996, Koehn et al., 2003] – emerged in the 1990s and matured in the 2000's to become widespread today – learn direct correspondences between surface forms in the two languages, without the need of abstract linguistic representations. The main advantages of SMT are versatility and cost-effectiveness: in principle, the same modeling framework can be applied to any pair of languages with minimal human effort.

However, experiences in a diverse range of language pairs have revealed that this form of shallow modeling makes SMT highly sensitive to structural differences between source and target language – e. g. at the level of morphology or word order. A number of enhancements to the original SMT formulation have then been proposed with the aim of overcoming these limitations. Morphology has been addressed, for instance, by affix segmentation (e. g. Habash and Sadat [2006], Durgar El-Kahlout and Oflazer [2006]) or factored models [Koehn and Hoang, 2007]. As for word order, it has been one of the main motivations for the development of tree-based SMT (e. g. Wu [1997], Yamada [2002], Galley et al. [2004], Chiang [2005]), a trend of major importance in the field.

Currently none of these methods can be said to unconditionally dominate the others. Rather, the choice of an optimal SMT framework for a new task appears to be mostly driven by empirical trials, in which shallow methods often prove stronger than the more structured and linguistically informed ones. From this observation, two important research questions arise: is linguistic structure necessary for machine translation? And, if so, to what to extent?

Human translation studies provide an interesting perspective on this issue. According to many scholars [Gile, 2005, Craciunescu et al., 2004], the human translation process involves phases of deep understanding of the source text, alternated with phases of semantic transfer and subsequent target text production. However, this view is challenged by other scholars [Mossop, 2003, Ruiz et al., 2008, Carl, 2011] who state that translation mostly requires just a shallow and partial understanding of the source text, and that text production can occur in parallel with understanding, at a very local level.

In the light of this second theory, shallow SMT methods need not be seen as a cheap and limited alternative to sophisticated, structure-rich methods. They become instead the natural choice for a wide range of language pairs where complete comprehension and structured representation of the source sentence is actually not necessary to produce an accurate translation.

In this thesis we aim at improving translation quality between distant languages, while maintaining the SMT machinery as shallow and unstructured as possible. In particular, we focus on the problem of word reordering between languages whose syntax differs considerably, yet not at the point of requiring a complex and costly tree-based solution.

Word reordering is probably the most difficult aspect of SMT, and an important factor of both its quality and efficiency. Experience shows that different kinds of reordering call for different SMT frameworks: namely, language pairs with very similar word orders naturally fit a shallow modeling framework, such as the phrase-based one. In contrast, language pairs with radically different word orders are better bridged through a structured representation, i. e. tree-based frameworks. Lying between these two extremes are language pairs where most of the reordering is local, and where long reorderings can be isolated and described by a handful of linguistic rules. We argue that tree-like structures are not needed to model this last kind of language pairs. Instead, we focus on enhancing the phrase-based approach to better account for uneven reordering distributions.

1.1 Motivating example

The error analysis of a strong Arabic-English PSMT baseline developed for the NIST 2009 evaluation[1] revealed that the incorrect reordering of Arabic Verb-Subject-Object (VSO) sentences was the main cause of disfluent outputs.

Indeed, modern Arabic syntax admits both VSO and SVO orders, with VSO being dominant in written narrative texts, such as news stories. Figure 1.1 shows two examples where the main verb precedes the subject in Arabic.[2] In such sentences, the subject can

[1] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[2] Throughout this thesis, the Arabic text is transliterated according to the Buckwalter scheme.

SRC	استدعت كل من السعودية و ليبيا و سوريا سفراء ـها في الدنمارك			
	verb	subj.	obj.	compl.
	w **AstdEt**	kl mn AlsEwdyp w lybyA w swryA	sfrA' hA	fy AldnmArk
REF	Each of Saudi Arabia, Libya and Syria **recalled** their ambassadors from Denmark			
MT	He **recalled** all from Saudi Arabia, Libya and Syria ambassadors in Denmark			
SRC	جدد العاهل المغربي الملك محمد السادس دعم ـه لـ مشروع الرئيس الفرنسي			
	verb	subj.	obj.	compl.
	jdd	AlEAhl Almgrby Almlk mHmd AlsAds	dEm h	l m$rwE Alr}ys Alfrnsy
REF	The Moroccan monarch King Mohamed VI **renewed** his support to the project of French President			
MT	The Moroccan monarch King Mohamed VI ∅ his support to the French President			

Figure 1.1: Arabic-English PSMT errors due to incorrect reordering of the verb in VSO sentences.

be followed by adjectives, adverbs, coordinations, or appositions that further increase the distance between the verb and its object. When translating into English – a primarily SVO language – the resulting long-distance reorderings are often missed by the SMT system. In our examples, missed verb reorderings resulted in different translation errors: the introduction of a spurious subject pronoun and the omission of the verb, respectively.

Interestingly, it appeared from the same error analysis that word reordering was mostly correct otherwise. Indeed, other order differences are also very frequent between Arabic and English sentences, due to the head-initial structure of Arabic noun phrases. These reorderings, however, are mostly local, hence likely to be captured by the standard PSMT reordering mechanisms.

As our analysis of PSMT reordering proceeded, it became more evident that long-range reordering errors were due to an ensemble of causes. On one hand, the *reordering constraints* used to control the complexity of the translation process were too coarse: they could either restrict the translation search space to short-range reorderings only, or open it to short and long-range reorderings of all kinds. On the other hand, the *reordering models* used to score different reordering decisions during translation were not discriminative enough to guide the search over very large sets of hypotheses. Given that reordering modeling had already been studied extensively, we decided to focus primarily on improving the definition of the search space by exploiting prior linguistic knowledge.

1.2 Contributions

The main research contributions of this thesis are as follows:

- we provide a theoretical ground for a number of empirical findings on the complexity of reordering in different language pairs;
- we specifically analyze long-range reordering patterns in Arabic-English and German-English, and develop shallow-syntax reordering rule sets to model them;
- we introduce a novel technique to suggest likely input reorderings to a PSMT system: i. e. modified distortion matrices;
- we present a fully data-driven method to dynamically refine the reordering search space explored by standard PSMT systems;
- we propose an evaluation metric to detect improvements in the reordering of specific word classes;
- we compare our best methods with a competitive tree-based approach – i. e. hierarchical SMT [Chiang, 2005] – and achieve comparable or higher reordering accuracies with systems that are up to two times faster.

1.3 Structure of the thesis

The remainder of this thesis is structured as follows.

Chapter 2 introduces the fundamental concepts of SMT and explains how the word reordering problem is treated in different SMT approaches. The previous work in advanced reordering modeling is extensively reviewed, and different evaluation methods are presented.

Chapter 3 analyzes the word order differences of several language pairs, based on a large body of theoretical linguistic knowledge. Empirical results in the SMT literature are shown to support the hypothesis that linguistic knowledge is useful to predict the reordering characteristics of a language pair and to select the SMT framework that best suits them.

Chapter 4 describes our first method to improve the handling of long-range reordering phenomena in PSMT. Simple rules based on shallow syntax are designed to predict probable verb reorderings in Arabic-English. Word reordering lattices are then used to provide such reordering suggestions to a PSMT system. Finally, the problem of dense and noisy lattices is addressed with a lattice pruning technique based on a discriminative classifier. Improvements in terms of translation quality and reordering accuracy are reported over a PSMT baseline tested on well-known news translation benchmarks.

Chapter 5 introduces a novel and efficient technique to suggest likely input reorderings to a PSMT system. This consists of modifying the distortion penalty function associated to each input sentence, so that the cost for the desired permutations is reduced. The technique is shown to improve a competitive PSMT baseline with no loss in translation speed, in both an Arabic-English and a German-English task.

Chapter 6 presents a fully-data driven method to dynamically shape the reordering search space. Specifically, the reordering options admitted by loose reordering constraints are pruned on-the-fly through a binary classifier that predicts whether a given input word should be translated right after another. Evaluated on Arabic-English and German-English against a strong PSMT baseline, the method is shown to preserve translation quality under very loose reordering constraints. Moreover the reordering of verbs is significantly improved and translation time is considerably reduced.

Chapter 7 concludes the thesis with a comparative evaluation of the proposed reordering techniques. In order to position this work in the broader field of SMT, a state-of-the-art tree-based SMT system is also included in the evaluation. After a detailed discussion of translation quality and efficiency results, the chapter summarizes the major findings of the thesis and suggests future research directions.

1.4 Related publications

Parts of Chapter 4 were published in the following papers:

- "Chunk-based verb reordering in VSO sentences for Arabic-English statistical machine translation" appeared in the proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR [Bisazza and Federico, 2010].
- "Chunk-lattices for verb reordering in Arabic-English statistical machine translation" appeared in the Machine Translation Journal, Special Issue on MT for Arabic [Bisazza et al., 2012].
- "Word lattices for morphological reduction and chunk-based reordering" appeared in the proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR [Hardmeier et al., 2010].

Chapter 6 elaborates on the following conference paper:

- "Modified distortion matrices for phrase-based statistical machine translation" published in the proceedings of the 50th Annual Meeting of the Association for Computational Linguistics [Bisazza and Federico, 2012].

Finally, Chapter 7 extends the journal article currently under revision:

- "Dynamically shaping the reordering search space of phrase-based statistical machine translation" submitted to the Transactions of the Association for Computational Linguistics [Bisazza and Federico, 2013].

Chapter 2

Word Reordering in Statistical Machine Translation

Word order differences are among the most important factors determining the performance of statistical machine translation on a given language pair.

The first vision of a computer translating between natural languages, based on the statistical techniques of information theory, dates back to the memorandum of Warren Weaver [1949]. Later on, the emergence of modern SMT methods was strongly influenced by the speech processing technology. Much like a speech recognizer, the first SMT system was based on a noisy channel formulation that viewed the source language text as a coded signal that had to be decoded to obtain the original message in the target language. While this idea was essential to spur decades of research that led to the modern SMT systems, it appeared soon that the translation problem had additional factors of complexity. Word reordering was probably the most disruptive among them.

Natural languages vary greatly in how they arrange the sentence constituents. Thus, finding the optimal translation of a sentence generally implies a much larger search space than finding the optimal transcription of an utterance. Since the beginning, SMT researchers tried to solve this problem with various modeling strategies, and by heuristically restricting the possible word reordering operations. Word reordering research has advanced along with the core SMT research and has sometimes directed it. Nevertheless, up to date, word order differences remain among the most important factors determining the performance of SMT systems on a given language pair.

2.1 Word-based models

According to the noisy channel formulation, the SMT process is called decoding and consists in searching for the most probable target (or English) sentence $\mathbf{e}^*$ given a source (or foreign) sentence $\mathbf{f}$:

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

Using the Bayes' theorem, the probability is decomposed as follows:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e}) \tag{2.2}$$

Here, $p(\mathbf{e})$ represents the target language model, whose task is to estimate the generated sentence's fluency independently from the input sentence, and $p(\mathbf{f}|\mathbf{e})$ represents the translation model, whose task is to estimate the likelihood of the input sentence $\mathbf{f}$ given any translation hypothesis $\mathbf{e}$. Early SMT approaches [Brown et al., 1990, 1993, Berger et al., 1996] are built precisely on these two components, factorizing each at the level of words. For the language model, a monolingual n-gram model assigning a probability to each target word given its n-1 preceding words. For the translation model, instead, dependencies between source and target words are first mediated by an *alignment model* $p(\mathbf{f}, \mathbf{a}|\mathbf{e})$ such that:

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{f}, \mathbf{a}|\mathbf{e}) \tag{2.3}$$

where the hidden variable $\mathbf{a}$ represents any map from the source word positions to the target word positions. Hence, the alignment model is factored at the level of words by introducing a bilingual lexicon model measuring the association of source-target word pairs. Notice that for computational efficiency reasons, during decoding (equation 2.2) the summation over all alignments is replaced with a maximization over the most probable alignment.

While the language model only needs large amounts of text in the target language to be trained, the translation model requires word-aligned parallel texts. In practice, though, word alignment is typically not available at the beginning of the training process and is treated as a hidden variable. The expectation maximization algorithm is therefore used to iteratively learn word alignment *and* translation model parameters from a parallel corpus only aligned at the level of sentences.

Word reordering was not explicitly modeled by the first SMT approaches. To overcome this and other limitations, word translation models of increasing complexity were designed throughout the 1990's, including, for instance, word-to-word distance probabilities. Eventually, another modeling advance marked a major breakthrough in the field: namely, the change of translation units from words to phrases.

2.2 Phrase-based models

The phrase-based approach (PSMT) [Zens et al., 2002, Koehn et al., 2003, Och and Ney, 2002] introduced two important novelties: namely, the incorporation of context in the translation unit, and the move from a generative to a discriminative modeling framework.

Formally, PSMT builds on the same equation as word-based SMT (2.2) but decomposes the translation probability $p(\mathbf{f}|\mathbf{e})$ differently:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f}|\mathbf{e}) = \arg\max_{e_1^L} p_{\text{LM}}(e_1^L) \max_{b_1^I} p_{\text{TM}}(f_1^J, b_1^I | e_1^L) \tag{2.4}$$

The target string is still modeled by a word-based n-gram model p_{LM}:

$$p_{\text{LM}}(e_1^L) = \prod_{l=1}^{L} p_{\text{LM}}(e_l | e_{l-n+1}..e_{l-1}) \tag{2.5}$$

while the translation model is governed by the phrase-alignment variable b_1^I embedding both a segmentation and a reordering of the source and target phrases. This is defined as:

$$b_1^I = ((J_1, I_1), (J_2, I_2), \ldots, (J_I, I_I)) \tag{2.6}$$

such that $I_1, \ldots, I_I$ are contiguous intervals partitioning the target word positions $1, \ldots, L$, and $J_1, \ldots, J_I$ are corresponding but not necessarily contiguous intervals partitioning the source word positions $1, \ldots, M$. Hence, the phrase alignment model is decomposed into a phrase-level model ϕ and a phrase distortion model d:

$$p_{\text{TM}}(f_1^M, b_1^I | e_1^L) = d(b_1^I)\ \phi(f_1^M | e_1^l, b_1^I) = \prod_{i=1}^{I} d(J_{i-1}, J_i)\ \phi(\tilde{f}_i | \tilde{e}_i) \tag{2.7}$$

where $\tilde{f}_i$ is a shorthand for the substring (phrase) of $\mathbf{f}$ spanning the source interval J_i, and $\tilde{e}_i$ is a shorthand for the substring (phrase) of $\mathbf{e}$ spanning the target interval I_i.

The second probability factor represents a basic reordering model that exponentially

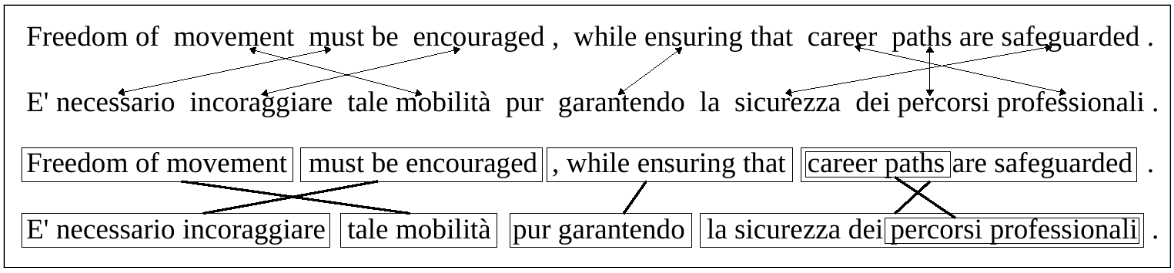

Freedom of movement must be encouraged , while ensuring that career paths are safeguarded .
E' necessario incoraggiare tale mobilità pur garantendo la sicurezza dei percorsi professionali .
Freedom of movement
must be encouraged
, while ensuring that
career paths
are safeguarded
.
E' necessario incoraggiare
tale mobilità
pur garantendo
la sicurezza dei
percorsi professionali
.

Figure 2.1: Word (top) and phrase (bottom) alignments of an English-Italian sentence pair.

penalizes longer jumps among consecutively translated phrases:

$$d(J_{i-1}, J_i) = e^{-\left|\text{start}(J_i) - \text{end}(J_{i-1}) - 1\right|} \tag{2.8}$$

This model, also known as distortion cost or penalty, assigns the maximum probability $e^0 = 1$ to translations that preserve the order of the source phrases (monotonic). Note, finally, that phrases in PSMT do not necessarily correspond to well-formed syntactic phrases.

Modeling phrases mainly helps to solve the problem of identifying one-to-many word equivalences, and of translating ambiguous words. Besides, it makes it possible to capture a considerable amount of local reordering phenomena into the translation units. The typical PSMT training pipeline involves word-level alignment of the parallel data with methods similar to those of word-level SMT. Source-target phrase pairs are then extracted from the word-aligned sentences by language-independent heuristics, and finally scored based on relative frequencies. The difference between word- and phrase-based translation is illustrated by Figure 2.1: the PSMT approach allows the models not only to abstract from the level of words in order to capture local context *inside* the translation units (e. g. from *must–necessario* and *encouraged–incoraggiare* to *[must be encouraged]–[necessario incoraggiare]*), but also to solve local reordering (e. g. from *career–professionali* and *paths–percorsi* to *[career paths]–[percorsi professionali])*.

The advent of PSMT also coincided with the transition to a log-linear modeling framework [Och and Ney, 2002], allowing for the integration of additional model components in the form of weighted feature functions. In this framework, equation (2.4) is replaced by:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \max_{\mathbf{b}} \; \exp\left[\sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{b})\right] \tag{2.9}$$

where $h_r(\mathbf{e}, \mathbf{f}, \mathbf{b})$ are R arbitrary feature functions and λ_r the corresponding feature weights. For instance, a log-linear model corresponding to the generative PSMT model presented above can be obtained with the following features:

- $h_1 = \log p_{\text{LM}}(\mathbf{e})$
- $h_2 = \log \phi(\mathbf{f}|\mathbf{e}, \mathbf{b})$
- $h_3 = \log d(\mathbf{b})$

and uniform weights. In addition to these three core features, state-of-the art PSMT systems typically include:

- an inverse phrase translation model: $\log \phi(\mathbf{e}|\mathbf{f}, \mathbf{b})$,
- direct and inverse lexical translation models: $\log lex(\mathbf{f}|\mathbf{e}, \mathbf{b})$ and $\log lex(\mathbf{e}|\mathbf{f}, \mathbf{b})$,
- a phrase penalty I controlling the number of phrases used to translate the sentence,
- a word penalty L controlling the target output length.

The rationale of weighting the feature functions is that different aspects of translation can have different importance for a given translation task. Feature weights can be tuned discriminatively, by directly optimizing translation performance on a development set. This approach is commonly called minimum error rate training (MERT) [Och, 2003], and relies on automatic evaluation metrics like BLEU (see Section 2.5).

More advanced reordering models will be presented in Section 2.2.2, after a description of the PSMT decoding process.

2.2.1 Decoding

Given a set of trained models, the actual translation process, called decoding, consists of finding the optimal target sentence satisfying equation (2.9).

During PSMT decoding, three main operations have to be performed in parallel: (i) segmenting the input into phrases, (ii) deciding in which order these should be translated, and (iii) choosing, for each input phrase, a translation option among those learnt during training. The target sentence is always built from left to right, while the input sentence positions can be covered in different orders. For instance, given the English sentence of Figure 2.1, a decoder may choose to first translate the source phrase *"must be encouraged"* by the Italian phrase *"E' necessario incoraggiare"*, then it may go back to the first part of the input and translate the phrase *"Freedom of movement"* by *"tale mobilità"* etc. Alternatively, it could start from the first word *"Freedom"* and translate it by *"Libertà"*, then it may continue with the two following words *"of movement"* and translate them by *"di movimento"* etc. After each translation step, the SMT models (or feature functions) incrementally score different aspects of the translation hypothesis, such as the probability of its target words being the translation of its source words, the fluency of the target word sequence produced so far, or the likelihood of a particular reordering. The weighted sum of all features over a complete hypothesis determines the optimal translation.

Because searching over the space of all possible translations would be NP-hard, SMT decoders employ heuristic search algorithms to only explore a promising subset of the search space. In particular, state-of-the-art PSMT systems rely on a dynamic programming beam-search algorithm [Tillmann and Ney, 2003], and on reordering constraints to limit decoding complexity.

The **dynamic programming** strategy consists of decomposing the problem into simpler and overlapping sub-problems, in such a way that the best overall solution can be obtained by combining the best partial solutions. Each sub-problem is then solved only once and its solution stored in a table for subsequent uses. More specifically, the PSMT decoder exploits the fact that, among all partial decoding paths covering the same subset of input positions, only the best one can lead to the best overall solution. Hence, hypotheses obtained by translating exactly the same input words can be *recombined* during decoding: that is, all but the highest-scoring one can be dropped without loss of optimality. As an example, consider the following translation hypotheses:

[Freedom of movement] ...	[Freedom] [of movement] ...	[Freedom of movement] ...
[La libertà di movimento] ...	[La libertà] [di movimento] ...	[La mobilità] ...

All three hypotheses share the same source words and could be in principle recombined. However, the conditions for hypothesis recombination also depend on the models included in the decoder. For instance, the target n-gram model uses the last n-1 words to compute the score of the next produced word. Recombination is then inhibited between hypotheses that differ in their last n-1 words. Thus, only the first two hypotheses above can be recombined if a bigram or higher-order target language model is used. In general, the complexity of a dynamic programming decoder is exponential in the sentence length.

The **beam search** strategy consists of advancing along many search paths in parallel and abandoning the least promising ones after each step. As opposed to hypothesis recombination, beam search does not guarantee that a pruned path was not actually the one leading to the best overall translation. To reduce the risk of search errors introduced by this approach, pruning is applied to subsets of comparable hypotheses. To this end, partial hypotheses are organized into multiple stacks based on the number of source words translated. For example, the following hypotheses would fall in the 3-source-word stack:

[Freedom of movement] ...	[Freedom] ... [must be]...	... [must be encouraged] ...
[La libertà di movimento] ...	[La libertà] [deve essere] ...	[E' necessario incoraggiare] ...

Moreover, an estimate of the cost needed to complete each hypothesis (i. e. future cost) is included in its score before pruning. Each stack is then pruned according to two criteria:

- **histogram pruning**: discard all but the N best hypotheses in the stack;
- **threshold pruning**: discard all the hypotheses whose score is lower than the stack's best score by a fixed threshold.

Beam search reduces the complexity of dynamic programming decoding from exponential to quadratic in the input sentence length.

Finally, **reordering constraints** are used to limit the space of explorable input permutations. The constraint originally included in the PSMT framework is called **distortion limit (DL)**. This consists of allowing the decoder to skip, or jump, at most k words from the last translated phrase to the next one. More precisely, the limit is imposed on the distortion D between consecutively translated phrases (cf. equation 2.8):

$$\mathrm{D}(J_{i-1}, J_i) = |\mathrm{start}(J_i) - \mathrm{end}(J_{i-1}) - 1| \ \leq \mathrm{DL} \tag{2.10}$$

Notice that a monotonic step means 0-distortion, whereas covering the position immediately preceding the current one, counts for 2. Thus, longer jumps are admitted forwards.

To avoid decoding dead-ends, the distortion limit has to be coupled with another constraint (gap constraint)[1] which ensures that the left-most uncovered input position (ℓ) will still be reachable after translating the next source phrase ($\tilde{f_i}$). Formally, to translate a new phrase $\tilde{f_i}$, the gap $\mathrm{G}(\tilde{f_i}, \ell)$ must not be larger than the DL:

$$\mathrm{G}(\tilde{f_i}, \ell) = |\ell - \mathrm{end}(J_i) - 1| \ \leq \mathrm{DL} \tag{2.11}$$

Setting a low distortion limit means only exploring local reorderings, based on the intuition that languages tend to arrange their sentence constituents in similar orders. This further reduces decoding complexity, making it linear in the sentence length. Besides being

[1] Although implemented in our reference PSMT toolkit Moses [Koehn et al., 2007], this constraint is not included in the original PSMT formulation, nor documented in the toolkit's manual. We use the term "gap constraint" following Chang and Collins [2011].

essential for efficiency, reordering constraints are also important for translation quality because the existing SMT models are typically not discriminative enough to guide the search over very large sets of reordering hypotheses. However, reordering constraints have also several drawbacks, as we will discuss later.

2.2.2 Advanced reordering modeling

Assuming a one-to-one correspondence between source and target phrases, reordering in PSMT can be viewed as the problem of searching through a set of permutations of the input sentence. Thus, two sub-problems arise: defining the set of allowed permutations (reordering constraints) and scoring the allowed permutations according to some likelihood criterion (reordering models or feature functions). We begin with the latter, returning to the constraints later in this section.

Reordering feature functions

Target language modeling is the primary way to reward promising reorderings during translation. This is done indirectly, through the scoring of target word n-grams that are produced by translating the source positions in different orders. However, the fixed window of language models used in SMT (typically 5 or 6 words) makes them mostly insensitive to global reordering phenomena.

In the last years, a growing interest for language pairs with different word orders, such as Arabic-English and Chinese-English, has favored the development of new techniques to explicitly model the reordering problem. Given a source sentence, the search for its optimal reordering is generally decomposed into a sequence of local reordering decisions, as is done for the whole translation process. Thus, the basic reordering step corresponds to the relative positioning of the word/phrase being translated with respect to the word/phrase previously covered. Existing reordering models range from the simplest distortion penalty, where longer reorderings are penalized purely based on the jump length, to more complex models that are conditioned on the words being translated and on their context.

We have already presented in equation (2.8) the **distortion cost** function, which is commonly employed as a baseline reordering model by modern PSMT systems, such as Moses [Koehn et al., 2007]. A weakness of this model is that it penalizes long jumps only after these have been performed, rather than accumulating their cost gradually. As an effect, hypotheses with gaps (i. e. uncovered input positions) can proliferate and cause the pruning of more monotonic hypotheses that could lead to overall better translations.

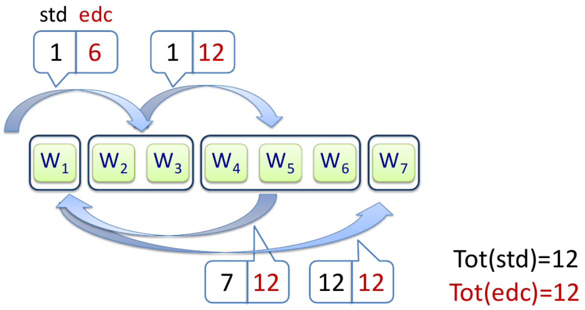


Figure 2.2: Phrase reordering example showing the difference between *standard* and *early* distortion costs (cumulative values). The total cost is the same, but early distortion [Moore and Quirk, 2007] anticipates its accumulation, by incorporating an estimate of the future jumps cost.

To solve this problem, Moore and Quirk [2007] propose an improved version of the distortion cost function which consists in "incorporating an estimate of the distortion penalty yet to be incurred into the estimated score for the portion of the source sentence remaining to be translated" (**early distortion cost**). This function has the same value as the standard one over a complete translation hypothesis, provided that the jump from the last translated word to the end of the sentence is taken into account. As a difference, though, it anticipates the gradual accumulation of the total distortion cost, making hypotheses with the same number of covered words more comparable with one another.

Early distortion cost is computed by an algorithm that keeps track of the uncovered input positions. In Figure 2.2 we provide an example to illustrate the difference between standard and early distortion costs, while we invite the reader to refer to Moore and Quirk [2007] for the detailed algorithm. We have implemented early distortion cost in the Moses platform and used it successfully in some of our experiments, as we will see in Chapters 6 and 7.

The more sophisticated models can be divided into three families: phrase orientation models, jump models and source decoding sequence models. A representative selection of state-of-the-art reordering models is summarized in Table 2.1.

Phrase orientation models [Tillmann, 2004, Koehn et al., 2005], also known as lexicalized reordering models, predict the orientation of a phrase with respect to the last translated one, by classifying it as *monotone, swap* or *discontinuous*.[2] The model probabilities are conditioned on the whole source and target phrases, and they are estimated from the relative frequencies observed in a parallel corpus. Hierarchical phrase orientation models [Galley and Manning, 2008] are a refinement of the latter, improving the way

[2] The discontinuous class can be further divided into *discontinuous left* and *discontinuous right*.

Reordering models	**References**	**Train mode**	**Reordering step classification**	**Features**
linear distortion cost	Koehn & al.′03	–	jump length based	–
lexicalized (hierarchical) phrase orientation	Tillmann′04 Koehn & al.′05 Galley&Manning′08	gener.	ternary (monotonic, swap, discontinuous)	source/target phrases
discriminative phrase orientation	Zens & Ney′06	discr.	binary (left, right)	source/target phrases, words, POS
inbound/outbound/pairwise lexicalized distortion	Al-Onaizan & Papineni′06	gener.	jump length based	source words
inbound/outbound discriminative length-bin	Green & al.′10	discr.	jump length-bin based (9 classes)	source words, POS, position; sent. length
reordered source n-gram	Feng & al.′10a	gener.	–	source words (9-gram context)
source word-after-word	Visweswariah&al.′11	discr.	–	source words, POS context-based

Table 2.1: An overview of state-of-the-art reordering models for PSMT.

of computing the orientation of a new phrase: adjacent blocks can be merged together to form longer phrases, so that a larger number of long-span swaps is detected. As an alternative to the relative-frequency approach, Zens and Ney [2006] addressed the same problem with a maximum-entropy model. Within this framework, they tested a richer combination of features (phrase- and word-level, lexical and word-class based) and found that source word features help most. Phrase orientation models have proven very useful for short and medium-range reordering and are probably the most widely used in PSMT nowadays. However, their coarse classification of reordering steps makes them unsuitable to model long-range reordering phenomena.

Jump models predict the direction and length of a *jump* to perform after a given input position. Al-Onaizan and Papineni [2006] proposed to model the probability of possible jumps given the last covered source word, the word to be translated, or both (outbound, inbound or pairwise lexicalized distortion models). Here, probabilities are conditioned on the exact jump length, wich yields a risk of over-fitting and data sparseness. To cope with this issue, a harsh smoothing factor penalizing longer jumps is applied, so that the lexicalized jump length probability accounts for only 1/9 of the distortion cost actually used in decoding. Green et al. [2010] introduced a discriminative classifier that scores different jumps depending on the words being translated, on their part-of-speech (POS), on their relative position in the sentence, and on the sentence length. In doing this, they take two steps towards robustness. First, jumps are grouped into length bins whose size increases with the jump length:

$$< -6 \quad [-6,-4] \quad [-3,-2] \quad [-1] \quad 0 \quad 1 \quad [2,3] \quad [4,6] \quad > 6$$

Second, irrelevant features are filtered out automatically by means of a regularization technique. Tested on an Arabic-English task, these models outperform hierarchical phrase orientation models only at high distortion limits, but no improvement is shown over the best available baseline. In both these works, best Arabic-English results were obtained within a rather small DL: namely, 8 in [Al-Onaizan and Papineni, 2006] and 5 in [Green et al., 2010], thus failing to capture the rare but crucial long reorderings that are the main motivation of these works. A drawback of the jump modeling approach is that long jumps are typically penalized because of their low frequency compared to short jumps.

Source decoding sequence models predict which input word[3] is likely to be translated at a given state of decoding. Reordered source language models [Feng et al., 2010a] are smoothed n-gram models trained on a corpus of source sentences reordered to match the target word order. When integrated into the SMT system, they assign a probability to each newly translated word given the n-1 previously translated words. In this way, jumps are not directly addressed. Instead, the model rewards reordered word sequences seen in the training data. When testing a 9-gram model on a gold reordered corpus, Feng et al. [2010a] reported a rather high perplexity, likely due to data sparseness and consequent abuse of back-off probabilities by the LM. Nevertheless, integrating the model into a PSMT system yielded a gain in performance comparable to the gain achieved by a maximum-entropy phrase orientation model Zens and Ney [2006]. Slight but consistent improvements were obtained when both models were used together. Finally, source word pair (or word-after-word) reordering models [Visweswariah et al., 2011] estimate, for each pair of input words i and j, the cost of translating j right after i, given various features of i, j and their respective contexts. Differently from reordered source LMs, these models are discriminative and can profit from richer feature sets. At the same time, they do not employ decoding history-based features, which allows for more effective hypothesis recombination. Visweswariah et al. [2011] only applied their model as pre-processing to the training and test data. We will see in this thesis how a similar model can be integrated into a PSMT decoder (Chapter 6).

[3] By “input word” or “source word”, we denote the word at a given position of the input sentence, as opposed to the notion of word type.

Reordering constraints

As previously discussed, limiting the input permutation space is necessary for phrase-based decoding to achieve linear complexity.

We have already described in equation (2.10) the **distortion limit (DL)**, which is commonly used by modern PSMT systems such as Moses Koehn et al. [2007]. In fact, the first constraining paradigms were formulated earlier for word-based SMT in the IBM Research labs [Berger et al., 1996]:

- **MS (max skip)**: at each decoding step, translate one of the first k uncovered source positions. In other words, the translation of a limited number of words (at most k) may be postponed indefinitely.
- **IS (inverted skip)**: at each decoding step, check how many source words after the first uncovered position j have been translated. If they are less than $k-1$, translate any uncovered word, otherwise translate j. This means that the translation of at most $k-1$ source words can be anticipated at any point, while the rest of the sentence is covered monotonically.

The growth of the permutation search space for a sentence of 10 words, with respect to the threshold k of the MS, IS and DL constraints is reported in Table 2.2. We notice that the space defined by DL is considerably smaller than the one defined by IBM constraints, as long as k is small compared to the sentence length. However, both types of constraint grow exponentially with k. The default Moses configuration includes a DL of 6 words, and this is widely accepted as an optimal baseline setting across language pairs.

	Number of permutations (in thousands)								
Threshold (k):	2	3	4	5	6	7	8	9	10+
MS, IS	0.5	13	98	375	933	1729	2581	3266	3629
DL	0.2	3	24	128	476	1246	2333	3266	3629

Table 2.2: Number of permutations allowed for a 10-word sentence by the maximum skip (MS), inverted skip (IS) and distortion limit (DL) reordering constraints, while varying the respective thresholds (k). The total number of permutations is 10!=3,628,800.

An enhancement of the IBM constraints was proposed by Tillmann and Ney [2003] to specifically address the reordering of the verb between German and English, in a word-based SMT system. This paradigm allows to set different thresholds for the anticipation and for the postponement of some input words. A reordering state is added to the decoder to ensure that any reordering pattern (either *skip* or *move*) is completed before initiating a new one. Both reordering patterns are strictly defined and, aside from them, decoding proceeds monotonically. For the German-English translation direction, Tillmann and Ney [2003] manually set their thresholds so that one word may be skipped for at most 4 positions, and up to 2 words may be moved up to 10 positions. In this way, a non-contiguous verb chunk in German may be correctly reordered into a contiguous verb chunk in English. Note, however, that these constraints do not take into account the actual position of the verbs in the input sentence.

A different kind of reordering constraint can be derived from the so-called Inversion Transduction Grammars (**ITG**) Wu [1997]. ITGs only admit permutations that are generated by recursively swapping pairs of adjacent blocks of words. In particular, ITG constraints disallow reorderings that generalize the patterns 3142 and 2413, which are very rarely attested in natural languages.[4] Enforcing ITG constraints in PSMT decoding is not as trivial as enforcing constraints based on word-to-word distances. Zens et al. [2004] proposed a method to do this by inspecting the source coverage vector. However, Cherry et al. [2012] later demonstrated that this method is not able to prevent all non-ITG permutations. Exact ITG-constraint enforcement can be achieved by integrating a deterministic permutation parser into the decoder, as proposed by Feng et al. [2010b]. Interestingly, Cherry et al. [2012] found no consistent benefit from adding hard ITG-constraints to a PSMT system which already included a hierarchical phrase orientation model [Galley and Manning, 2008].

Whether based on word-to-word distances (IBM and Moses-style) or on permutation patterns (skip/move and ITG), the existing reordering constraints are not sensitive to the word being translated nor to its context. Rather, they are uniform throughout the input sentence. This results in a very coarse definition of the reordering search space, which is problematic in language pairs with different syntactic structures.

To address this problem, Yahyaei and Monz [2010] presented a technique to dynamically vary the DL during decoding: they train a discriminative jump model to predict the most probable jump length after each input word, and use the predicted value as the DL after that position. Unfortunately, this method appears to generate inconsistent constraints leading to decoding dead-ends. As a solution, the dynamic DL is relaxed when needed to reach the first uncovered position. The authors reported translation improvements only on a small-scale task with short sentences (BTEC), over a baseline that includes a very simple reordering model.

[4] Refer to [Zens and Ney, 2003] for a comparative study of the IBM and ITG constraints.

2.3 Tree-based models

The SMT approaches presented so far are shallow in the sense that they learn direct correspondences between source and target words, only relying on a local context. Sometimes, though, the translation process would highly benefit from a structured representation of the sentence, where syntactic dependencies between words are made explicit. For instance, the choice of the correct form for the translation of an adjective could depend on a noun that lies outside its immediate context. Likewise, in our motivating example (Figure 1.1), knowing the span of the subject may be enough to predict the correct reordering of the verb. Indeed, word reordering is one of the translation aspects that most motivated the development of tree-based SMT.

To date, many frameworks have been proposed to model translation via tree-like structures, such as Wu [1997], Yamada [2002], Galley et al. [2004], Chiang [2005], Menezes and Quirk [2005], Zollmann and Venugopal [2006] among others. These can differ in the formalism they use to represent the trees, or in how they apply the trees (i. e. to the source, to the target, or to both languages). Moreover, some approaches build on trees produced by pre-trained monolingual parsers, while others induce their grammar directly from word-aligned parallel texts. We refer to Koehn [2010] for a comprehensive overview of the tree-based SMT field, while here we only focus on one approach that has gained popularity as a strong competitor to the phrase-based approach: namely, **hierarchical phrase-based SMT (HSMT)** [Chiang, 2005].

HSMT models build on the formalism of probabilistic Synchronous Context-Free Grammars (SCFG), and are directly learnt from word-aligned parallel data. HSMT rules are not syntactically motivated as they only admit two non-terminal symbols: S for the sentence root, and X. The major strength of HSMT compared to PSMT, is the ability to learn discontinous phrases and long-range reordering rules.

Figure 2.3 shows an example SCFG extracted from a word-aligned parallel sentence. Each synchronous rule associates a non-terminal symbol (left-hand) with a source *and* a target symbol sequence (right-hand). The right-hand side may include non-terminals only (rules 1–2), or a mix of terminals and non-terminals (rule 3), or only terminals (rules 4–5), the latter being equivalent to regular phrase pairs. In particular, rule 3 includes two non-terminal symbols, X_1 and X_2, that are swapped in the target side, thus providing an example of both discontinous phrase and reordering rule. From the same rule, we can see that reordering in HSMT is triggered by lexical items – the so-called lexical anchors of a rule. Indeed, compared to syntax-based approaches where rules contain labeled non-terminals (e. g. VP, NP etc.), HSMT reordering is more precise but generalizes less to new

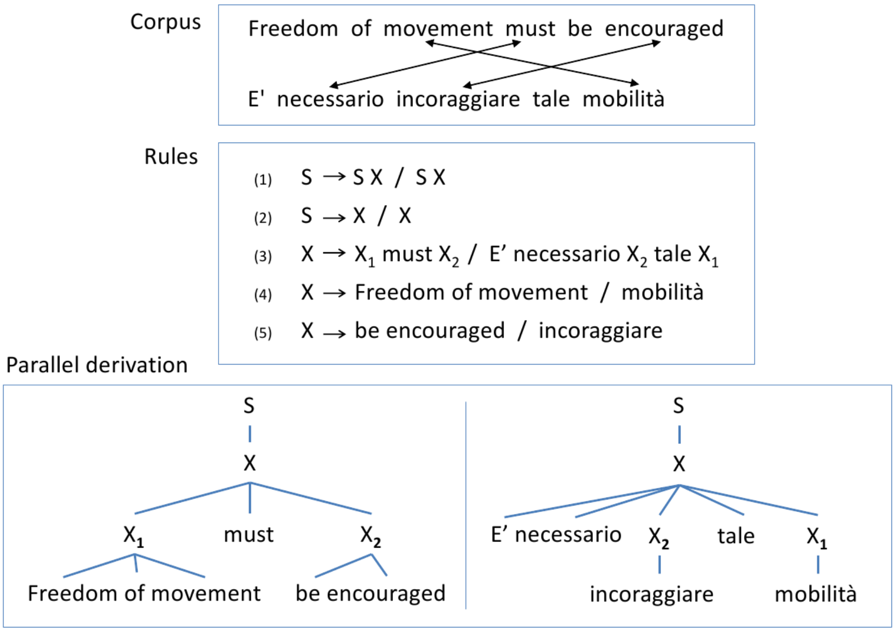

Corpus
Freedom of movement must be encouraged
E' necessario incoraggiare tale mobilità
Rules
(1) S → S X / S X
(2) S → X / X
(3) X → X1 must X2 / E' necessario X2 tale X1
(4) X → Freedom of movement / mobilità
(5) X → be encouraged / incoraggiare
Parallel derivation
S
X
X1
must
X2
Freedom of movement
be encouraged
S
X
E' necessario
X2
tale
X1
incoraggiare
mobilità

Figure 2.3: An example Synchronous Context-Free Grammar extracted from a word-aligned parallel sentence.

data.

Similarly to phrase pairs in PSMT, hierarchical rules are scored by maximum likelihood estimates using relative frequencies.

Tree-based SMT decoders work similarly to parsers. In particular, HSMT decoders are typically based on a chart parsing algorithm. Chart decoding complexity is cubic in the input length, and even higher when taking into account the target language model. This issue can be addressed by different strategies such as cube pruning [Chiang, 2007], which reduces the LM complexity to a constant, or rule application constraints. Although effective in reducing the size of the search space, these methods cannot change the fundamental order of complexity of the algorithm, which remains cubic.

Compared to PSMT decoding, chart decoding is a radically different approach to the word reordering problem, as the target sentence is not produced from left to right but following a tree derivation order. The resulting reordering space is more linguistically motivated, but still too large to be exhaustively explored, therefore HSMT decoders impose a hard limit on the maximum number of source words that may be covered by non-terminal symbols (span constraint). As a result, long-range reordering represents a challenge also for HSMT. This problem has only recently been addressed by Braune et al. [2012], who propose to relax the span constraint only for a specific subset of the SCFG rules which are more likely to capture long reordering patterns in German-English.

We conclude this section by describing two SMT frameworks that couple together tree-based and string-based aspects: namely, discontinous phrase-based SMT [Galley and Manning, 2010] and the so-called "context-free reordering, finite-state translation" approach by Dyer and Resnik [2010].

In order to combine the efficiency of string-based decoding with the benefits of discontinuous phrases modeled in HSMT, Galley and Manning [2010] propose a method to extract phrases with gaps from word aligned parallel data, and modify a standard *string-based* PSMT decoder to support them. The resulting system is shown to significantly outperforms both a PSMT and an HSMT system on a Chinese-English task. Long-range reordering is however not addressed by this work, who adopts a standard distortion limit of 6 words.

Dyer and Resnik [2010] aim instead at combining the efficiency and modeling flexibility of string-based decoding with the powerful reordering mechanisms of syntax-based SMT. They use the syntactic parse tree of the input sentence to define the space of possible reorderings – represented as a reordering forest – but perform the actual translation with string-based models.

2.4 Word reordering as pre-processing

Given the difficulties of solving word reordering during the decoding process, a productive line of research has focused on decoupling reordering decisions from translation decisions. These approaches typically aim at arranging the source sentence in a target-like order *before* translating it. Thus, word reordering is solved as pre-processing in a monolingual fashion: i. e. *pre-ordering*.

Different strategies have been proposed: **deterministic pre-ordering** aims at finding a single optimal reordering for each input sentence, which is then translated monotonically [Xia and McCord, 2004] or with a low DL [Collins et al., 2005, Costa-jussà and Fonollosa, 2006, Habash, 2007, Tromble and Eisner, 2009, Genzel, 2010, Gojun and Fraser, 2012]; **non-deterministic pre-ordering** encodes multiple alternative reorderings into a word lattice and lets a monotonic decoder choose the best path according to its models [Zhang et al., 2007, Crego and Habash, 2008, Elming and Habash, 2009, Niehues and Kolss, 2009]. Both kinds of methods are conceived as alternatives, rather than enhancements, to standard PSMT reordering.

As for pre-ordering rules, they can be manually written [Collins et al., 2005, Gojun and Fraser, 2012] or automatically learned from syntactic parses [Xia and McCord, 2004, Habash, 2007, Elming and Habash, 2009, Genzel, 2010], shallow syntax chunks [Zhang et al., 2007, Crego and Habash, 2008] or part-of-speech labels [Niehues and Kolss, 2009]. In [Costa-jussà and Fonollosa, 2006], pre-ordering is learnt by training a monolingual PSMT system on a parallel corpus of original-to-preordered source sentences. In [Tromble and Eisner, 2009], pre-ordering is cast as a permutation problem and solved by a model that estimates the probability of reversing the relative order of any two input words.

Some of these works are particularly relevant for our thesis because they specifically address reordering between Arabic and English. Habash [2007] extracts reordering rules from a word-aligned parallel corpus with full parses on the source side. The rules reorder syntactic constituents and are applied in a deterministic way (always the most probable) to preprocess the training and test data. The technique achieves consistent improvements only in very restrictive conditions: maximum phrase length of 1 and monotonic decoding. The use of shallow syntax-based reordering rules, first proposed by Zhang et al. [2007] for a Chinese-English system, is applied to Arabic-English by Crego and Habash [2008]. In both works, the source permutations generated by the rules are represented in a lattice, which is then processed by a monotonic phrase- or n-gram-based decoder. Elming and Habash [2009] follow a similar approach for English-Arabic, but with rules learnt from full syntactic parses.

The German language was also addressed in several works. Collins et al. [2005] propose a set of six rules aimed at arranging the German sentence in an English-like order. The rules address in particular the position of verbs, verb particles and negation particles, and they are applied on full parse trees. Following a similar approach, Gojun and Fraser [2012] develop a set of rules for the opposite translation direction. Both works achieve significant improvements in terms of BLEU. From a manual analysis of their system, though, Gojun and Fraser [2012] report that about 10% of the English clauses were wrongly pre-ordered, mostly due to parsing errors.

Two works have explicitly challenged the reliability of syntactic parses for pre-ordering, particularly in Arabic-English: Green et al. [2009] analyzed two state-of-the-art parsers [Bikel, 2004, Klein and Manning, 2003] and reported F-measures of only 55-56% at the sub-task of detecting Arabic NP subjects in verb-initial clauses. Similar results were observed by Carpuat et al. [2010] using a dependency parser [Nivre et al., 2006]. The same paper also showed that the correct pre-ordering could not be safely predicted even from correct parses.

Rather than relying on supervised parsers trained on golden treebanks, two recent works have induced specific parsers directly from non-annotated parallel texts [DeNero and Uszkoreit, 2011, Neubig et al., 2012]. In these works, source sentence reorderings are first inferred from the word alignments with the target translation. Then, a parsing model is trained to maximize the likelihood of source trees that can generate such reorderings. Evaluated on the English-Japanese language pair, these methods almost reach the performance of a pre-ordering method that employs a supervised parser.

In alternative to the pre-processing approach, a smaller line of research has instead focused on reordering the target output *after* a monotonic translation process (e. g. Bangalore and Riccardi [2000], Sudoh et al. [2011]) or on re-scoring a set of n-best translation candidates produced by a medium-distortion PSMT system – for instance by means of POS-based reordering templates ([Chen et al., 2006]) or word-class specific distortion models [Gupta et al., 2007].

2.5 Evaluating word reordering

Automatically evaluating translation quality is a complex problem because there are innumerable ways to correctly render the same source sentence's meaning in the target language. Therefore, SMT systems are generally judged on the extent to which their outputs resemble a set of reference translations produced by different human translators. Despite relying on a very rough approximation of language variability, this approach provides SMT researchers with fast automatic metrics that can guide, at least in part, their steps towards improvement. Besides, fast evaluation metrics are used to automatically tune SMT feature weights on a development corpus, by means of minimum error rate training (MERT) procedures [Och, 2003].

While BLEU remains a widespread choice for both system evaluation and optimization, countless other measures have been proposed to overcome its many limitations. The design of MT evaluation metrics correlating with human judgements is an active research area, promoted for instance by the Workshop on Statistical Machine Translation [Callison-Burch et al., 2010]. In this thesis we adopt two general-purpose metrics (BLEU and METEOR) and a reordering-specific metric (KRS).

BLEU (Bilingual Evaluation Understudy metric) [Papineni et al., 2001] is a lexical match-based score that represents a de-facto standard for SMT evaluation. Here, proximity between candidate and reference translations is measured in terms of overlapping n-grams, with n typically ranging from 1 to 4. For each n a modified precision score $p(n)$ is computed on the whole test set and combined in a geometric mean.[5] The resulting score

[5] See Papineni et al. [2001] for more details on the computation of modified (or clipped) precisions.

(P_{BLEU}) is then multiplied by a brevity penalty (BP) that accounts for length mismatches between reference and candidate translations:

$$BLEU = P_{\text{BLEU}} \cdot BP \tag{2.12}$$

$$BP(ref, out) = \begin{cases} 1 & \text{if } |out| > |ref| \\ \exp(1 - \frac{|ref|}{|out|}) & \text{otherwise} \end{cases} \tag{2.13}$$

Using n-grams is a limited solution to the problem of word ordering evaluation: First, because only perfect surface matches are counted, without any morphology or synonymy notion. Second, because the absolute positioning of words in the sentence is not captured, but only their proximity within a small context.

The former issue is addressed, at least to some extent, by **METEOR** (Metric for Evaluation of Translation with Explicit ORdering metric) [Banerjee and Lavie, 2005], which relies on language-specific stemmers and synonymy modules to go beyond the surface-level similarity. The core formula of METEOR is the harmonic mean (F_{mean}) of word-level precision and recall computed over a word alignment between the hypothesis and reference translations. As for word order, METEOR treats it separately with a fragmentation penalty $FragP$. That is:

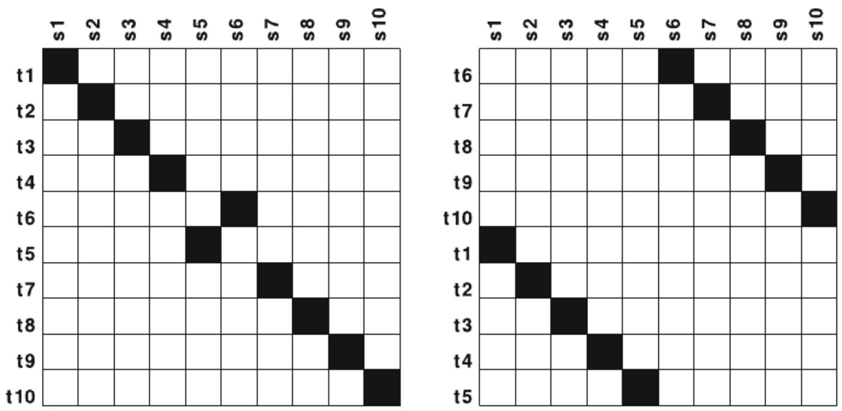
$$METEOR = F_{\text{mean}} \cdot (1 - FragP) \tag{2.14}$$

$$FragP = \gamma \left(\frac{ch}{m}\right)^{\beta} \tag{2.15}$$

where ch is the smallest number of chunks that the hypothesis must be divided into to align with the reference translation; m is the number of words matched by the alignment; β and γ are free scaling parameters. This measure too is poorly sensitive to word reordering errors because it only counts breaks in the word order with no distinction between short and long-range reordering.

The weakness of BLEU and METEOR with respect to word order was demonstrated by Birch et al. [2010] with a significant example that we report in Table 2.3. For simplicity, the example assumes that the reference order is monotonic and that hypotheses and reference translations contain exactly the same words. According to both metrics, the hypothesis (a) is worse than (b), although in (a) only two adjacent words are swapped while in (b) the two halves of the sentence are swapped.

To overcome these limitations, we complement our evaluation with a metric specifically


s1 s2 s3 s4 s5 s6 s7 s8 s9 s10
t1 t2 t3 t4 t6 t5 t7 t8 t9 t10
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10
t6 t7 t8 t9 t10 t1 t2 t3 t4 t5

(1 2 3 4 • 6 • 5 • 7 8 9 10)

Example **(a)**

(6 7 8 9 10 • 1 2 3 4 5)

Example **(b)**

Example:	**(a)**	**(b)**
BLEU	61.80	81.33
METEOR	86.91	92.63

Table 2.3: Two example alignments and their respective BLEU and METEOR scores, assuming that the reference alignment is monotonic. The permutation resulting from the hypothesis alignment is reported under each matrix, where bullet points represent jumps between non-sequential indices. Taken from Birch et al. [2010].

designed to measure word order differences. Following the approach proposed in the same paper [Birch et al., 2010], we measure the similarity between the reorderings needed to reach the reference translations from the source sentence and those applied by the decoder to produce the candidate translation. In practice, this is done by converting word alignments to permutations and computing a permutation distance among them. Among the metrics proposed in their paper, we chose the square root of the Kendall's Tau, as this was shown to be reliable and highly correlated with human judgements.

The normalized Kendall's Tau distance K is originally a measure of disagreement between rankings. Given a set of n elements and two permutations π and σ, the K distance corresponds to the number of discordant pairs (i.e. pairs of elements whose relative ordering differs in the two permutations) normalized by the total number of ordered element pairs:

$$K(\pi,\sigma) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{d}(i,j)}{\frac{1}{2}n(n-1)}$$

$$\text{where} \quad \mathbf{d}(i,j) = \begin{cases} 1 & \text{if } \pi_i < \pi_j \text{ and } \sigma_i > \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

Birch et al. [2010] further suggested to extract the square root of K to obtain a function that is more discriminative on lower distance ranges, i.e. for translations that are closer to the reference word ordering. Finally, the **Kendall Reordering Score (KRS)** – a positive measure of quality ranging from 0 to 1 – is computed by substracting the latter quantity from one, and by multiplying the result by a brevity penalty (BP) that accounts for length mismatches between reference and candidate translations:

$$KRS(\pi, \sigma) = (1 - \sqrt{K(\pi, \sigma)}) \cdot BP$$

The BP definition corresponds to that of BLEU, with the difference that this one is computed at the sentence level. In case of multiple references, the one that yields the highest score is retained for each test sentence. Finally, the average of all sentence-level KRS scores gives the global KRS of the test set.

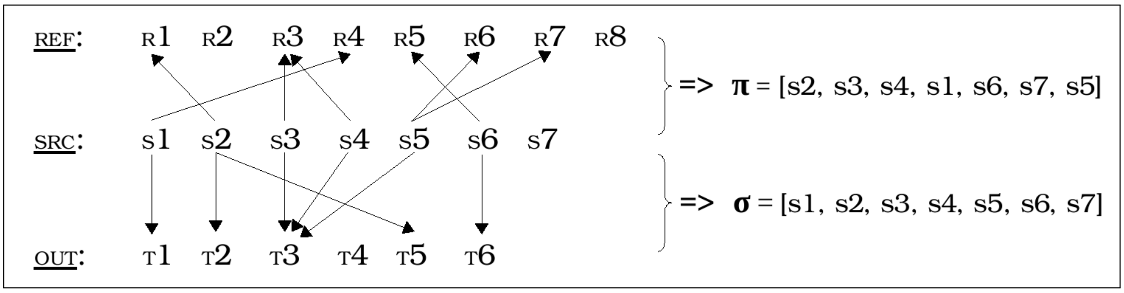


Figure 2.4: Example of KRS computation showing how word alignments are converted to permutations.

Figure 2.4 illustrates an example of KRS computation. A source sentence composed of 7 words (in the center) is aligned to a reference translation of 8 words and to an MT output of 6. The two word-alignment sets are converted to permutations, as shown in the right side of the figure, according to the following rules: (i) multiple source words aligned to the same target word are considered to be in monotonic order, (ii) non-aligned source words are assumed to immediately follow the previous source word, and (iii) if a source word is aligned to non-adjacent words in the target, only the first alignment is retained. Thus, for example, the link S2-T5 is ignored, while the word S7 is inserted right after S6 in the resulting permutation σ. The discordant pairs between π and σ are: (s1,s2),(s1,s3),(s1,s4),(s5,s6),(s5,s7), hence:

$$K(\pi, \sigma) = \frac{5}{\frac{1}{2}7(7-1)} = 0.2381 \quad , \quad BP(\pi, \sigma) = \exp\left(1 - \frac{8}{6}\right) = 0.7165$$

$$\text{and} \quad KRS = (1 - \sqrt{0.2381}) \cdot BP = 0.3669$$

Going back to the example of Table 2.3, hypothesis (a) would rightly obtain a much higher KRS than (b): that is 0.8509 versus 0.2546.

In a related work, Isozaki et al. [2010] proposed to directly measure the reordering occurring between the words of the hypothesis and those of the reference translation. A weakness of this approach is that only identical words contribute to the score. To address this problem, the permutation distance is then multiplied by a word precision score that penalizes hypotheses containing few reference words. Still, the resulting metric assigns different scores to hypotheses differing in their lexical choice, but not in their word reordering. On the contrary, KRS is robust to lexical choice because it does not directly compare output and reference *words* but only the *positions* of their translations. For this reason, we find it more suitable to specifically evaluate the reordering aspect of SMT.

2.6 Open issues

Translating words in the correct order is essential to preserve the meaning of a sentence. For instance, taking English as the target language, it is precisely the relative positioning of the predicate arguments that determines their role, in the absence of case markers. Thus, a wrongly reordered verb with minor impact on automatic scores, can be judged very badly by a human evaluator.

Although many solutions have been proposed to explicitly model word reordering during decoding, PSMT still largely fails to handle long-range word movements in language pairs with different syntactic structures.[6]. We believe this is mostly not due to deficiencies of the existing reordering *models*, but rather to a very coarse definition of the reordering search *space*. Indeed, the reordering constraints that are currently used to limit the space of explorable input permutations, are rather simple and typically based on word-to-word distances. For instance, the distortion limit constraint used by the open-source toolkit Moses is not sensitive to the actual words being translated, nor to their context, but is uniform throughout the input sentence. Relaxing this kind of constraints means dramatically increasing the size of the search space, making the reordering model's task extremely complex and intensifying the risk of model errors. As a result, even in language pairs where long reordering is regularly observed, PSMT quality degrades when long word movements are allowed to the decoder – typically, when the DL is higher than 6 words. Indeed, decent performances are obtained with a low or medium DL, but this obviously comes at the expense of long reorderings, which are often crucial to preserve the general meaning of a translated sentence.

On the other hand, tree-based SMT methods have a radically different approach to

[6] For empirical evidence, see for instance Birch et al. [2009], Galley and Manning [2008].

the word reordering problem. While being more linguistically motivated than PSMT, these methods fail to outperform it on many language pairs, including Arabic-English (see for example Birch et al. [2009]). This can be due to different factors: in the hierarchical framework, reordering information is totally lexicalized, thus the presence of words triggering reordering is essential for the success of these models. Moreover, the HSMT reordering space is typically restricted by hard constraints that improve efficiency and translation quality overall, at the expense of long-range word movements. Thus, HSMT can suffer from similar problems as PSMT.

Alternatively, reordering can be solved by relying on a syntactic parse tree of the input. In this case the reordering space corresponds to only those permutations that can be generated by permuting the children of each node in the tree. This strategy is adopted by syntax-based SMT methods, where the tree is reordered and translated simultaneously, and by syntactic pre-ordering methods, where the tree is transformed before translation. The success of these approaches depends on the degree of isomorphism of the modeled language pair, and also on the parser's performance, which can vary substantially across languages.

To address these problems, we adopt the PSMT framework and investigate various techniques to refine its reordering search space based on prior linguistic knowledge. As a difference from pre-ordering, we do not take hard reordering decisions before the actual translation process, as these naturally interact with the other aspects of translation. By leaving the final reordering decision to the decoder, we can thus take advantage of the most recent advances in reordering modeling. Finally, considering that parsing resources are still scarce and inaccurate in many languages, we only employ other kinds of annotation – like part-of-speech and shallow syntax – which are simpler to automate, less prone to errors, and available in more languages.

In the following chapter, we will analyze the word reordering characteristics of various language pairs, based on a large body of theoretical linguistic knowledge.

Chapter 3

Word Reordering in Different Language Pairs

Developing a methodology to predict the complexity of reordering in a given language pair is key to selecting the right SMT models and to improving them.

To date, word reordering phenomena have mainly been analyzed from a quantitative perspective [Birch et al., 2008, 2009]. We argue that, besides measuring the *amount* of reordering, it is important to understand which *kinds* of reordering occur in a given language pair. In this chapter, we present a qualitative analysis of word reordering based on linguistic knowledge, which can guide the choice of a suitable SMT approach. To this end, we consider a large body of syntactic information collected from more than 1500 languages, and systematized in the World Atlas of Language Structures (WALS) [Dryer and Haspelmath, 2011].

Following the seminal work of Matthew S. Dryer, we describe the word order profile of a language by the canonical orders of some of its constituent sets (word order features). The resulting language pair classification is primarily based on the order of subject, object and verb, and further refined according to the order of several other element pairs, such as noun-adjective, verb-negation, etc. We then compare the word order features of several languages that were studied in the SMT field, and show that empirical results generally confirm existing theoretical knowledge.

3.1 A qualitative analysis

The amount of word reordering found in a language pair is known to be a good predictor of SMT performance. Birch et al. [2008] considered three variables – reordering quantity, morphological complexity and historical relatedness – and found the first to have the highest correlation with the BLEU scores of a standard PSMT system, over a sample of 110 European language pairs. Birch et al. [2009] further analyzed the distribution of different reordering widths in Arabic-English and Chinese-English, and the ability of two SMT approaches to model them. They found that the PSMT approach is more suitable for language pairs where most reordering is local (Arabic-English), while the hierarchical approach is stronger when medium-range reorderings are dominant (Chinese-English). Still, both approaches failed to capture most of the long-range reorderings found in the reference corpora.

These findings are indeed relevant to our work, but we believe there is also much to learn from theoretical linguistic knowledge. Moreover, a quantitative analysis can suffer from noise in the data, typically originating from automatic word alignments.[1] Noise can also be due to what we could call "unnecessary" reordering. In fact, human translators can choose to restructure the sentence according to their personal taste, or to accomodate style and conventions of the target language. Here is an example:

Arabic sentence:

و طمأن بوش (55 سنة) الصحافيّن قبيل مغادرته البيت الابيض الى انه يشعر بانه في
حال ” رائعة “ و صحة ” جيدة جدا “ .

Literal translation:

Bush, aged 55, assured journalists before leaving the White House that he felt "great" and that his health was "very good".

Human translation:

Before leaving the White House, Bush, aged 55, assured journalists that he felt "great"and that his health was "very good".

This kind of reordering is not strictly necessary to produce accurate and fluent translations, but its occurrence in parallel corpora affects the automatic reordering measures.

On the contrary, a qualitative analysis can profit from the extensive work done by linguists and grammaticians to abstract the fundamental properties of a language. In this chapter, we draw largely on Dryer [2007] and on the sections of the WALS devoted to word order (Dryer [2011], ch. 81-97, 143-144).

[1] Birch et al. [2009] used manual word alignment in their study, but this kind of resource is available only for very few language pairs.

3.2 Word order profiles

The word order profile of a language is determined by the canonical order of its constituent sets (word order features). In general, the basic or canonical order of a constituent set can be established by the criteria of frequency (i. e. the most common), distribution (the one with the least restricted usage) or pragmatics (the neutral one) [Dryer, 2007]. Although many languages are said to have free (or flexible) order, it is often possible to detect one that is dominant and neutral. Consider for instance English, a subject-verb-object (SVO) language where other orders are used, but only to achieve specific emphasis or topicalization effects.

(1) a. I saw the cat.

b. The cat, I saw.

However, there exist cases where no particular order can be defined as dominant. An example of mix-ordered constituent set in English is the pair noun and genitive.

(2) a. the tail of the cat

b. the cat's tail

Based on Dryer [2007] and on the availability of data points in the WALS, we have established a set of 13 core features to determine the word order profile of a language. For the purpose of describing word order differences between language pairs, we have divided the features in two broad categories: clause-level and phrase-level[2].

3.2.1 Clause-level order features

- **Subject, Object, Verb** [WALS feature 81A]
  The first and most important feature consists of the "ordering of subject, object, and verb in a transitive clause, more specifically declarative clauses in which both the subject and object involve a noun (and not just a pronoun)" [Dryer, 2011]. For instance, English and French are SVO languages, while Turkish is SOV. The distribution of main word order types in a large sample of world languages is given in Table 3.1. This feature is often used alone to denote the word order of a language, because it can be a good predictor of many other features.

[2] Here, phrase is used in its traditional syntactic sense – a group of words forming a constituent – as opposed to the notion of data-driven phrase adopted by phrase-based SMT (cf. n-gram).

Order	Languages	
SOV	565	41%
SVO	488	35%
VSO	95	7%
VOS	25	2%
OVS	11	1%
OSV	4	<1%
mixed/no-dominant	189	14%
total sample size	1377	

Table 3.1: The distribution of main word order types (Verb, Subject and Object) in the world languages. From the World Atlas of Language Structures, chapter 81 [Dryer, 2011].

- **Oblique or Adpositional Phrase** [84A]
  This feature refers to the position of a phrase functioning as adverbial modifier of the verb, relative to the position of object and verb. For instance, English is VOX because it places oblique phrases after verb and subject.
- **Noun and Relative Clause** [90A]
  The location of the relative clause with respect to the noun it modifies.
- **Adverbial Subordinator and Subordinate Clause** [94A]
  Subordinators are used to link adverbial subordinate clauses to the main clause. They can take the form of verbal suffixes or separate words, such as the English subordinating conjunctions 'when' and 'because'.
- **Polar Question Particle** [92A]
  In many languages, polar or *yes-no* questions are signaled by specific particles. This feature denotes their position in the sentence (not defined for English).
- **Content Question Phrase** [93A]
  As opposed to polar questions, content questions are characterized by the presence of an interrogative word or phrase (e.g. 'who', 'which one'). In some languages, like English, these are always placed at the beginning of the sentence. In some others, like Turkish, they take the position of the constituent they replace: for instance, the word '*ne*/what' replacing the object naturally occurs between subject and verb.
- **Negation and Verb** [143A]
  The position of the negative morpheme[3] with respect to the *main* verb. Note that more than one morpheme may be necessary to express negation (e. g. in French).

[3] Differently from the WALS, we do not distinguish between negative words and affixes for this feature.

3.2.2 Phrase-level order features

- **Noun and Adpositions** [WALS feature 85A]
  Whether a language uses mainly prepositions or postpositions.
- **Noun and Genitive** [86A]
- **Noun and Adjective** [87A]
- **Noun and Demonstrative** [88A]
- **Noun and Numeral** [89A]
- **Adjective and Degree Word** [91A]

3.2.3 Language sample

For our study, we have chosen six widely spoken languages[4] representing various language families and very different word order profiles. These are English, German, French, Arabic (Modern Standard), Turkish and Chinese (Mandarin). Mainly based on the WALS, we have summarized the word order feature values for all these languages in Table 3.2. Whenever possible, features were assigned one (or two) values corresponding to the dominant order(s) in that language. When no particular order was given as dominant we marked it as 'mixed'.

The main word order of German and Arabic deserves a special mention. In German, the positioning of subject, object and verb is syntactically determined: main clauses with no auxiliary verb are SVO, while subordinate clauses and clauses containing an auxiliary are SOV. A further complication, not shown in Table 3.2, is that the German finite verb must be placed in second position, in which case the pattern becomes S*Aux*OV, with the object intervening between auxiliary and main verb. As regards Arabic, while the WALS classifies Modern Standard Arabic as VSO, the corpora typically used in SMT studies show a very mixed distribution of VSO and SVO clauses.[5] Carpuat et al. [2012] examined the Arabic-English Treebank[6] and found that, when the subject is expressed, it follows the verb in 70% of the cases, but precedes it in 30%. Similarly, in the Pennsylvania Arabic Treebank[7], they found an order distribution of 67% VS and 33% SV. Besides

[4] More than 50 million native speakers each, according to Wikipedia: http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers.

[5] VOS order is also admitted in Arabic, but only in specific contexts (e. g. when the object is expressed by a pronoun).

[6] Corresponding to the Linguistic Data Consortium (LDC) code LDC2009E82.

[7] LDC code LDC2008E22.

		Indo-European			*Afro-Asiatic*	*Altaic*	*Sino-Tibet.*
	Features	**English**	**German**	**French**	**Arabic**	**Turkish**	**Chinese**
Clause-level	Subject,Object,Verb [Tom] [chases] [Jerry]	SVO	SVO/ SOV	SVO	VSO/ SVO*	SOV	SVO
	Oblique Phrase [chases] [Jerry] [with a stick]	VOX	mixed	VOX	VOX	XOV	XVO
	Noun,RelClause [a stick] [that he stole]	N-Rel	N-Rel	N-Rel	N-Rel*	Rel-N	Rel-N
	Subordinator,Clause [because] [he was hungry]	Sub-C	Sub-C	Sub-C	Sub-C	C-Sub/ Sub-C	mixed**
	PolarQuest.Particle ∅ [did he steal it?]	*none*	*none*	initial	initial	final	final
	ContentQuest.Phrase [why] [did he steal it?]	initial	initial	initial	initial*	other	other
	Negation,Verb he did [not] [steal]	Neg-V	Neg-V/ V-Neg	Neg-V-Neg/ V-Neg	Neg-V	V-Neg	Neg-V
Phrase-level	Noun,Adpositions [with] [a stick]	Adp-N	Adp-N	Adp-N	Adp-N	N-Adp	N-Adp/ Adp-N
	Noun,Genitive [Tom's] [stick]	N-Gen/ Gen-N	N-Gen	N-Gen	N-Gen	Gen-N	Gen-N
	Noun,Adjective [hungry] [Tom]	A-N	A-N	N-A	N-A	A-N	A-N
	Noun,Demonstrative [this] [stick]	Dem-N	Dem-N	Dem-N	Dem-N	Dem-N	Dem-N
	Noun,Numeral [two] [sticks]	Num-N	Num-N	Num-N	Num-N	Num-N	Num-N
	Adjective,DegreeW. [very] [hungry]	Deg-A	Deg-A	Deg-A	A-Deg	Deg-A	Deg-A
	Feature	**English**	**German**	**French**	**Arabic**	**Turkish**	**Chinese**

Table 3.2: The word order profile of six world languages. Sources: the World Atlas of Language Structures [Dryer and Haspelmath, 2011], (*) personal knowledge, and (**) Li [2008].

frequency, it can be noted that the SVO sentences attested in these corpora are in general pragmatically neutral. We believe that this variability in Modern Standard Arabic may be due to the effect of spoken language varieties such as Egyptian, Gulf, Kuwaiti, Iraqi (all listed as SVO by the WALS), and Syrian (listed as VSO/SVO). For these reasons, we classify Arabic as a mixed VSO/SVO language.

It is worth noting that our six-language sample covers the main word order types of the large majority of the world languages: namely, SOV, SVO and VSO (see Table 3.1).

3.3 Word order differences

Linguistically motivated word order profiles can be very helpful to predict the kind of word reordering problems that an SMT system will have to face. Clearly, these will also vary in relation to the genre (written news, speeches, etc.) and to the translation's style and degree of literality. However, we can reasonably expect the syntactic properties of two languages to determine the general reordering characteristics of that pair.

We will now confront the reordering characteristics of six language pairs: English paired with the other five languages presented in Table 3.2, as well as the French-Arabic pair.[8] To this end, we propose the following analysis procedure. As a first indication of reordering complexity, we look at the main word order feature (subject, object, verb). A difference at this level typically results in poor SMT performances. Then, we count the total number of discordant features. To simplify, if a particular element does not exist in a language (e. g. polar question particles in English) we count *zero* difference for that feature, and when one of the languages has a mixed order we count a *half* difference. We insist, however, on the qualitative nature of our analysis: numbers are only meaningful in combination with the list of specific discordant features, as these have different impact on word reordering. In particular, we find it essential for SMT to distinguish between clause-level and phrase-level differences (CDiff and PDiff) because the former account for most longer-range word movements, and the latter for the shorter. Thus, a language pair with only phrase-level discordant features is likely to be suitable for a PSMT approach, where reordering is managed through local distortion or inside translation units (phrase pairs). Conversely, the presence of many clause-level differences calls for a tree-based solution, either at preprocessing or at decoding time. As we will see, some pairs lay on the borderline, by displaying only one or few clause-level differences. Finally, it should be noted that, even among features of the same group, some have more impact on SMT than

[8] The language pair direction (e. g. French-Arabic or Arabic-French) is irrelevant for the purposes of this chapter.

others due to their frequency or to the average length of their constituents. For instance, the order of noun and genitive is more important than that of adjective and degree word.

English and German [Main order: different; CDiff=1.5; PDiff=0.5]

The main word order of German is SVO or SOV according to the syntactic context (cf. Section 3.2). German also differs from English with respect to the position of oblique phrases and that of the negation: both fixed in English but mixed in German. At the phrase level, German predominantly places the genitive after the noun, while English displays both orders.

Thus, despite belonging to the same family branch (Indo-European/Germanic), this pair is characterized by at least some complex reordering patterns. Indeed, German-English reordering has been widely studied in SMT and is still an open topic. Looking at the results of the Workshop of Machine Translation's last edition (WMT12) [Callison-Burch et al., 2012], no particular SMT approach appears to be winning. In both language directions (official results excluding the online systems) the rule-based systems outperformed all SMT approaches, and among the best SMT systems we find a variety of approaches: pure phrase-based, phrase-based and hierarchical systems combination, n-gram based, a rich syntax-based approach, and a phrase-based system coupled with POS-based pre-ordering. This gives an idea of how challenging this language pair is for SMT.

English and French [Main order: same; CDiff: 0.5; PDiff: 1.5]

Most clause-level features have the same values in French as in English, except for the negation which is typically expressed by two words in French: one preceding and one following the verb. At the phrase level, differences are found in the location of genitives and adjectives. Thus, English and French have very similar clause-level orders, but reordering is abundant at the local level.

As a reference, among the three top English-to-French WMT12 systems (official results excluding online and rule-based systems), two were phrase-based and one was hierarchical. The same thing was observed in the French-to-English track.

English and Arabic [Main order: different; CDiff: 0.5; PDiff: 2.5]

The dominant Arabic order is VSO, followed by SVO (cf. Section 3.2). Apart from this important difference, all other clause-level features agree between Arabic and English. At the phrase level, differences are found in genitives, adjectives and degree words.

As a result, reordering is overwhelmingly local but few crucial long-range reorderings also regularly occur. This pair is, in fact, challenging for PSMT but, at the same time, not well suited to a tree-based approach. As shown by Zollmann et al. [2008] and Birch et al. [2009], PSMT performs similarly or better than hierarchical SMT for the Arabic-to-English language pair.

English and Turkish [Main order: different; CDiff: 5.5; PDiff: 1.5]

Turkish is a typical example of verb-final language, except for the fact that it can employ both clause-final and clause-initial subordinators.[9] As a result, almost all clause-level features are discordant in this pair. At the phrase level, Turkish mainly differs from English for the use of postpositions.

Among the language pairs of our sample, this is the most difficult to reorder for an SMT system. The complex nature of its reordering phenomena makes it particularly suitable for tree-based SMT approaches. Although this language pair has been less studied than the others, we know from Ruiz et al. [2012] that hierarchical SMT can significantly outperform PSMT on Turkish-to-English.

English and Chinese [Main order: same; CDiff: 3.5; PDiff: 1]

Despite belonging to the same main order type, these two languages differ in the positioning of oblique phrases, relative clauses, interrogative phrases and subordinating words. The latter can in fact occur at the beginning of the subordinate clause, at its end, or even inside it [Li, 2008]. Comparing the two languages at the phrase level, we find partial disagreement in the use of genitive and adpositions (Chinese has both prepositions and postpositions).

Thus, this pair too is characterized by very complex reordering, hardly manageable by a PSMT system. This is confirmed by a number of empirical results showing that tree-based approaches (particularly HSMT) consistently outperform PSMT in Chinese-to-English evaluations [Zollmann et al., 2008, Birch et al., 2009].

French and Arabic [Main order: different; CDiff: 1.5; PDiff: 1]

This pair displays the same clause-level differences as the English-Arabic pair. On the other hand, the phrase stucture is notably more similar, with only one discordant feature of minor importance (adjective and degree word).

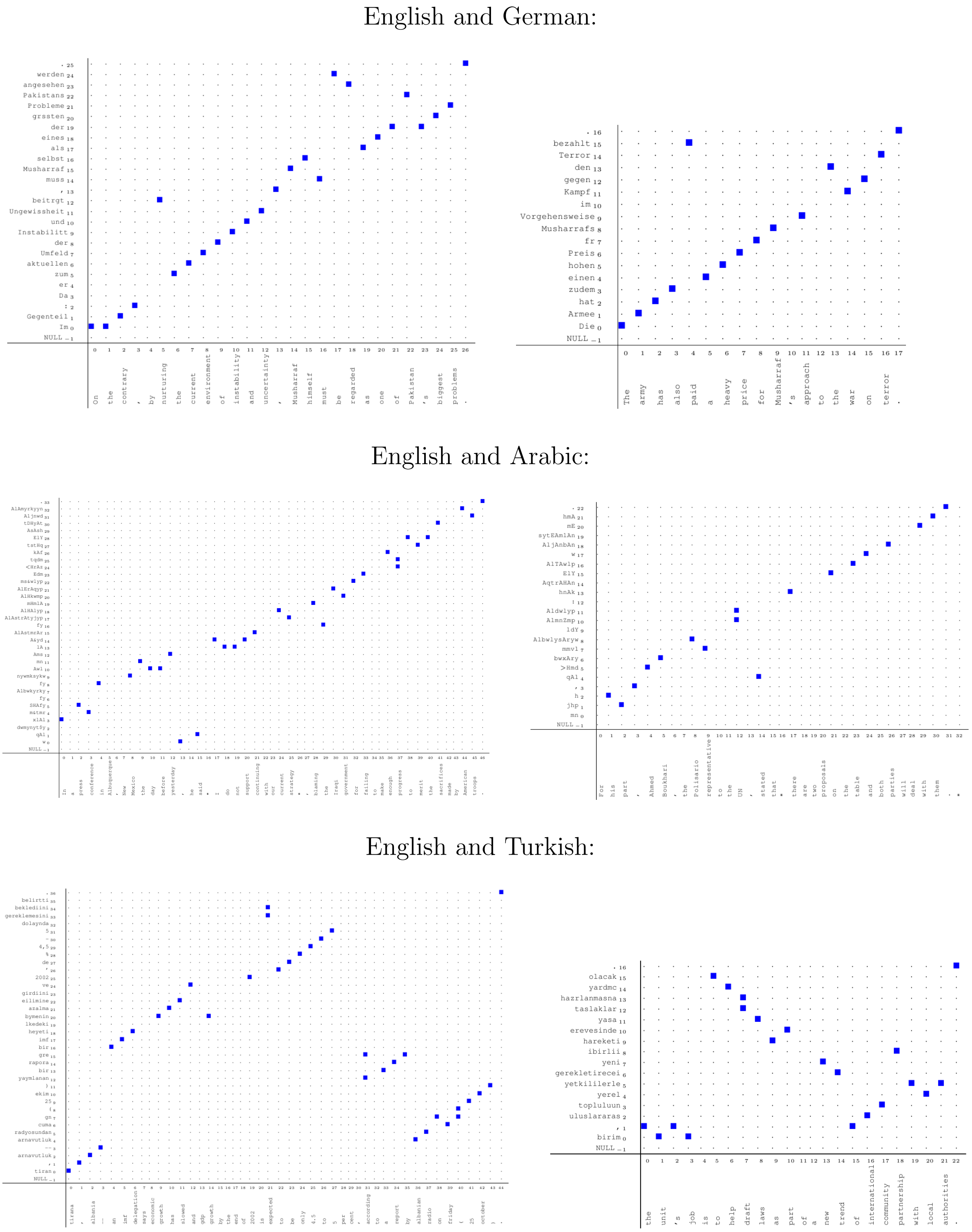
[9] In Turkish, non-finite subordinate clauses are typically placed before the main clause and linked to it by a clause-final subordinator (e. g. *rağmen/although*), whereas finite subordinate clauses can be placed after the main clause and introduced by a clause-initial subordinator (e. g. *ama/but*). The former case is by far more common.

> Little research was published on this language pair. To our knowledge the state of the art in Arabic-to-French is represented by Hasan and Ney [2008] and Schwenk and Senellart [2009], both of which employ a PSMT approach.

Figure 3.1 illustrates the reordering characteristics of three of our language pairs, by means of examples drawn from parallel corpora. On the first row, we can see two English-German sentence pairs: in both cases, most of the points lie close to the diagonal representing an overall monotonic translation, whereas the few isolated points stand for verbs that are placed in very different positions. Similarly, in the two English-Arabic sentence pairs, we mostly observe very local reorderings, with the exception of few isolated points corresponding to the Arabic clause-initial verbs. Finally, the two Turkish-English examples display massive and global reordering, due to the high number of clause-level order differences.

3.4 Conclusions

We conclude from our analysis that linguistic knowledge is indeed useful to predict the reordering characteristics of a language pair and to select the SMT approach that best suits them. In particular, language pairs with many clause-level order differences (Turkish-English and Chinese-English) are best handled by tree-based SMT approaches that can handle complex, hierarchical reordering patterns. On the other hand, PSMT is preferable for language pairs with only phrase-level differences (French-English), as these mostly imply short or medium-range reordering patterns that can be captured by local distortion. Finally, the pairs with mostly phrase-level differences and only one or few clause-level differences (German-English and Arabic-English) do not fit into either category. In the absence of global reordering, tree-based SMT underperforms PSMT, likely due to a much larger search space. At the same time, applying PSMT to such pairs can lead to systematic errors in the positioning of specific constituents. The working hypothesis of this thesis is that these ‘borderline’ language pairs are best handled by a hybrid approach where local reorderings are captured by the regular PSMT reordering mechanisms while long reordering patterns are treated by specific techniques. In the next chapters, we will propose different ways to achieve this.



English and German:

English and Arabic:

English and Turkish:

Figure 3.1: Word-alignment matrices of sentence pairs taken from three parallel news corpora: the NIST-MT-08 Arabic-English evaluation benchmark, the WMT-10 German-English training corpus, and the Turkish-English South European Times corpus. English is always on the x axis.

Chapter 4

Chunk Reordering Lattices

Reordering lattices serve to represent multiple permutations of the input sentence. We use them to suggest likely long reorderings to the decoder, while short and medium-range reordering is handled by decoding-time distortion.

4.1 Introduction

As discussed in the introduction and in Chapter 3, the Arabic-English language pair is characterized by an uneven distribution of reordering phenomena: namely, frequent local reorderings due to the head-initial structure of Arabic noun phrases, and isolated long reorderings of the main verb in Arabic VSO sentences.

In this chapter, we present our first method to improve the performance of a PSMT system with respect to this issue: chunk-based verb reordering lattices. We develop a simple set of shallow syntax-based rules to reorder clause-initial verbs in the Arabic side of a word-aligned parallel corpus (Section 4.2). This technique is used (i) to preprocess the training data by minimizing long reordering in the alignments, and (ii) to collect statistics about verb movements (Section 4.3). From this analysis, we build a word lattice representing a set of likely verb reorderings for each test sentence (Section 4.5). Translation is then performed by a lattice decoder which explores additional (local) reordering. As a result, the space of local reorderings defined by the standard constraints is augmented with few long reorderings encoded in the lattice. Lastly, because our lattices can be very dense, we devise a pruning technique based on a discriminative classifier (Section 4.6).

Related work [Zhang et al., 2007, Crego and Habash, 2008, Elming and Habash, 2009] mostly aimed at representing *all* word reorderings in the lattices, which were then processed by monotonic phrase- or n-gram-based decoders. These methods were, in fact, conceived as alternatives, rather than integrations, to the standard PSMT reordering models. We instead focus on a single type of significant long reordering, and improve a system that already includes state-of-the-art PSMT reordering models. To our knowledge, the only example of reordering lattices coupled with reordering at decoding time is the work by Niehues and Kolss [2009] on German-English. Their phrase-based decoder admits local reordering within a fixed window of 2 words, while we perform experiments up to a distortion limit of 10. Another major difference is that, while their rules are POS-based, we use shallow syntactic chunks to reduce the number of possible permutations.

The application of our chunk-based reordering methods to the training and test data results in consistent improvements on the NIST-MT 2009 Arabic-English benchmark, both in terms of BLEU score (+1.06) and of reordering quality measured with the Kendall Reordering Score (+0.85).

4.2 Chunk-based verb reordering

We performed a manual analysis on a random sample of 100 sentences drawn from the newswire parts of the NIST-MT09 training data. This set contains 51 verbs in pre-subject position (ignoring clauses where the subject pronoun is dropped), and in 10 of these cases the subject spans more than 6 tokens. This suggests that a special treatment of VSO constructions could benefit significantly PSMT, which is known to perform poorly on long-range reordering.

Our approach is two-fold: (i) reorder verbs in the source side of the parallel training data so that long reordering between the two languages is minimized. We expect this to benefit both word alignment and phrase extraction. (ii) Suggest likely verb reorderings to the decoder.

To model verb reordering we can employ different kinds of linguistic annotation. One option would be to use syntactic parse trees to detect the Arabic constituents, and then swap verb and subject in all VSO sentences. This kind of technique was adopted, for instance, by Green et al. [2009] and Carpuat et al. [2010]. Another option would be to use shallow syntactic chunks as the reordering blocks, and generate multiple verb reorderings by means of non-deterministic (fuzzy) rules.[1] Then, in a word-aligned parallel corpus, we could establish the optimal position of each verb chunk by minimizing distortion in the alignments. As for the test sentences, we could represent multiple reorderings with a word lattice and let the decoder choose the one that leads to an overall better translation. We

[1] Chunk annotation does not identify subject and complement boundaries, nor the relations among constituents that are needed to deterministically rearrange a sentence in SVO order.

choose the second option because it has several advantages: First, shallow syntax chunking is a simpler task than full parsing, therefore less prone to errors. Second, because multiple suggestions are provided to the decoder, all the SMT models can contribute to the final reordering decision. Third, by using word alignments as supervision for the training data, we avoid reordering verb-subject constructions that for some reason were not inverted in the English translation. Admittedly, we incur the risk of missing correct verb reorderings due to alignment errors, but we hope that this type of error on the training data will have a minor impact.

We propose a simple set of fuzzy chunk-based rules aimed at transforming VS(O) sentences into SV(O):

R1) move *the verb chunk* by up to M positions to the right;

R2) move *the verb chunk and the chunk following it* by up to M positions to the right.

The second rule addresses the case where the verb chunk (VC) needs to be moved along with an adverbial chunk or a complement. The maximum movement M is set empirically to ensure substantial coverage of the verb reorderings observed in parallel data (see Section 4.3). To prevent verb reordering from overlapping with the following clause, we also limit the maximum movement of a given VC to the position of the next VC found in the sentence.[2] Thus, for each VC our rule set generates $2 \times M$ reorderings, or less according to the context.

Given a word-aligned translation of the sentence, we define the optimal reordering as the one that minimizes the amount of distortion in the alignment, defined as the number of swaps of source words in target order. If no movement reduces the swaps found in the original order, then the verb is left at its original position. On the other hand, if several movements are found to satisfy this criterion, a second minimization is applied to the sum of distances between source positions aligned to consecutive target positions, i.e. $\sum_i |a_i - (a_{i-1} + 1)|$ where a_i is the index of the foreign word aligned to the i^{th} English word. These two conditions generally suffice to define a single best reordering, but when this is not the case, the shortest best movement is heuristically selected.

Figure 4.1 illustrates the process of chunk-based verb reordering in a word-aligned sentence pair. According to the alignments, the optimal rearrangement of the source sentence is obtained by moving the VC by 2 chunks to the right. In fact, among the reorderings admitted by the rules, this is the only one that reduces the number of swaps (from 4 to 3).

[2] In fact, this condition inhibits the reordering of clause-initial verbs whose subject contains a verb, e.g. in the case of relative clauses. However, this occurrence is rare in the data.

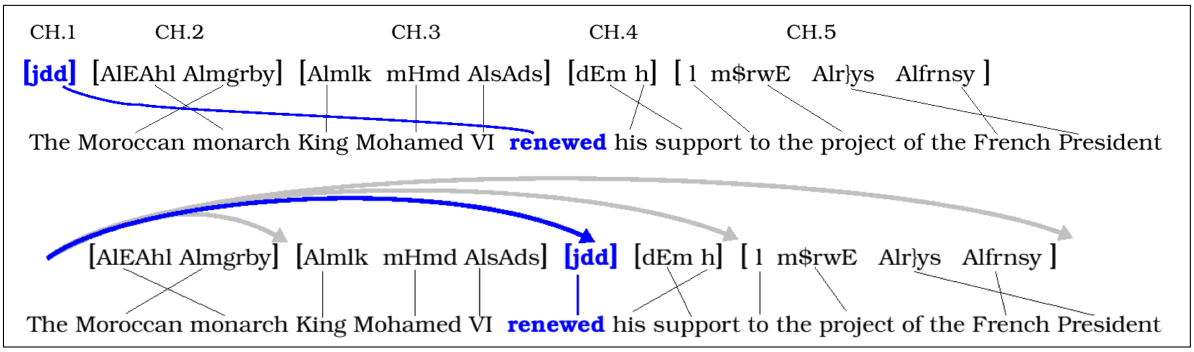

CH.1
CH.2
CH.3
CH.4
CH.5
[jdd] [AlEAhl Almgrby] [Almlk mHmd AlsAds] [dEm h] [l m$rwE Alr}ys Alfrnsy]
The Moroccan monarch King Mohamed VI renewed his support to the project of the French President
[AlEAhl Almgrby] [Almlk mHmd AlsAds] [jdd] [dEm h] [l m$rwE Alr}ys Alfrnsy]
The Moroccan monarch King Mohamed VI renewed his support to the project of the French President

Figure 4.1: Example application of chunk-based verb reordering to a word-aligned sentence pair. The chunk-based rules generate a set of likely verb reorderings (grey arrows), among which the one minimizing distortion in the alignment is chosen as optimal (bold arrow).

We manually re-inspected the 100-sentences sample after application of verb reordering. It appeared that 40 clause-initial verbs out of 51 were indeed moved after the subject. The remaining 11 were left in their original position because they were unaligned, which suggests that our reordering technique can be affected by alignment errors. In Section 4.4 we will measure whether verb reordering of the training is nevertheless helpful.

The proposed verb reordering technique is used to (i) perform a quantitative analysis of verb reordering, (ii) train a PSMT system on more monotonic alignments, and finally (iii) produce training examples for an SVM classifier that predicts likely verb reorderings.

4.3 Corpus analysis of verb reordering

We applied the above technique to two parallel corpora provided for the NIST-MT09 Arabic-English evaluation: the first corpus (gale-nw)[3] contains manual alignments; as for the second (eval08-nw)[4] automatic alignments were generated with Moses as the *Intersection* of the direct and inverse alignments computed by GIZA++ [Och and Ney, 2003]. The choice of such a high-precision, low-recall alignment set is supported by the findings of Habash [2007] on syntactic rule extraction from parallel corpora. Sentences longer than 80 tokens were filtered out to make word alignment feasible, resulting in 4337 (gale-nw) and 777 (eval08-nw) sentence pairs respectively.

Both corpora were preprocessed with the AMIRA toolkit [Diab et al., 2004] for morphological segmentation according to the ATB scheme[5] and for shallow syntax chunking.

[3] Newswire section of the LDC2006E93 data set.

[4] Newswire section of the LDC2009E08 data set.

[5] The Arabic Treebank tokenization scheme isolates conjunctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*.

The average length of the Arabic sentences, after morphological segmentation, is 32.2 words in gale-nw and 29.1 words in eval08-nw. The average length of the chunks is 1.6 words.

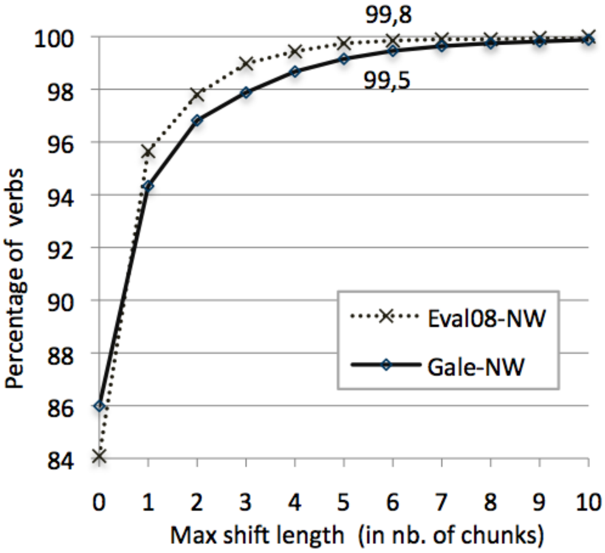


Figure 4.2: Percentage of verb reorderings by maximum shift (0 stands for no movement).

There are 1,955 verb chunks in gale-nw and 11,833 in eval08-nw. Among all verb chunks, 86% and 84% respectively do not need to be moved according to the alignments. The remaining 14% and 16% are distributed by movement length as shown in Figure 4.2: most verb reorderings consist of a 1-chunk jump to the right (8.3% in gale-nw and 11.6% in eval08-nw). The rest of the distribution is similar in the two corpora, which indicates a good correspondence between verb reordering observed in automatic and manual alignments. By increasing the maximum movement length from 1 to 2, we cover an additional 3% of verb reorderings, and around 1% when moving from 2 to 3. Recall that the length measured in chunks may not correspond to the number of jumped words. We use these figures to determine an optimal set of reordering rules: from now on we will focus on verb movements of at most 6 chunks, as these account for 99.5% of the observed occurrences.

We then performed another analysis to measure the impact of chunk-based verb reordering on the total word distortion observed in parallel data. For the sake of reliability, we only examined the manually aligned corpus (gale-nw). Figure 4.3 shows the positive effect of verb reordering on the total distortion, which is measured as the number of *words* that have to be *jumped* on the source side in order to cover the sentence in the target order (that is $|a_i - (a_{i-1} + 1)|$, see Section 4.2). Jumps have been grouped by length and

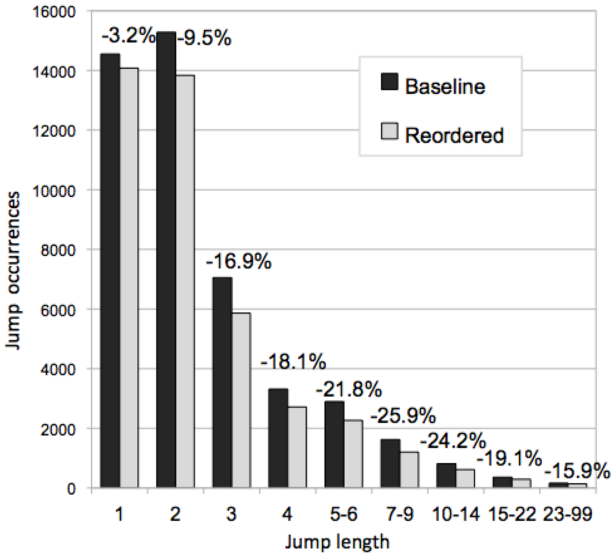


Figure 4.3: Distortion reduction in the gale-nw corpus: jump occurrences grouped by binned length (in number of words).

the relative decrease of jumps per length is shown on top of each double column.

These figures do not prove as we hoped that verb reordering resolves *most* of the long range reorderings. Thus we manually inspected a sample of verb-reordered sentences that still contain long jumps, and found out that several of these were due to what we could call "unnecessary" reordering. In fact, human translations that are free to some extent, often display a global sentence restructuring that makes distortion dramatically increase (cf. Section 3.1). We believe this phenomenon introduces noise in our analysis since these are not reorderings that an MT system needs to capture to produce a correct translation.

Nevertheless, the relative decreases shown in the plot suggest that, although short jumps are by far the most frequent, verb reordering affects especially medium and long-range distortion. More precisely, our selective reordering technique solves 21.8% of the 5-to-6-words jumps, 25.9% of the 7-to-9-words jumps and 24.2% of the 10-to-14-words jumps, against only 9.5% of the 2-words jumps, for example. Since our primary goal is to improve the handling of long reorderings, this lets us think that we are advancing in a promising direction.

4.4 Preliminary SMT experiments

In this section we investigate how verb reordering of the source language can affect translation quality. We apply verb reordering to both training and test data. While the parallel data can be reordered by exploiting word alignments, for the test set we need a verb reordering prediction model. We then assume that optimal verb reordering of the test is provided by an *oracle* that has access to alignment with the reference translations.

We first trained a Moses baseline on a subset of the NIST-MT09 training data[6] for a total of 981K sentences, roughly corresponding to 30M English words. We then used the resulting GIZA++ word alignments (*Intersection* set) to apply the technique explained in Section 4.2 and retrained the whole PSMT system – from word alignment to phrase scoring – on this reordered dataset. For the evaluation we used two different versions of eval08-nw: original and verb-reordered. In the latter, reordering was obtained from the alignment with the first English reference. With the first experiment we measure the impact of verb reordering on training only. With the second, we estimate the maximum improvement achievable by applying a verb reordering prediction technique to the test data.[7] In all experiments we used a typical Moses configuration including a 4-feature phrase translation model, a phrase and a word penalty, a 7-feature lexicalized phrase-orientation model [Och et al., 2004, Koehn et al., 2007] and a 6-gram language model trained on the English side of all the available NIST-MT09 parallel data (147M words). The language model was estimated by the SRILM toolkit [Stolcke, 2002] with modified Kneser-Ney smoothing [Chen and Goodman, 1999]. Feature weights were tuned by minimum error rate traning (MERT) [Och, 2003] on the newswire part of the NIST-MT06 evaluation set (dev06-nw) – original version for the baseline, verb-reordered for the reordered system.

Figure 4.4 shows the results in terms of BLEU score [Papineni et al., 2001] for (i) the baseline system, (ii) the system only trained on reordered data and (iii) the system trained and tested on reordered data. The scores are plotted against the distortion limit (DL) used in decoding. To let the decoder explore the larger search space induced by the higher DL (8-10), we relaxed the pruning parameter for these conditions (maximum stack size: 1000 instead of the default 200).

We observe that, on the plain test set, the system trained on reordered data always performs better than the baseline (+0.5∼0.6 absolute), despite the mismatch between training and test ordering. This may be due to the fact that automatic word alignments are in general more accurate when less reordering is present in the data, although previous

[6] We use all the in-domain parallel data available for the NIST-MT09 task, that is everything except the large UN corpus. As reported by Green et al. [2010] the removal of UN data does not affect baseline performances on news texts.

[7] Given our experimental setting, it could be argued that our BLEU scores are biased because one of the references was also used to generate the verb reordering. However, in a series of experiments not reported here, we evaluated the same systems using only the remaining three references and observed similar trends as when all four references are used.

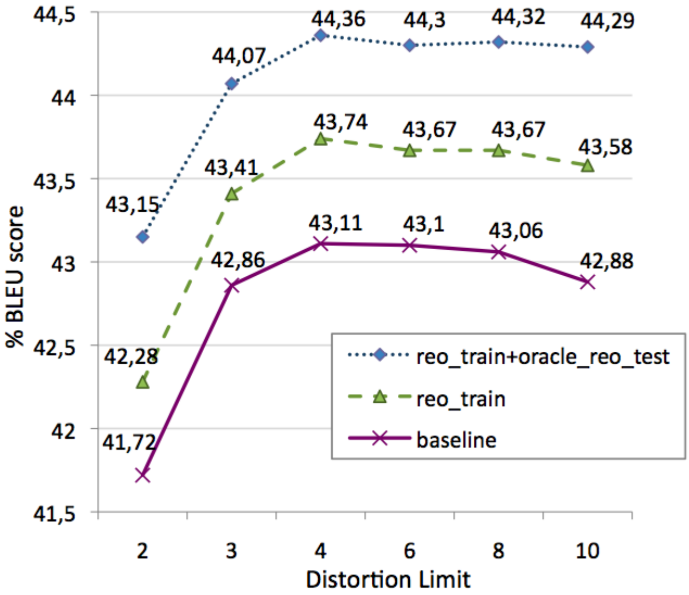


Figure 4.4: Impact of verb reordering on translation quality measured by BLEU [%]; the three curves refer to (i) the baseline system, (ii) the system only trained on reordered data and (iii) the system trained and tested on reordered data.

work [Lopez and Resnik, 2006] showed that gains in alignment accuracy seldom lead to better translations. Moreover, phrase extraction may benefit from a distortion reduction, since its heuristics are directly sensitive to word order. In fact, a higher number of phrases are extracted from the verb-reordered training corpus (15.9M compared to 14.6M from the non-reordered). Results on the oracle-reordered test set are also interesting: a gain of at least 1.2 BLEU over the baseline is reported in all tested DL conditions. These improvements are remarkable, keeping in mind that only 31% of the train and 33% of the test sentences get modified by verb reordering. Concerning distortion, although long verb movements are often observed in parallel corpora, relaxing the DL to high values does not benefit translation accuracy, even with our 'generous' setting of the pruning parameter. In fact, when more distortion is allowed, the risk of model errors increases as the reordering model has to rank an exponentially growing set of permutations.

4.5 Verb reordering lattices

Having assessed the potential improvement in verb reordering, we propose a technique to address this phenomenon at decoding time. The basic idea is to feed the decoder a word lattice that augments the source text with probable movements of its verb chunks.

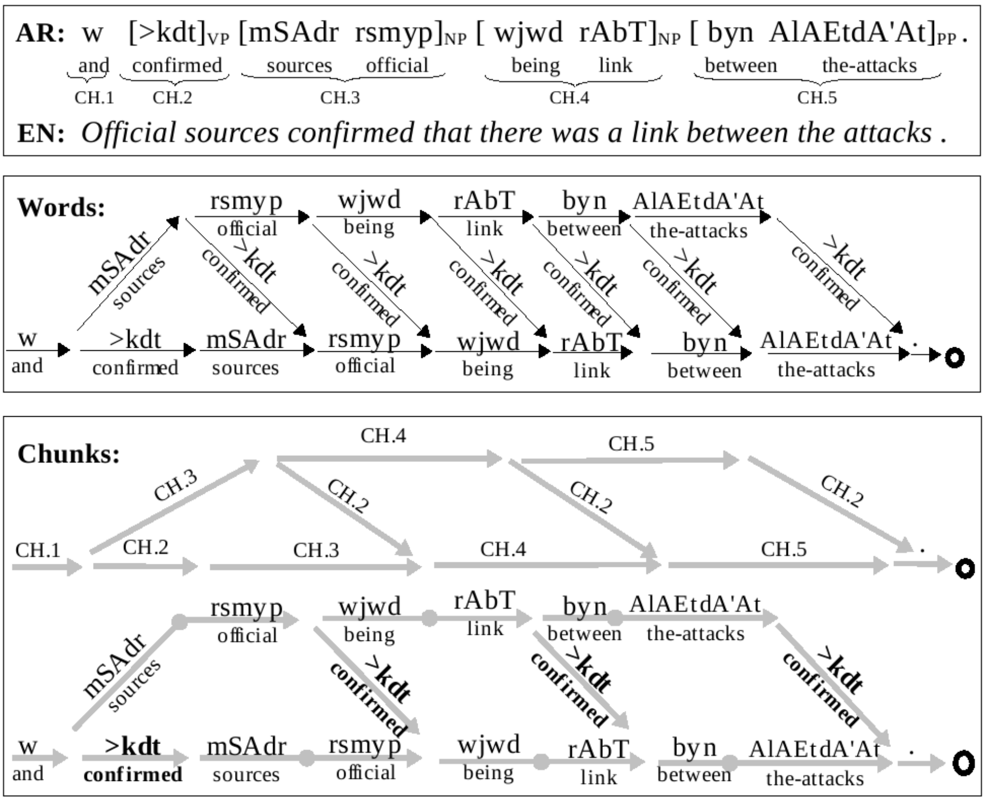

AR: w [>kdt]VP [mSAdr rsmyp]NP [wjwd rAbT]NP [byn AlAEtdA'At]PP .
and confirmed sources official being link between the-attacks
CH.1 CH.2 CH.3 CH.4 CH.5
EN: Official sources confirmed that there was a link between the attacks .
Words:
w and
>kdt confirmed
mSAdr sources
rsmyp official
wjwd being
rAbT link
byn between
AlAEtdA'At the-attacks
.
Chunks:
CH.1
CH.2
CH.3
CH.4
CH.5

Figure 4.5: Reordering lattices for Arabic VSO sentences. **Top**: a chunked sentence and its English meaning. **Center**: word-based lattice representing 7 possible word orders. **Bottom**: chunk-based lattice before and after word expansion. The final lattice thus obtained represents 4 input sentence permutations.

Word lattices were initially employed in SMT to compactly encode multiple transcription hypotheses produced by a speech recognizer (see Casacuberta et al. [2008] for a survey). More recently, they have been used to represent various forms of input ambiguity, both at the level of word order (starting from Zens et al. [2002]), and of token boundaries [Dyer et al., 2008]. A major issue with reordering lattices is that their size grows quickly with the amount and length of represented reorderings. We are particularly concerned with this issue because our decoder will perform additional reordering on the lattice input. Indeed, we can produce compact lattices by assuming the same conditions we put on the rules described in Section 4.2: (i) only reordering between chunks and (ii) no overlap between consecutive verb chunks movement.

Figure 4.5 illustrates the advantage of using chunk-based reordering lattices. Here, the main Arabic verb (>*kdt, ‘confirmed’*) appears in pre-subject position. If we considered for this sentence all possible movements of the verb to the right, we would obtain 6 reorderings as represented by the top lattice. With the chunk-based rules, instead, we treat chunks as units and obtain only 3 reordering paths represented by the second lattice.[8] Then, by expanding each chunk edge, we obtain the final word lattice ready for decoding.

Following the results of our corpus analysis (Section 4.3), we use a set of rules that move each verb alone or with its following chunk by 1 to 6 chunks to the right, which is consistent with the reordering applied to the training data. With these settings, we generate lattices with at most 5 × 6 additional chunk edges for each verb, as shown in Figure 4.6.

Before translation, each edge is assigned a weight used by the decoder as an additional feature. We tested two weighting schemes: the first (*W-switch*) favors the original word order, by arbitrarily assigning weights of 1 to the plain path edges, and 0.25 to the reordering path edges. The second (*W-prob*) models the length of verb movement: each edge is shared by one or more reordering paths; its probability is computed by summing the relative frequencies of the corresponding verb-chunk movements as observed in the parallel data (cf. Figure 4.2). For example, the left-most edge labeled *CH3* in Figure 4.6 corresponds to the probability of moving the verbal chunk alone by 1 to 6 chunks to the right, i.e. 0.132.

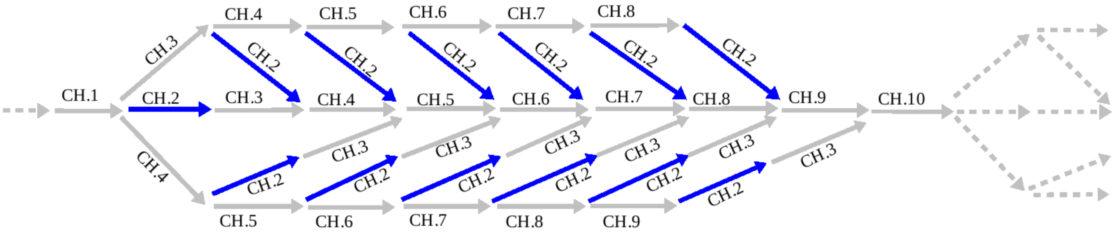


Figure 4.6: Structure of a chunk-based reordering lattice for verb reordering, before chunk-to-word expansion. The maximum verb chunk movement is set to 6. Bold edges represent the verb chunk.

Evaluation. For the experiments, we use the implementation of non-monotonic decoding for word lattices available in Moses [Dyer et al., 2008]. The translation system is the same as the one presented in Section 4.4, to which we added a feature function for the lattice score (*weight-i*). In order to minimize the influence of feature weight tuning on the outcome of our experiments, we do not run MERT a second time. Instead, we reuse the weights of the system trained and tuned on verb-reordered data. We manually optimize the lattice weight on the devset with a linear search over the interval [0.002,0.5].

[8] For simplicity, in this example, we do not consider the rule that moves the verb chunk along with the chunk following it (R2).

		eval08-nw		eval09-nw		reo08	
System	**DL**	**bleu**	**krs**	**bleu**	**krs**	**bleu**	**krs**
baseline	6	43.10	80.57	48.13	83.17	46.90	82.54
reord. training +							
plain input	6	43.67	80.62	48.53	83.58	46.64	82.23
lattice(W-switch)	4	44.04	80.93	48.96	83.75	**47.51**	**83.41**
lattice(W-prob)	4	**44.18**	**81.13**	**49.06**	**84.02**	47.40	83.22
oracle reordered	4	*44.36*	*81.29*	*49.26*	*84.30*	*48.25*	*84.03*

Table 4.1: BLEU and KRS [%] of baseline and reordered system on plain input, reordering lattices and oracle reordered test.

The resulting optimal value is 0.05. Empirically, the optimal distortion limit (DL) when translating lattices is 4, as opposed to 6 when translating text.

As evaluation metrics, we complement BLEU with the Kendall Reordering Score or KRS [Birch et al., 2010, Bisazza et al., 2012], which is a positive score based on the Kendall's Tau distance between the source-output permutation and the source-reference permutation.[9] The source-references and source-output word alignments needed to compute the reordering score[10] were obtained with the Berkeley Aligner [Liang et al., 2006] trained on our baseline system's training data (see Section 4.4).

Table 4.1 presents the results on two benchmarks: eval08-nw which was used to calibrate the reordering rules, and eval09-nw a yet unseen data set (newswire section of the NIST-MT09 evaluation set, 586 sentences). To focus the evaluation on sentences that contain verb reordering, we also consider a subset of eval08-nw including only sentences that are actually modified by oracle verb reordering. We call this subset reo08 (258 sentences).

We first consider the full test sets eval08-nw and eval09-nw. Here, the system trained on reordered data always outperforms the baseline, and verb reordering lattices yield further improvements according to both metrics. Concerning the lattice weighting schemes we can see that frequency-based weighting (*W-prob*) is slightly better than switch-based weighting (*W-switch*). Finally, the gap between the baseline and the best score obtainable by oracle verb reordering (44.36/81.29 on eval08-nw, 49.26/84.30 on eval09-nw) is largely filled.

However, figures are different on reo08: here, a degradation is observed when the reordered models are applied to non-reordered (plain) input. This suggests that the mis-

[9] See Section 2.5 for more details on the Kendall Reordering Score.

[10] MT outputs must be word-aligned to the source sentences in their original order, regardless of any transformation they undergo before decoding. That is why we apply a supervised aligner, rather than simply using the word alignments produced by the decoder.

match between training and test data has a negative impact on the reordering capabilities of the system with respect to verbs. We speculate that such negative effect is diluted in the full test set (eval08-nw) and compensated by the positive influence of verb reordering on phrase extraction. Indeed, when the lattice technique is applied we get an improvement of about 0.6/0.9 BLEU/KRS over the baseline, which is still a fair result, but not as good as the one obtained on the generic test sets. Another difference is that switch-based weighting (*W-switch*) slightly outperforms frequency-based weighting (*W-prob*) on 'the reordering-specific test. This is probably due to the fact that *W-switch* equally penalizes all chunk movements, letting the decoder catch all the needed long reorderings. On the other hand, the same scheme may cause spurious long reorderings, which would explain the lower score obtained on the generic tests. Finally, the oracle scores on reo08 show that there is still room for improvement on VSO sentences: from 47.51 to 48.25 BLEU, and from 83.41 to 84.03 KRS.

From the point of view of efficiency, lattice decoding is a very costly solution. Decoding times are three times longer than for standard text decoding, that is on average 177 milliseconds per word (ms/word) instead of 53 when translating eval09-nw.[11]

4.6 Discriminative lattice pruning

In order to refine our lattices and possibly overcome the lack of a reliable weighting scheme, we explore a lattice pruning technique aimed at discard unlikely reorderings. We model this step as a binary classification problem: that is, given a verb chunk movement in a sentence, predict whether it will minimize the reordering needed to produce a good English translation. Reorderings that do not meet this criterion should be excluded from the lattice.

As a supervised learning framework, we use Support Vector Machines [Boser et al., 1992, Vapnik, 1998] and syntactic tree kernels (STK) [Collins and Duffy, 2001] to fully exploit the availability of lexical and shallow-syntactic information. Tree kernels are a family of convolution kernels [Haussler, 1999] defined over pairs of trees. The trees are projected onto a very high dimensional space, where each subtree is mapped onto a distinct dimension. Here, pairwise similarity is measured in terms of the number of substructures shared by two trees. For our experiments, we adopt the implementation of the SVM-Light-TK toolkit,[12] which extends the SVM optimizer with support for tree kernel functions.

[11] The run times reported in this chapter were computed by an Intel Xeon E5420 processor.

[12] `http://disi.unitn.it/moschitti/Tree-Kernel.htm`

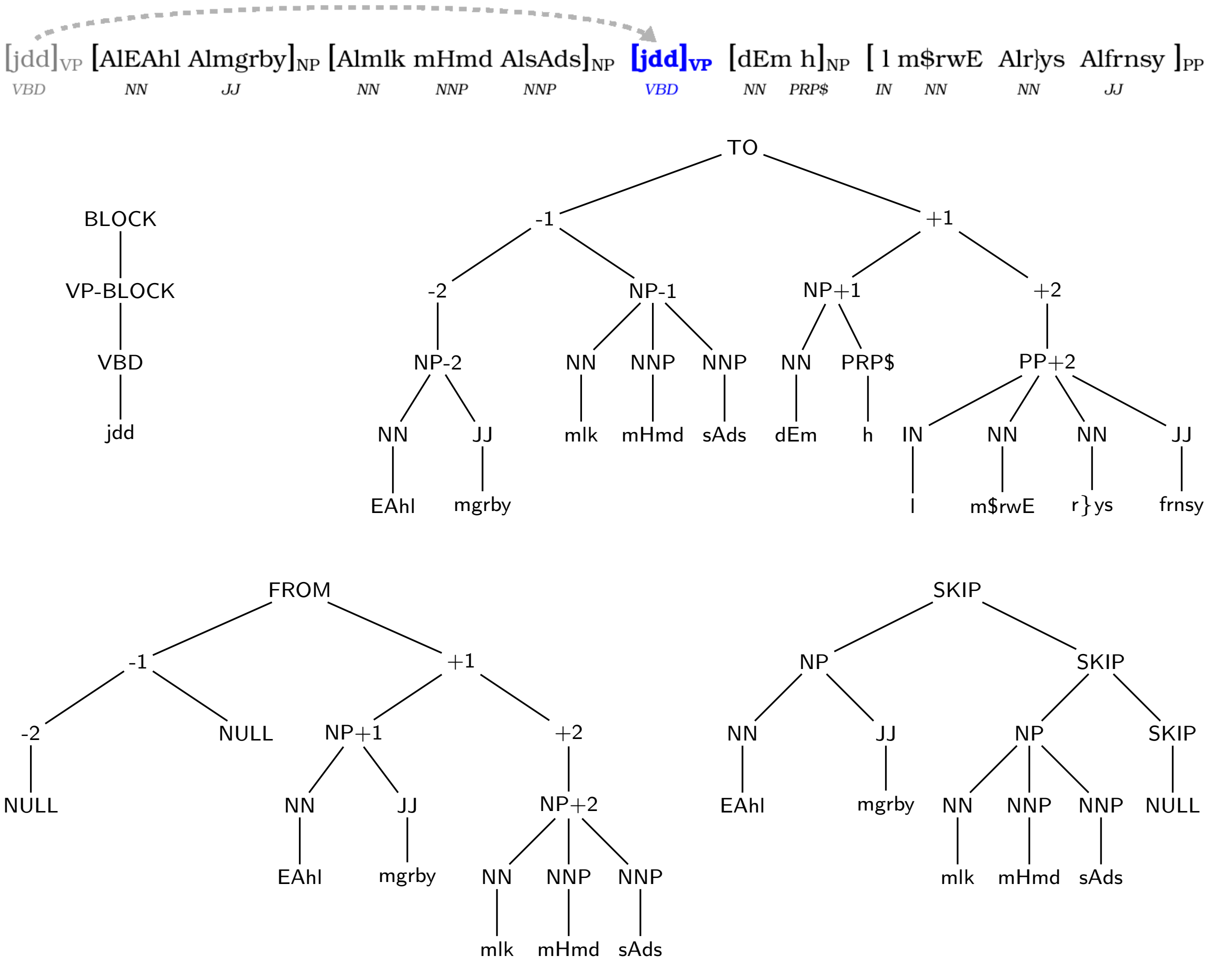


Figure 4.7: A forest of trees describing the movement of a verbal chunk. The four trees describe: the moving chunk(s) (top left); the context of the chunk in its original position (bottom left); the context of the chunk after the movement (top right); and the sequence of skipped chunks (bottom right).

We represent each reordering example as a forest of 4 trees that focus on different aspects of the movement (see example in Figure 4.7). These trees are artificial structures specifically designed to encode relevant information in a compact form. The similarity between two examples is computed as the sum of the tree kernels evaluated between the 4 pairs of corresponding trees. The trees specify respectively:

- the moving block, consisting of either one or two adjacent chunks (BLOCK);
- the context of the moving block in its original position (FROM);
- the context of the moving block in its final position (TO);
- the sequence of chunks lying between the original and final positions of the moving block (SKIP).

Within each tree, a chunk is modeled as a subtree of depth 3 whose root is the type of the chunk (verbal VP, nominal NP, prepositional PP etc.), and whose children are the words composing it. A word is encoded by its part-of-speech tag and its stemmed form. Fake root nodes and additional label decorators are used to glue the chunks together, and to provide other relevant information (e.g. relative positioning). The FROM and TO trees model the context of the verb chunk before and after the movement, respectively: the nodes labeled +1 and +2 describe the first and the second chunks to its right, while nodes -1 and -2 describe the chunks to its left. The use of such deep structures allows the model to capture various levels of increasingly fine-grained information.

The training set is constructed as follows: based on the findings of Section 4.3, we only consider movements of up to 6 positions involving the verbal chunk alone or together with the following chunk. This results in 12 reordering examples for each verb. Only the example that maximizes the optimization criteria defined in Section 4.2 is labeled as positive. If the verb is already in its optimal position, the whole set of examples will be negative. For testing, we generate all the possible reorderings and classify them with this model. For each example, the SVM classifier outputs its distance from the separating hyperplane, which we use as a confidence value to establish a ranking among the possible reorderings.[13] Note that the plain sentence ordering is always included since, according to our training data, it is the best choice in 84-86% of the cases (see Section 4.3).

Evaluation. The binary classifer was trained and tested on two subsets of the PSMT

[13] The problem addressed here is naturally a binary decision. In fact, a verb reordering can only be correct or incorrect, according to our formulation. Still, due to class imbalance issues – large majority of null reordering examples – it may help to model this as a ranking problem, where the correct reordering should simply score higher than all others. We have not however explored this option.

	#sent.	#verbs	#instances (positive)
train	20,000	46,047	320,191 (6,646)
test	722	1,776	12,193 (236)

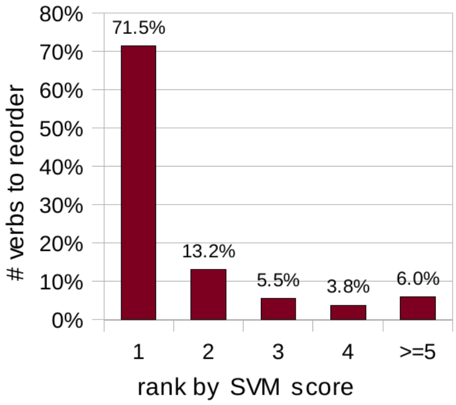


Figure 4.8: Left: Statistics of the SVM classifier's training and test sets. Right: Verb movement ranking results. Null movement instances are not included in the figures.

training data, described on the left of Figure 4.8. Its performance is 42.3% F-measure, 34.5% precision and 54.7% recall. To understand how informative the SVM margin values are, we also count how many times the optimal verb reordering received the highest score (1st) or the second highest score (2nd) etc. The results are presented on the right of Figure 4.8: in 71.5% of the cases in which a verb needs to be reordered, the classifier assigns the highest score to the correct movement. In 13.2% of the cases, it assigns the second highest score to the correct movement and so on. Interestingly, we can capture 84.7% of the actual verb reorderings by considering the 2-best SVM predictions, and 90% by considering the 3-best. Hence, we apply pruning to the translation test sets and repeat the SMT experiments previously described.

Table 4.2 shows BLEU and KRS scores obtained by translating SVM-pruned lattices. The row labeled *1-best-pruned* refers to the configuration in which the lattice only includes the original chunk order and the best SVM-ranked reordering. Similarly, the rows *2-best-pruned* and *3-best-pruned* correspond to lattices including the 2-best and the 3-best reorderings, respectively. The lattice weighting scheme used here is always *W-switch*, that is 1 for the plain path edges and 0.25 for the reordering edges. We can see that, according to both metrics, discriminative pruning yields slight but consistent improvements with respect to translating the full lattices as described in Section 4.5. Interestingly, the highest scores are obtained when only 1 or 2-best reordering paths among the 12 possible are retained[14]. In particular, when translating *2-best-pruned* lattices, the BLEU score increases from 44.04 to 44.29 on eval08-nw, and from 48.96 to 49.19 on eval09-nw. The

[14] The KRS achieved by *2-best-pruned* on eval08-nw is slightly higher than the oracle KRS. Although not significant, this result is plausible because the oracle itself is not perfect: it reorders verbs based on automatic alignments that may contain errors. In the lattice, instead, all verb reorderings are considered and then pruned.

		eval08-nw		eval09-nw		reo08	
System	**DL**	**bleu**	**krs**	**bleu**	**krs**	**bleu**	**krs**
baseline	6	43.10	80.57	48.13	83.17	46.90	82.54
reord. training +							
lattice(W-switch)	4	44.04	80.93	48.96	83.75	47.51	83.41
1-best-pruned	4	**44.34**	81.18	49.10	**84.15**	**48.04**	83.69
2-best-pruned	4	44.29	**81.30**	**49.19**	84.02	47.87	**83.88**
3-best-pruned	4	44.11	81.13	49.05	83.90	47.60	83.57
oracle reordered	4	*44.36*	*81.29*	*49.26*	*84.30*	*48.25*	*84.03*

Table 4.2: BLEU and KRS [%] of baseline and reordered system on SVM-pruned lattices and oracle reordered test.

KRS increases from 80.93 to 81.30 on eval08-nw, and from 83.75 to 84.02 on eval09-nw. The positive effect is now also evident on the reordering of the specific test set (reo08): KRS increases from 83.41 with full *lattice(W-switch)* to 83.88 with *2-best-pruned*.

Compared to the baseline, our best configuration – reordering of training data and *2-best-pruned* reordering lattice – leads to an overall gain of 1.06 BLEU and 0.85 KRS on the blind test set (eval09-nw), that is from 48.13 to 49.19 BLEU and from 83.17 to 84.02 KRS. We have reached the oracle upper-bound on eval08-nw, and closely approached it on eval09-nw.

Lattice pruning has also a positive effect on decoding time: for instance, translating *1-best-pruned* lattices takes on average 62 ms/word, versus 177 ms/word when the full lattices are used. However, the classification phase needed for pruning is very costly (162 ms/word), which makes the whole translation workflow considerably slower than the baseline (53 ms/word).

4.7 Conclusions

Based on the intuition that a few reordering patterns would suffice to handle the most significant cases of long-range reordering in Arabic-English, we focused on the problem of VSO sentences. Starting from simple linguistic assumptions about verb movement, we developed an effective technique to (i) partially reorder the training data and (ii) better handle verb reordering at decoding time. In particular, translation is performed on a word lattice representing a set of likely chunk-reordering paths, as ranked by a discriminative model that has access to a rich representation of the verb context.

The resulting system produces overall more readable translations (see examples in Figure 4.9) and outperforms a strong baseline both in terms of BLEU and of Kendall Reordering Score – a metric that directly addresses word ordering choices. Still, our approach has some limitations. First, efficiency is heavily affected when the full lattices are used. Lattice pruning considerably reduces decoding times, but implies a costly classification phase at pre-processing time. Second, the success of our method mainly depends on the particular distribution of reorderings found in the Arabic-English language pair. Applying it to another pair would require the development of a new rule set, which may be more complex than the one we have presented.

The next two chapters aim to overcome these limitations: Chapter 5 mainly addresses efficiency and presents a novel method to suggest likely input reorderings to the decoder, as an alternative to the lattice solution; Chapter 6 addresses versatility and presents a fully data-driven and decoding-integrated approach to dynamically shape the reordering search space. Concerning the languages, we will keep Arabic-English as our primary case study but extend the evaluation to the German-English pair.

SRC	و اشار السناتور الى دعم ـه مشروعا عرض على مجلس الشيوخ w **A$Ar** AlsnAtwr AlY dEm h m$rwEA ErD ElY mjls Al$ywx
REF	The Senator **referred** to his support for a project proposed to the Senate
BASE	The Senator to support projects presented to the Senate
NEW	Senator **noted** his support projects presented to the Senate
SRC	جدد العاهل المغربي الملك محمد السادس دعم ـه لـ مشروع الرئيس الفرنسي **jdd** AlEAhl Almgrby Almlk mHmd AlsAds dEm h l m$rwE Alr}ys Alfrnsy
REF	The Moroccan monarch King Mohamed VI **renewed** his support to the project of the French President
BASE	the Moroccan monarch King Mohammed VI his support to the French President
NEW	the Moroccan monarch King Mohammed VI **renewed** his support for the French President
SRC	و يمتد المشروع 500 كم متر و يربط المدينتين المقدستين بـ مدينة جدة w **ymtd** Alm$rwE 500 km mtr w yrbT Almdyntyn Almqdstyn b mdynp jdp
REF	The project **is** 500 kilometers **long** and connects the two holy cities with the city of Jeddah
BASE	**It extends** the project 500 km and linking the two holy cities in the city of Jeddah
NEW	The project **extends** 500 km, linking the two holy cities in the city of Jeddah

Figure 4.9: Examples showing SMT improvements obtained by chunk-based verb reordering. SRC is the reference translation, BASE is the baseline PSMT output, and NEW is the output of a PSMT system trained on verb-reordered data and tested on SVM-pruned (*2-best*) lattices.

Chapter 5

Modified Distortion Matrices

Modified distortion matrices are a novel method to suggest likely input reorderings to the decoder, which consists in reducing the distortion cost for specific pairs of input positions on a per-sentence basis.

5.1 Introduction

We have seen in Chapter 4 how reordering lattices can be used to suggest specific reordering patterns to a PSMT decoder. The proposed methods have a positive impact on translation quality, but at the expense of efficiency.

We present here a novel method to suggest reorderings to the decoder, which consists in reducing the distortion cost for specific pairs of input words.[1] Indeed, distortion can be thought of as a *matrix* assigning a cost to all pairs of input words. A set of multiple input reorderings can then be represented by modifying selected entries of this matrix so that the cost for the desired permutations is reduced. Compared to reordering lattices, modified distortion matrices provide a more compact and implicit way to encode likely reorderings in a sentence-specific fashion. Moreover, the matrix representation does not require multiplication of nodes for the same source word and is naturally compatible with the PSMT decoder's standard reordering mechanisms.

Added to the space of local permutations defined by a low distortion limit (DL), the modified distortion matrix results in a linguistically informed definition of the search space that simplifies the task of the in-decoder reordering model. As a difference from

[1] By "input word" or "source word" we denote the word at a given position of the input sentence, as opposed to the notion of word type.

the reordering lattice approach, here the input sentence is presented to the decoder in its original order, therefore the training data does not need to be reordered.

In this chapter we consider two language pairs where long reordering concentrates on few patterns: Arabic-English and German-English. We use fuzzy chunk-based reordering rules like those presented in Section 4.2 to generate probable long reorderings for each input sentence (see Section 5.2). Then, we use so-called *reordered n-gram language models* to rank and select the n-best permutations for translation (Section 5.3). Finally, we encode these reorderings by modifying selected entries of the distortion cost matrix (Section 5.4). Evaluated on well-known SMT benchmarks against a competitive baseline that includes state-of-the-art reordering models [Galley and Manning, 2008], the proposed technique leads to better translation quality with similar or even shorter decoding time (Section 5.5).

5.2 Fuzzy chunk-based reordering rules

The reordering characteristics of Arabic-English and German-English have been dicussed in Section 3.3. To generate probable long reorderings for these language pairs, we use fuzzy chunk-based rules. Shallow syntax chunking is indeed a lighter and simpler task compared to full parsing, and it can be used to constrain the number of reorderings in a softer way. While rules based on full parses are generally deterministic, chunk-based rules are non-deterministic or fuzzy, as they generate several permutations for each matching sequence. Besides defining a unique segmentation of the sentence, chunk annotation provides other useful information that can be used by the rules – namely chunk type and POS tags.[2]

For **Arabic-English** we apply the rules proposed in Section 4.2 to transform VS(O) sentences into SV(O). Namely, reorderings are generated by moving each verb chunk (VC), alone or with its following chunk, by 1 to 6 chunks to the right. The maximum movement of each VC is limited to the position of the next VC, so that neighboring verb-reordering sequences may not overlap. This rule set was shown to cover most (99.5%) of the verb reorderings observed in a parallel news corpus, including those where the verb must be moved along with an adverbial or a complement.

For **German-English** we propose a set of three rules aimed at arranging the German constituents in SVO order:

[2] We use AMIRA [Diab et al., 2004] to annotate the Arabic data, and Tree Tagger [Schmid, 1994] to annotate the German data.

- **infinitive**: move each infinitive VC right after a preceding punctuation. To bound the number of reorderings, at most three punctuations preceding the VC are considered. For example:

ORIG	Die EZB ist bestrebt, die Inflationrate unter zwei Prozent, oder zumindest knapp an der zwei-Prozent- Marke **zu halten**.
REO	Die EZB ist bestrebt, **zu halten** die Inflationrate unter zwei Prozent, oder zumindest knapp an der zwei-Prozent-Marke.
REF	The ECB wants to hold inflation to under two percent, or somewhere in that vicinity.

- **subordinate**: if a VC is immediately followed by a punctuation, place it after a preceding subordinating conjunction (KOUS) or substitutive relative pronoun (PRELS). One to three chunks are left between the conjunction (or pronoun) and the moved VC to account for the subject.

ORIG	Nachdem diese Infektion vorwiegend in Krankenhäusern und Altersheimen **vorkommt**, ...
REO	Nachdem diese Infektion **vorkommt** vorwiegend in Krankenhäusern und Altersheimen, ...
REF	Since this type of infection typically occurs in hospitals and nursing homes, ...

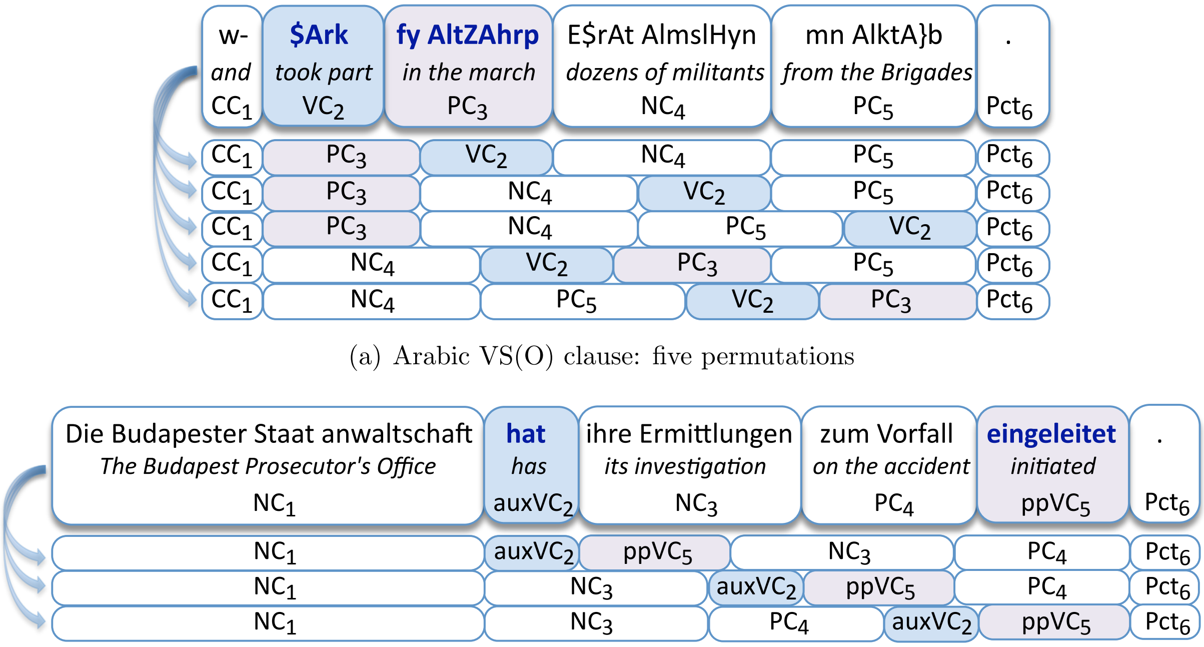
- **'broken' verb chunk**: join each finite VC (auxiliary or modal) with the nearest following non-finite VC (infinitive or participle). Place the resulting block in any position between the original position of the finite verb and that of the non-finite verb. If the distance between the finite and non-finite verb is more than 10 chunks, only the first 5 and last 5 positions of the verb-to-verb span are considered.

ORIG	Die Budapester Staatsanwaltschaft **hat** ihre Ermittlungen zum Vorfall **eingeleitet**.
REO	Die Budapester Staatsanwaltschaft **hat eingeleitet** ihre Ermittlungen zum Vorfall.
REF	The Budapest Prosecutor's Office has initiated an investigation on the accident.

Figure 5.1 illustrates the application of the fuzzy reordering rules.[3] In the Arabic sentence (a), the subject *'dozens of militants'* is preceded by the main verb *'took part'* and its argument *'to the march'*. The rules generate 5 permutations for one matching sequence (chunks 2 to 5), out of which the 5^{th} is the best for translation. The German sentence (b) contains a broken VC with the inflected auxiliary *'has'* separated from the past participle *'initiated'*. Here, the rules generate 3 permutations for the chunk sequence 2 to 5, corresponding to likely locations of the merged verb phrase, the 1^{st} being optimal.

By construction, both rule sets generate a limited number of permutations per matching sequence: in Arabic at most 12 for each VC; in German at most 3 for each infinitive

[3] The Arabic and German texts shown in the figure were pre-processed by a morphological segmenter and a compund splitter, respectively. See Section 5.5 for more details.



(a) Arabic VS(O) clause: five permutations

(b) German *broken* verb chunk: three permutations

Chunk types: CC conjunction, VC verb (auxiliary/past participle), PC preposition, NC noun, Pct punct.

Figure 5.1: Chunk permutations generated by fuzzy chunk-based reordering rules for translation into English.

VC and for each VC-punctuation sequence, at most 10 for each broken VC. Empirically, this yields on average 22 reorderings per sentence in the NIST-MT Arabic benchmark (dev06-nw) and 3 on the WMT German benchmark (test08).[4] Arabic rules are indeed more noisy, which is not surprising as all verb chunks can trigger some reordering.

5.3 Reordering selection

The number of chunk-based reorderings per sentence varies according to the rule set, the size of chunks, and the context. A high degree of fuzziness can complicate the decoding process, leaving too much work to the in-decoding reordering model. A solution to this problem is using an external model to score the rule-generated reorderings and discard the least likely. In such a way, a further part of reordering complexity is taken out of decoding.

At this end, instead of using a Support Vector Machine classifier as was done in Chapter 4, we apply *reordered n-gram models* that are lighter-weight and more suitable

[4] All benchmarks are described in detail in Section 5.5.

for a ranking task. A reordered n-gram model – or source-side decoding sequence model, as introduced by Feng et al. [2010a] – is a smoothed n-gram language model (LM) trained on a corpus of source sentences reordered to match the target word order. Similar models were used by Costa-jussà and Fonollosa [2006] as the main component of a *statistical machine reorderer* – that is, a decoder trained to reorder source sentences prior to monotonic translation. In contrast to these works, our models are trained on source sentences that are only partially reordered by applying the chunk-based rules described above. Consequently, at test time, the models are used to score the set of reorderings generated by the rules for each matching sequence. The task of our reordered LMs is thus considerably simpler than the one addressed in previous works. As another difference from Feng et al. [2010a], who integrate the LMs into the decoder, we apply our models only before translation, thus avoiding the multiplication of decoding states.

We introduce a novel type of reordered n-gram model trained at the level of chunks rather than words, with the aim of better capturing constituent-level reordering phenomena. Since the models are applied outside decoding, conflicts between phrase and chunk segmentation are not an issue.

Reordering selection is performed as follows:

1. chunk-based reordering rules are applied to the source side of the parallel training data, and word alignment is used to choose the optimal permutation for each rule-matching sequence ("oracle reordering");[5]
2. one or several chunk-level 5-gram LMs are trained on such reordered data, using different chunk representation modes;
3. reordering rules are applied to the test sentences and the resulting sets of rule-matching sequence permutations are scored by the LM (or by a log-linear combination of LMs);
4. the n-best permutations of each rule-matching sequence are selected for translation.

The question remains on how to represent chunks for LM training. A basic solution is to just use the chunk type label but this leads to poorly informative probability distributions. We then experiment with a combination of the chunk's type label and head word.[6] Head words can also be represented in several ways: surface form, POS or stem.

[5] As defined in Section 4.2, the optimal reordering for a source sentence is the one that minimizes distortion in the word alignment to a target translation, measured by number of swaps and sum of distortion costs.

[6] The head word of a chunk is detected by a simple heuristic: in Arabic it is the *first* (in German the *last*) word of the chunk whose POS corresponds to the chunk type.

Our representation modes are presented in Table 5.1 and evaluated in Section 5.5.

Chunk Representation Modes		Examples	
–	full	PC(fy|IN AltZAhrp|DET_NNFS) in the-march	NC(E$rAt|NNSFP AlmslHyn|DET_NNSMP) dozens (of) the-militants
T	chunk type	PC	NC
PT	chunk type, preposition if PC	PC:fy	NC
PHP	chunk type, preposition if PC, head word (POS)	PC:fy(DET_NNFS)	NC(DET_NNSMP)
HW	chunk type, head word (surface)	PC(AltZAhrp)	NC(E$rAt)
HS	chunk type, head word (stem)	PC(tZAhr)	NC(E$r)

Table 5.1: Chunk representation modes for reordered LMs. Explanation and examples based on two Arabic chunks: a prepositional and a nominal one.

5.4 Modified distortion matrices

We present here a novel technique to encode likely long reorderings of an input sentence, which can be seamlessly integrated into the PSMT framework.

During decoding, the distance between source positions is used for two main purposes: (i) generating a distortion penalty for the current hypothesis and (ii) determining the set of source positions that can be covered at the next hypothesis expansion. We can then tackle the coarseness of both distortion penalty and reordering constraints, by replacing the distance function with a function defined *ad hoc* for each input sentence.

Distortion can be thought of as a matrix assigning a positive integer to any ordered pair of source positions (s_x, s_y). In the linear distortion model this is defined as:

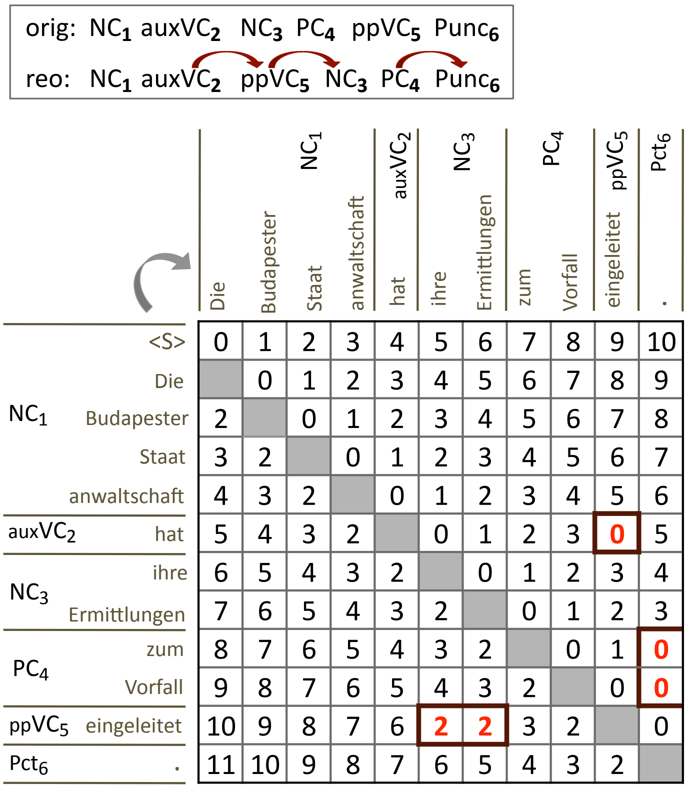
$$D_L(s_x, s_y) = |s_y - s_x - 1|$$

hence, moving to the right by 1 position costs 0, and by 2 positions costs 1. Moving to the left by 1 position costs 2, and by 2 positions costs 3, and so on. At the level of phrases, distortion is computed between the last word of the last translated phrase and the first word of the next phrase. We retain this equation as the core distortion function for our model. Then, we modify entries in the matrix so that the distortion cost is minimized for the decoding paths pre-computed by the fuzzy reordering rules.

Given a source sentence and its set of rule-generated permutations, the linear distortion matrix is modified as follows:

1. non-monotonic jumps (i.e. ordered pairs (s_i, s_{i+1}) such that $s_{i+1} - s_i \neq 1$) are extracted from the permutations;
2. then, for each extracted pair, the corresponding point in the matrix is assigned the lowest possible distortion cost, that is 0 if $s_i < s_{i+1}$ and 2 if $s_i > s_{i+1}$. We call these points *shortcuts*.

Although this technique is approximate and can overgenerate minimal-distortion decoding paths,[7] it practically works when the number of encoded permutations per sequence is limited. This makes modifed distortion matrices particularly suitable to encode just those reorderings that are typically missed by phrase-based decoders.



orig: NC_1 $auxVC_2$ NC_3 PC_4 $ppVC_5$ $Punc_6$

reo: NC_1 $auxVC_2$ $ppVC_5$ NC_3 PC_4 $Punc_6$

		NC_1				$auxVC_2$	NC_3		PC_4		$ppVC_5$	Pct_6
		Die	Budapester	Staat	anwaltschaft	hat	ihre	Ermittlungen	zum	Vorfall	eingeleitet	.
	<S>	0	1	2	3	4	5	6	7	8	9	10
NC_1	Die		0	1	2	3	4	5	6	7	8	9
	Budapester	2		0	1	2	3	4	5	6	7	8
	Staat	3	2		0	1	2	3	4	5	6	7
	anwaltschaft	4	3	2		0	1	2	3	4	5	6
$auxVC_2$	hat	5	4	3	2		0	1	2	3	**0**	5
NC_3	ihre	6	5	4	3	2		0	1	2	3	4
	Ermittlungen	7	6	5	4	3	2		0	1	2	3
PC_4	zum	8	7	6	5	4	3	2		0	1	**0**
	Vorfall	9	8	7	6	5	4	3	2		0	**0**
$ppVC_5$	eingeleitet	10	9	8	7	6	**2**	**2**	3	2		0
Pct_6	.	11	10	9	8	7	6	5	4	3	2	

Figure 5.2: Modified distortion matrix (mode A×A) of the German sentence given in Figure 5.1. The chunk reordering shown on top generates three *shortcuts* corresponding to the 0's and 2's highlighted in the matrix.

Since in this work we use chunk-based rules, we also have to convert chunk-to-chunk jumps into word-to-word shortcuts. We propose two ways to do this, given an ordered pair of chunks (c_x, c_y):

[7] In fact, any decoding path that includes a jump marked as shortcut benefits from the same distortion discount in that point.

mode L×F : create only one shortcut from the last word of c_x to the first of c_y;

mode A×A : create a shortcut from each word of c_x to each word of c_y.

The former solution implies that the first word of a reordered chunk is covered first and the last is covered last, whereas the latter admits more chunk-internal permutations with the same minimal distortion cost. To better understand this difference, consider for instance a noun chunk containing an adjective. Adjectives follow nouns in Arabic, but precede them in English. If the shortcut is created only to the first word of that chunk (mode L×F), the shortcut will only function if the chunk is translated as a single phrase. On the contrary, in mode A×A, the shortcut will function even if the decoder chooses to first translate the adjective as a separate phrase. In practice, the impact of this distinction on translation performance will mainly depend on the size of chunks.

Figure 5.2 shows the distortion matrix of the German sentence of Figure 5.1, with starting positions as columns and landing positions as rows. Suppose we want to encode the reordering shown on top of Figure 5.2, corresponding to the merging of the broken VC *'hat ... eingeleitet'*. This permutation contains three jumps: (2,5), (5,3) and (4,6). Converted to word-level in **A×A** mode, these yield five word shortcuts:[8] one for the onward jump (2,5) assigned 0 distortion; two for the backward jump (5,3), assigned 2; and two for the onward jump (4,6), also assigned 0. The desired reordering is now attainable within a DL of 2 words instead of 5. The same process is then applied to other permutations of the sentence.

Compared to dynamically varying the distortion limit, as done by Yahyaei and Monz [2010], modifying the distortion function makes it possible to expand the permutation search space by a much finer degree.

Distortion matrices have been integrated into the Moses toolkit [Koehn et al., 2007] using a sentence-level XML markup. The list of word shortcuts for each sentence is provided as an XML tag that is parsed by the decoder to modify the distortion matrix just before starting the search. As usual, the distortion matrix is queried by the distortion penalty generator and by the hypothesis expander, which is in charge of enforcing the distortion limit and gap constraint.[9] Note that the lexicalized reordering model is not affected by changes in the matrix, because it uses real word distances to compute the

[8] In **L×F** mode, instead, each chunk-to-chunk jump would yield exactly one word shortcut, for a total of three.

[9] The gap constraint checks that the left-most uncovered position is attainable from the end of the new source phrase, with the aim of avoiding decoding dead-ends (see Section 2.2.1). In practice, though, this condition can inhibit some of the reorderings encoded in the matrix. We have examined some heuristics to relax the gap constraint, but found that the standard one works better empirically.

orientation class of a new hypothesis.

5.5 Evaluation

In this section we evaluate the accuracy of reordering selection and the impact of modified distortion matrices on two news translation system.

For **Arabic-English**, we use all the in-domain parallel data provided for the NIST-MT09 evaluation for a total of 986K sentences (31M English words).[10] The target LM is trained on the English side of all available NIST-MT09 parallel data, UN included (147M words). For development and test, we use the newswire sections of the NIST benchmarks: dev06-nw, eval08-nw and eval09-nw consisting of 1033, 813 and 586 sentences respectively. All benchmarks include four reference translations, and the average sentence length is 33 words.

The **German-English** system is trained on WMT10 data: namely Europarl (v.5) plus News-commentary-2010 for a total of 1.6M parallel sentences, 43M English words. The target LM is trained on the monolingual news data provided for the constrained track (1133M words). For development and test, we use the WMT10 news benchmarks test08, test09 and test10: 2051, 2525 and 2489 sentences respectively, all with one reference translation.

To focus our SMT evaluation on problematic reordering, we also extract from each test set the sentences that got permuted by "oracle reordering" (see Section 5.3) and use these as "reordering-specific" test sets. These subsets constitute about a half of the Arabic sentences (reo08, reo09), and about a third of the German (reo09, reo10).

Concerning pre-processing, we apply standard tokenization to the English data, while for Arabic we use our in-house tokenizer that removes diacritics and normalizes special characters. Arabic text is then segmented with AMIRA [Diab et al., 2004] according to the ATB scheme.[11] The same tool also produces POS tagging and shallow syntax annotation. German tokenization and compound splitting are performed with Tree Tagger [Schmid, 1994] and the Gertwol morphological analyser [Koskenniemi and Haapalainen, 1994].[12] Tree Tagger is also used for POS tagging and shallow syntax chunking.

[10] The in-domain parallel data includes all the provided corpora except the UN proceedings, and the non-newswire parts of the GALE-Y1-Q4 consisting of 9K sentences of audio transcripts and web data. As reported by Green et al. [2010] the removal of UN data does not affect baseline performances on the news benchmarks.

[11] The Arabic Treebank tokenization scheme isolates conjunctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*.

[12] http://www2.lingsoft.fi/cgi-bin/gertwol

Reordered	**Ar-En** (dev06-nw)		**De-En** (test08)	
LM(s)	**Top-1**	**TopR-1|3**	**Top-1**	**TopR-1|3**
baseline	**81.7**	—	**60.0**	—
T	57.9	32.0 | 66.9	52.7	61.3 | 88.7
PT	54.8	35.6 | 65.4	50.5	60.3 | 87.8
PHP	54.1	40.3 | 71.9	48.5	61.2 | 88.4
HW	59.7	45.1 | 74.2	45.8	55.4 | 85.0
HS	59.7	47.4 | 72.7	46.3	54.6 | 83.5
PHP,HW	65.4	50.0 | 77.5	49.9	**63.0** | **89.3**
PHP,HW,HS	65.2	**50.4** | **77.2**	49.0	60.0 | 88.0
word 9-gram	63.1	49.8 | 72.5	47.2	55.9 | 86.2

Table 5.2: Permutation ranking accuracies of reordered n-gram LMs trained on different chunk representations (cf. Table 5.1): **Top-1** – accuracy at 1^{st} ranked reordering, including identity permutation; **TopR-1|3** – accuracy at 1^{st} and 3^{rd}, excluding identity permutation. Multiple LMs are log-linearly combined with uniform weights.

5.5.1 Reordering ranking accuracy

For intrinsic evaluation, we measure the ability of the reordered LMs to rank sets of chunk sequence permutations. Therefore, instead of perplexity, we compute the following accuracy measures:

Top-*n* indicates how often the true permutation[13] lies in the first n reorderings including "non-reordering" instances (identity permutation). This score denotes the LM's generic performance in ranking permutations;

TopR-*n* (reordering accuracy) is the same score, but computed only on sequences that are actually reordered in the true permutation (i. e. "non-reordering" instances are excluded). This score denotes the LM's ability to rank reorderings, but not to recognize sequences that shouldn't be reordered at all.

Because we always encode reorderings *in addition to* the original input order and let the decoder choose the optimal path, the latter measure is more important for the evaluation of our reordered LMs.

Table 5.2 presents results obtained with the parallel training and test data described above. In addition to various chunk-level 5-gram LMs and LM log-linear combinations, we include the results of a conservative baseline that always prefers no-reordering, and those of a *word*-level 9-gram LM that best approximates the work by Feng et al. [2010a].

[13] The true permutations of the test sets are obtained by oracle reordering, using the word alignment with the reference as supervision. In case of multiple references, only the first is used to this end.

Baseline accuracies show that 81.7% of the Arabic rule-matching sequences do not need reordering, versus 60% of the German, which confirms our initial observations on the rule sets' noisiness. Among the single chunk-level LMs, HS (chunk type + head stem) achieves the highest reordering accuracy for Arabic (TopR-1=47.4%), while T (only chunk type) is the best for German (TopR-1=61.3%). The word-level 9-gram LM outperforms all single chunk-level LMs in Arabic (but not in German). However, the best reordering accuracies overall are achieved by combining chunk-level LMs of different granularities: PHP, HW and HS for Arabic; PHP and HW for German. In the rest of the evaluation, we use these two combinations to select the 3-best reorderings of each rule-matching sequence, with a reordering accuracy (TopR-3) of 77.2% in Arabic and 89.3% in German.

5.5.2 SMT results and discussion

Using Moses we build competitive baselines on the training data described above. More specifically, for each language, word alignment is produced by the Berkeley Aligner [Liang et al., 2006]. The decoder is based on the log-linear combination of a phrase translation model, a lexicalized reordering model, a 6-gram target language model, distortion cost, word and phrase penalties. The language model is estimated by the IRSTLM toolkit [Federico et al., 2008] with modified Kneser-Ney smoothing [Chen and Goodman, 1999]. The reordering model is a hierarchical phrase orientation model [Tillmann, 2004, Koehn et al., 2005, Galley and Manning, 2008] trained on all the available parallel data. The hierarchical variant [Galley and Manning, 2008] was shown to outperform the default word-based on an Arabic-English task. As proposed by Johnson et al. [2007], statistically improbable phrase pairs are removed from the translation model.

Note that, in this work, the SMT models (phrase translation and orientation tables) are trained on non-reordered parallel data because the test sentences are presented to the decoder in their original order, differently from the lattice solution.

The DL is initially set to 5 words for Arabic-English and to 10 for German-English. For German-English only, we enable the Moses option *monotone-at-punctuation* which forbids reordering across strong punctuation marks. According to our experience, these are the optimal settings for the evaluated tasks. Feature weights are optimized by minimum error training [Och, 2003] on the development sets (dev06-nw and test08).

We measure translation quality with BLEU, METEOR and KRS.[14] To obtain the reference word alignments needed to compute the KRS, we apply the Berkeley Aligner (trained on the training data) to the test data. As for the source-output word alignments,

[14] Kendall Reordering Score, see Section 2.5.

(a) Arabic-English

Distortion Function	DL	eval08-nw			reo08	eval09-nw			reo09	ms/
		bleu	met	krs	krs	bleu	met	krs	krs	word
† plain *[baseline]*	5	44.5	34.9	81.6	82.9	49.9	38.0	84.1	84.4	263
plain	8	44.2$^{\triangledown}$	34.8	80.7$^{\blacktriangledown}$	82.2$^{\blacktriangledown}$	49.8	37.9	83.3$^{\blacktriangledown}$	83.5$^{\blacktriangledown}$	389
† modified: allReo, L×F	5+	44.4	34.9	82.2$^{\blacktriangle}$	83.7$^{\blacktriangle}$	49.9	37.8$^{\blacktriangledown}$	84.3	84.4	275
modified: 3bestReo, L×F	5+	44.5	35.1$^{\blacktriangle}$	82.3$^{\blacktriangle}$	83.5$^{\blacktriangle}$	50.7$^{\blacktriangle}$	38.1	84.8$^{\blacktriangle}$	85.0$^{\blacktriangle}$	267
† modified: 3bestReo, A×A	5+	44.8$^{\vartriangle}$	35.1$^{\blacktriangle}$	82.3$^{\blacktriangle}$	83.6$^{\blacktriangle}$	50.8$^{\blacktriangle}$	38.2$^{\blacktriangle}$	84.7$^{\blacktriangle}$	85.0$^{\blacktriangle}$	273

(b) German-English

Distortion Function	DL	test09			reo09	test10			reo10	ms/
		bleu	met	krs	krs	bleu	met	krs	krs	word
† plain *[baseline]*	10	18.8	27.5	65.8	66.7	20.1	29.4	68.7	68.9	292
plain	20	18.4$^{\blacktriangledown}$	27.4$^{\blacktriangledown}$	63.6$^{\blacktriangledown}$	65.2$^{\blacktriangledown}$	19.8$^{\blacktriangledown}$	29.3$^{\blacktriangledown}$	66.3$^{\blacktriangledown}$	66.6$^{\blacktriangledown}$	369
plain	4	18.4$^{\blacktriangledown}$	27.4$^{\blacktriangledown}$	67.3$^{\blacktriangle}$	66.9	19.6$^{\blacktriangledown}$	29.1$^{\blacktriangledown}$	70.2$^{\blacktriangle}$	69.6$^{\blacktriangle}$	158
† modified: allReo, L×F	4+	19.1$^{\blacktriangle}$	27.6$^{\blacktriangle}$	67.6$^{\blacktriangle}$	68.1$^{\blacktriangle}$	20.4$^{\blacktriangle}$	29.4	70.6$^{\blacktriangle}$	70.7$^{\blacktriangle}$	161
modified: 3bestReo, L×F	4+	19.2$^{\blacktriangle}$	27.7$^{\blacktriangle}$	67.4$^{\blacktriangle}$	68.1$^{\blacktriangle}$	20.4$^{\blacktriangle}$	29.4	70.4$^{\blacktriangle}$	70.6$^{\blacktriangle}$	160
† modified: 3bestReo, A×A	4+	19.2$^{\blacktriangle}$	27.7$^{\blacktriangle}$	67.4$^{\blacktriangle}$	68.4$^{\blacktriangle}$	20.6$^{\blacktriangle}$	29.5$^{\vartriangle}$	70.4$^{\blacktriangle}$	70.7$^{\blacktriangle}$	163

Table 5.3: Impact of modified distortion matrices on translation quality, measured with BLEU, METEOR and KRS (all in percentage form, higher scores mean higher quality). The settings used for weight tuning are marked with †. Statistically significant differences wrt the baseline are marked with $^{\blacktriangle\blacktriangledown}$at the $p \leq .05$ level and $^{\vartriangle\triangledown}$at the $p \leq .10$ level. Decoding time is measured in milliseconds per input word.

we use those produced by the decoder. Statistically significant differences are assessed by approximate randomization as in Riezler and Maxwell [2005].[15]

Table 5.3 reports results obtained by varying the DL and modifying the distortion function. To evaluate the reordering selection technique, we also compare the encoding of all rule-generated reorderings against only the 3 best per rule-matching sequence, as ranked by our best performing reordered LMs (cf. Section 5.5.1). We mark the DL with a '+' to denote that some longer jumps are allowed by modified distortion. Run times refer to the translation of the first 100 sentences of eval08-nw and test09 by an Intel Xeon E5420 processor (including models loading time).

Arabic-English. As anticipated, raising the DL does not improve, but rather worsens performances. The decrease in BLEU and METEOR reported with DL=8 is not significant, but the decrease in KRS is both significant and large. Efficiency is heavily af-

[15] Translation scores and significance tests are computed with the tools multeval [Clark et al., 2011] and sigf [Padó, 2006].

fected, with a 48% increase of the run time from 263 ms/word with DL=5 to 389 ms/word with DL=8.

Results in the row "allReo" are obtained by encoding all the rule-generated reorderings in **L**×**F** chunk-to-word conversion mode. Except for some gains in KRS reported on eval08-nw, most of the scores are lower or equal to the baseline. Such inconsistent behavior is probably due to the low precision of the Arabic rule set, pointed out in Section 5.2.

Finally, we arrive to the performance of 3-best reorderings per sequence. With **L**×**F** we obtain several improvements, but it is with **A**×**A** that we are able to beat the baseline according to all metrics. BLEU and METEOR improvements are rather small but significant and consistent across test sets, the best gain being reported on eval09-nw (+.9 BLEU). Most importantly, substantial word order improvements (+.7/+.6 KRS) are achieved on both full test sets and selected subsets (vs*). According to these figures, word order is affected only in the sentences that contain problematic reordering. This is good evidence, suggesting that the decoder does not get "confused" by spurious shortcuts.

Looking at run times, we can say that modified distortion matrices are a very efficient way to address long reordering. Even when all the generated reorderings are encoded, translation time increases only by 5%. Reordering selection naturally helps to further reduce decoding overload. As for conversion modes, **A**×**A** yields slightly higher run times than **L**×**F** because it generates more shortcuts for the same number of reorderings.

German-English. In this task we manage to improve translation quality with a setting that is almost twice as fast as the baseline: from 292 ms/word to 163 ms/word, that is a 44% decrease of the run time. In fact, as shown by the first part of the table, the best baseline results are obtained with a rather high DL of 10 (only KRS improves with a lower DL). However, with modified distortion, the best results according to all metrics are obtained with a DL of 4.

Looking at the rest of the table, we see that reordering selection is not as crucial as in Arabic-English. This is in line with the properties of the more precise German reordering rule set (two rules out of three generate at most 3 reorderings per sequence). Considering all scores, the last setting (3-best reordering and **A**×**A**) appears as the best one, achieving the following gains over the baseline: +.4/+.5 BLEU, +.2/+.1 METEOR, +1.6/+1.7 KRS, and +1.7/+1.8 KRS on the test subsets (vs*). The agreement observed among such diverse metrics makes us confident about the goodness of the approach.

5.6 Conclusions

We have addressed the problem of word reordering in two language pairs – Arabic-English and German-English – where most long-range phenomena are describable by a handful of linguistic rules. By means of non-deterministic chunk reordering rules, we have generated likely permutations of the test sentences and ranked them with n-gram LMs trained on pre-ordered source language data. We have then introduced the notion of modified distortion matrices to naturally encode a set of likely reorderings in the decoder input. Compared to varying the distortion limit, modifying the distortion function allows for a finer and linguistically informed definition of the search space, which is reflected in better translation outputs and more efficient decoding.

The main limitation of this work lies in the need of language-specific reordering rules. As a solution, in the next chapter we propose a fully data-driven approach to dynamically shape the reordering search space.

Chapter 6

Dynamic Reordering Space Pruning

During decoding, the reordering model can be used not only as a feature function, but also as an early indication of whether or not a given reordering path should be further explored. We exploit this idea to refine the reordering search space in a dynamic and fully data-driven way.

6.1 Introduction

The reordering techniques proposed so far assume that most long-range reorderings in the working language pair concentrate on few patterns. Moreover, they require the availability of language-specific *rules* describing such patterns. In the present chapter, instead, we explore a fully-data driven approach to dynamically shape the reordering search space. While arising from the same motivations as the rest of this thesis, this approach is language-independent and can in principle improve PSMT reordering in any type of language pair.

Reordering in PSMT can be viewed as the problem of choosing the input permutation that leads to the highest-scoring output sentence. Due to efficiency reasons, however, the input permutation space cannot be fully explored, and is therefore limited with hard reordering constraints. Such constraints are also important for translation quality because the existing models are typically not discriminative enough to guide the search over very large sets of reordering hypotheses (i. e. relaxing the reordering constraints generally results in more model errors).

The existing reordering constraints, however, are rather simple and typically based on word-to-word distances. We propose instead to dynamically define the reordering search space, based on the scores of a specific reordering model. To this end, we build a binary classifier that predicts whether a candidate input position should be translated right after another, given the words at those positions and their contexts. When this model is integrated into decoding, its predictions can be used not only as an additional feature function, but also as an early indication of whether or not a given reordering path should be further explored. More specifically, at each hypothesis expansion, we compute the set of input positions that are reachable according to permissive reordering constraints, and prune it based only on the reordering model score. Then, the hypothesis is expanded normally by covering the non-pruned positions. This technique makes it possible to dynamically refine the search space while decoding with a very high distortion limit, which can improve translation quality *and* efficiency at the same time.

This chapter is closely related to the work of Yahyaei and Monz [2010] on *dynamic distortion limit*, which consists in training a classifier to predict the most probable jump length after each input word,[1] and using the predicted value as the DL after that position. In our work we develop this idea further, and use a classifier to predict which specific input words, rather than input intervals, should be translated next. This makes it possible to shape the reordering space in a finer way, as compared to simply varying the distortion limit. Our method also differs from the dynamic distortion limit in that it does not generate inconsistent constraints, that is leading to decoding dead-ends.

The remainder of this chapter is organized as follows. We start by describing in detail our reordering model and its features. In the following section, we introduce early pruning of reordering steps as a way to dynamically shape the input permutation space. Finally, we present an empirical analysis of our approach including intrinsic evaluation of the model and SMT experiments on two popular news translation tasks, from Arabic to English and from German to English.

6.2 The WaW reordering model

As discussed in Section 2.2.2, many solutions have already been proposed to explicitly model word reordering during decoding. *Phrase orientation models* [Tillmann, 2004, Koehn et al., 2005] predict the orientation of a phrase with respect to the last translated one, therefore they are not suitable to predict the kind of long-range reorderings that we address in our work. Another option would be to use *jump models* [Al-Onaizan and Papineni, 2006, Green et al., 2010], as was done by Yahyaei and Monz [2010], but this approach has another drawback: classifying reordering steps by their jump length has the

[1] As in Chapter 5, we denote here by "input word" the word at a given position of the input sentence, as opposed to the notion of word type.

effect of overly penalizing long jumps because of their low frequency compared to short jumps. This bias is undesirable, as we are especially interested in detecting probable long reorderings. Finally, *source decoding sequence models* [Feng et al., 2010a, Visweswariah et al., 2011] predict which word of the input sentence is likely to be translated at a given state of decoding. The model we present here belongs to this group, and more precisely to the sub-group of *source word pair reordering models*, which we find especially suitable to predict long reorderings. According to this approach, reordering is modeled as the problem of judging whether a given input word should be translated right after another (**W**ord-**a**fter-**W**ord). This formulation is particularly useful for the decoder to decide whether a reordering path is promising enough to be further explored. Moreover, when translating a sentence, choosing the next source word to translate appears as a more natural problem than guessing how much to the left or to the right we should move from the current source position.

The WaW reordering model addresses a binary decision task through the following maximum-entropy classifier:

$$P(R_{i,j}{=}Y|f_1^J,i,j) = \frac{exp[\sum_m \lambda_m h_m(f_1^J,i,j,R_{i,j}{=}Y)]}{\sum_{Y'} exp[\sum_m \lambda_m h_m(f_1^J,i,j,R_{i,j}{=}Y')]}$$

where f_1^J is a source sentence of J words, h_m are feature functions and λ_m the corresponding feature weights. The outcome Y can be either 1 or 0, with $R_{i,j}{=}1$ meaning that the word at position j is translated *right after* the word at position i. Features are extracted from the local context of positions i and j, and from the words occurring between them (see details below).

Our WaW reordering model is strongly related to that of Visweswariah et al. [2011] – hereby called Travelling Salesman Problem (TSP) model – with few important differences: (i) we do not include in the features any explicit indication of the jump length, in order to avoid the bias on short jumps; (ii) they train a linear model with MIRA [Crammer and Singer, 2003] by minimizing the number of input words that get placed after the wrong position, while we use a maximum-entropy classifier trained by maximum-likelihood; (iii) they use an off-the shelf TSP solver to find the best source sentence permutation and apply it as pre-processing to training and test data. By contrast, we integrate the maximum-entropy classifier directly into the SMT decoder and let all other models (phrase orientation, translation, target LM etc.) contribute to the final reordering decision.

6.2.1 Features

Like the TSP model [Visweswariah et al., 2011], the WaW model builds on binary features similar to those typically employed for dependency parsing [McDonald et al., 2005]: namely, combinations of surface forms or POS tags of the words i and j and their context. Our feature templates are presented in Table 6.1. The main novelties with respect to the TSP model are the mixed word-POS templates (rows 16-17) and the shallow syntax features. In particular, we use the chunk types of i, j and their context (18-19), as well as the chunk head words of i and j (20). Finally we add a feature to indicate whether the words i and j belong to the same chunk (21). The jump orientation – forward/backward – is included in the features that represent the words comprised between i and j (rows 6, 7, 14, 15). However, no explicit indication of the jump length is included in any feature.

	$\boldsymbol{i-2}$	$\boldsymbol{i-1}$	$\boldsymbol{i}$	$\boldsymbol{i+1}$	$\boldsymbol{b}$	$\boldsymbol{j-1}$	$\boldsymbol{j}$	$\boldsymbol{j+1}$
1			w				w	
2		w	w				w	
3	w	w	w				w	
4		w	w				w	w
5			w	w		w	w	
6					w	w	w	
7					w_{all}	w	w	
8			p				p	
9		p	p				p	
10	p	p	p				p	
11		p	p				p	p
12			p	p		p	p	
13		p	p	p		p	p	p
14					p	p	p	
15					p_{all}	p	p	
16			w				p	
17			p				w	
18			c				c	
19		c	c	c		c	c	c
20			h				h	
21	belong_to_same_chunk(i, j)?							

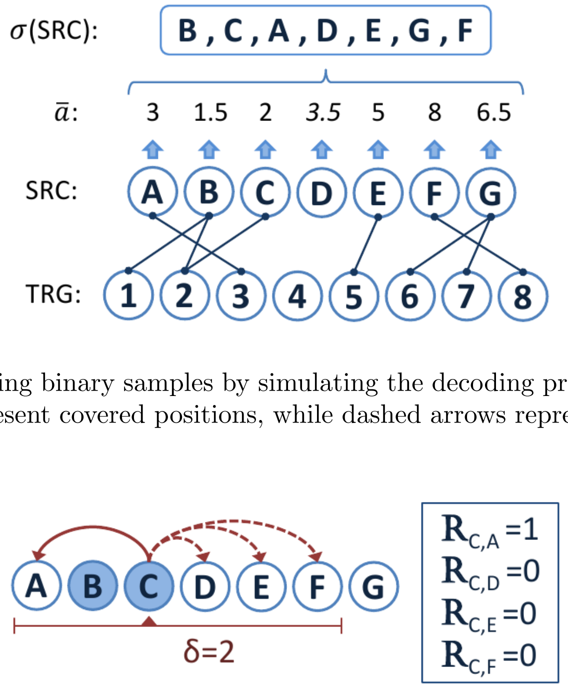
$\boldsymbol{w}$: word identity, $\boldsymbol{p}$: POS tag, $\boldsymbol{c}$: chunk type, $\boldsymbol{h}$: chunk head word

Table 6.1: Feature templates used to learn whether a source position j is to be translated right after i. Positions comprised between i and j are denoted by b and generate two feature templates: one for each position (6 and 14) and one for the concatentation of them all (7 and 15).

6.2.2 Training data

To generate training data for the classifier, we first extract reference reorderings from a word-aligned parallel corpus. Given a parallel sentence, different heuristics may be used to convert arbitrary word alignments to a source permutation [Birch et al., 2010, Feng et al., 2010a, Visweswariah et al., 2011]. Similarly to this last work, we compute for each source word f_i the mean $\overline{a_i}$ of the target positions aligned to f_i, then sort the source words according to this value.[2] As a difference, though, we do not discard unaligned words but assign them the mean of their neighbouring words' alignment means, so that a complete permutation of the source sentence (σ) is obtained. Table 6.2(a) illustrates this procedure.

(a) Converting word alignments to a permutation: source words are sorted by their target alignments mean $\bar{a}$. The unaligned word "D" is assigned the mean of its neighbouring words' $\bar{a}$ values $(2 + 5)/2 = 3.5$:



(b) Generating binary samples by simulating the decoding process: shaded rounds represent covered positions, while dashed arrows represent negative samples:

Table 6.2: The classifier's training data generation process.

[2] Using the mean of the aligned indices makes the generation of training data more robust to alignment errors. This heuristic does not handle well the case of source words that are correctly aligned to non-consecutive target words. However, this phenomenon is also not captured by standard PSMT models, who only learn to translate continuous phrases.

Given the reference permutation, we then generate positive and negative training samples by simulating the decoding process. We traverse the source positions in the order defined by σ, keeping track of the positions that have already been covered and, for each $t : 1 \leq t \leq J$, generate:

- one positive sample ($R_{\sigma_t,\sigma_{t+1}}$=1) for the source position that comes right after it,
- a negative sample ($R_{\sigma_t,u}$=0) for each source position in $\{u : \sigma_t{-}\delta{+}1 < u < \sigma_t{+}\delta{+}1 \;\wedge\; u \neq \sigma_{t+1}\}$ that has not yet been translated.

Here, the **sampling window** δ serves to control the size of the training data and the proportion between positive and negative samples. Its value naturally correlates with the DL used in decoding. The generation of training samples is illustrated by Table 6.2(b).

6.2.3 Integration into phrase-based decoding

Rather than using the new reordering model to pre-process the input as done by Visweswariah et al. [2011], we directly integrate it into the PSMT decoder Moses [Koehn et al., 2007].

Two main computation phases are required by the WaW model: (i) at system initialization time, all features weights are loaded into memory, and (ii) before translating each source sentence, features are extracted from it[3] and model probabilities are pre-computed for each position pair (i, j) such that $|j - i - 1| \leq$ DL. Note that this solution is possible and efficient because our model does not employ features depending on the decoding history, like the word that was translated before the last one, or like the previous jump legth. This is an important difference with respect to the reordered source LM proposed by Feng et al. [2010a], which requires inclusion of the last n translated words in the decoder state.

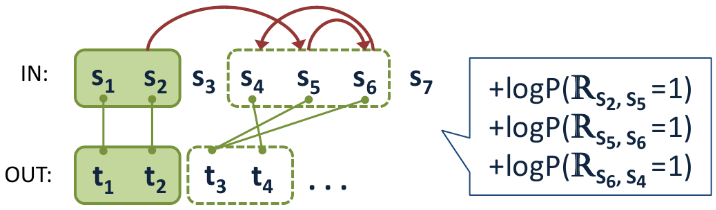


Figure 6.1: Integrating the binary word reordering model into a phrase-based decoder: when a new phrase is covered (dashed boxes), the model returns the log-probability of translating its words in the order defined by the phrase-internal word alignment.

Figure 6.1 illustrates the scoring process: when a partial translation hypothesis $\mathcal{H}$ is expanded by covering a new source phrase $\tilde{f}$, the model returns the log-probability of

[3] POS tags and chunk annotation are encoded as input factors.

translating the words of $\tilde{f}$ in that particular order, just after the last translated word of $\mathcal{H}$. In details, this is done by converting the phrase-internal word alignment[4] to a source permutation, in just the same way it was done to produce the model's training examples. Thus, the global score is independent from phrase segmentation, and normalized across outputs of different lengths: that is, the probability of any complete hypothesis decomposes into J factors, where J is the length of the input sentence.

The WaW reordering model is fully compatible with, and complementary to the lexicalized reordering (phrase orientation) models included in Moses.

6.3 Model-based reordering space definition

We now explain how the WaW reordering model can be used to dynamically refine the input permutation space. This method is not dependent on the particular classifier described above, but can in principle work with any device estimating the probability of translating a given input word after another.

6.3.1 Early pruning of reordering steps

A way to refine the reordering search space is to query the reordering model at the time of hypothesis expansion, and to filter out hypotheses solely based on their reordering score. The rationale is to avoid costly hypothesis expansions for those source positions that the reordering model considers very unlikely to be covered at a given point of decoding. In practice, this works as follows:

- at each hypothesis expansion, we first enumerate the set of uncovered input positions that are reachable within a fixed DL, and query the WaW reordering model for each of them;
- solely based on the WaW reordering score, we apply histogram and threshold pruning to this set, and proceed to expand only the non-pruned positions.

Furthermore, it is possible to ensure that local reorderings are always allowed, by setting a so-called **non-prunable-zone** of width ϑ around the last translated position.

According to how the DL, pruning parameters, and ϑ are set, we can actually aim at different targets: with a low DL, loose pruning parameters, and ϑ=0 we can try to speed up search without sacrificing much translation quality. With a high DL, strict pruning

[4] Phrase-internal word alignment is provided in the phrase table.

parameters, and a medium ϑ we ensure that the standard medium-range reordering space is explored, as well as those few long jumps that are promising according to the reordering model. In our experiments, we explore this second option with the setting DL=18 and ϑ=5.

The underlying idea is similar to that of *early pruning* proposed by Moore and Quirk [2007], which consisted in discarding possible extensions of a partial hypothesis based on their estimated score *before* computing the exact language model score. Our technique too has the effect of introducing additional points at which the search space is pruned. However, while theirs was mainly an optimization technique meant to avoid useless LM queries, we instead aim at refining the search space by exploiting the fact that some SMT models are more important than others at different stages of the translation process. Our approach actually involves a continuous alternation of two processes: during hypothesis expansion the reordering score is combined with all other scores, while during early pruning some reordering decisions are taken only based on the reordering score. In this way, we try to combine the benefits of fully integrated reordering models with those of monolingual pre-ordering methods.

6.3.2 Technical details

We have explained above how early reordering pruning works at the conceptual level. In practice, though, there are some technical issues due to the fact that the WaW reordering model operates at the level of input positions (words), while hypothesis expansion proceeds at the level of input ranges (phrases).

Let us consider again Figure 6.1 and assume that ($\mathrm{s}_1\mathrm{s}_2|\mathrm{t}_1\mathrm{t}_2$) is the last translated phrase pair of a partial hypothesis $\mathcal{H}$. At this point, $\mathcal{H}$ can be expanded by covering any range of input positions $[x..y]$ that satifies the standard conditions:

- there exists at least one translation option matching $[x..y]$ on the source side;
- x is reachable within the DL from the last word of the last covered phrase (here s_2);
- the left-most uncovered position of the input (here s_3) is reachable within the DL from y ("gap constraint").

Now, we want to prune this set of input ranges, but the WaW reordering scores apply to single input positions. We then proceed with the following steps:

1. the set of reachable input *positions* is pruned to obtain the set of positions allowed for expansion (A);

2. the input *ranges* that do not include any position in A are discarded;
3. all the remaining input *ranges* are considered for expansion, but only with translation options whose first target word aligns to a source *position* in A.

Hence, in Figure 6.1, the input range [4..6] would be discarded in step (2) if all positions $\{4, 5, 6\}$ were pruned in step (1). Otherwise, the translation options matching the input range [4..6] would be examined along with their internal word alignment. At this point (step 3) the translation option showed in the figure ($\mathrm{s}_4\mathrm{s}_5\mathrm{s}_6|\mathrm{t}_3\mathrm{t}_4$) would be discarded if s_5 was pruned in step (1), because s_5 aligns to the option's first target word t_3.

At the end of the pruning process, we check that at least one range can be expanded. If not, we let nevertheless expand the ranges starting at the left-most uncovered position. This measure effectively prevents decoding dead-ends.

6.4 Evaluation

We test our approach on two news translation tasks where sentences are typically long and complex: the Arabic-English NIST-MT09 task and the German-English WMT10 task.

In Arabic-English, long reordering errors mostly concern verbs, as all of SVO, VSO and, more rarely, VOS constructions are attested in modern written Arabic. This issue is well known in the SMT field and was addressed by several recent works, with deep or shallow parsing-based techniques [Green et al., 2009, Carpuat et al., 2012, Andreas et al., 2011, Bisazza et al., 2012]. In German-English too, verbs are among the hardest words to reorder, due to the verb-second nature of German and to the particular order of subordinate clauses.[5] Reordering in this language pair was addressed, among others, by Collins et al. [2005] with manually written syntax-based pre-processing rules. We question whether our approach – which is not conceived to solve these specific problems, nor requires manual rules to predict verb reordering – will succeed in improving long reordering in a fully data-driven way.

The SMT training, development and test corpora used in this chapter, as well as the pre-processing pipelines, are the same as those used in the evaluation of Chapter 5 (see in particular Section 5.5).

[5] See Section 3.2 for a detailed discussion of Arabic and German word order.

(a) Arabic-English results on tides-mt04.

Features	[templates]	P	R	F
W	[1-7]	73.11	16.39	26.78
P	[8-15]	69.46	54.82	61.28
W,P	[1-17]	70.16	56.49	62.58
W,P,C	[1-21]	70.59	58.14	**63.77**

(b) German-English results on test08.

Features	[templates]	P	R	F
W	[1-7]	66.14	11.37	19.40
P	[8-15]	66.87	48.29	56.08
W,P	[1-17]	67.17	48.87	56.58
W,P,C	[1-21]	66.97	49.96	**57.23**

Table 6.3: WaW reordering model performance (precision, recall and F-score) achieved by different feature subsets. The template numbers refer to the rows of Table 6.1.

6.4.1 Reordering model intrinsic evaluation

Before proceeding to the SMT experiments, we evaluate the performance of the WaW reordering model in isolation.

All the tested configurations are trained with the freely available MegaM Toolkit,[6] implementing the conjugate gradient method [Hestenes and Stiefel, 1952], in maximum 100 iterations. Training samples are generated within a sampling window of width δ=10, from a subset (30K sentences) of the parallel data described above, resulting in 8M training word pairs for each language pair.[7] Arabic-English test samples are generated from tides-mt04 (1324 sentences, 370K samples generated with δ=10), one of the corpora included in our SMT training data. German-English test samples are generated from the development set test08 (2051 sentences, 444K samples). Features with less than 20 occurrences are ignored.

Classification accuracy

Table 6.3 presents precision, recall, and F-score achieved by different feature subsets, where W stands for word-based, P for POS-based and C for chunk-based feature templates. We can see that all feature types contribute to improve the classifier's performance. In Arabic-English, the word-based model achieves the highest precision but a very low

[6] http://www.cs.utah.edu/˜hal/megam/ [Daumé III, 2004].

[7] This is the maximum number of samples manageable by MegaM. However, even scaling from 4M to 8M was only slightly helpful in our experiments.

recall, while the POS-based has much more balanced scores. A better performance overall is obtained by combining word-, POS- and mixed word-POS-based features (62.58% F-score). Finally, the addition of chunk-based features yields a further improvement of about 1 point, reaching 63.77% F-score. In German-English we observe similar trends, except for the lower precision of the word-based model. Here too, the best F-score is achieved by including all feature types (W,P,C) in the classifier. Given these results, we decide to use the W,P,C model for the rest of the intrinsic evaluation and for all the SMT experiments. However, we underline the fact that performances would be only slightly worse if the shallow syntax annotation was not available.

In general, classification accuracy scores are rather low, which shows that the reordering problem is very hard to solve even when rich context-based features are used.

Ranking accuracy

A more important aspect to evaluate for our application is how well our model's probability can *rank* a typical set of reordering options. In fact, the WaW model is not meant to be used as a stand-alone classifier, but as one of several SMT feature functions. Moreover, for early reordering pruning to be effective, it is especially important that the correct reordering option be ranked in the top n among those available at the time of a given hypothesis expansion.

In order to measure this, we simulate the decoding process by traversing the source words in target order and, for each of them, we examine the ranking of all words that may be translated next (that is the uncovered positions within a given DL). We check how often the correct jump was ranked first (Top-1) or at most third (Top-3). We also compute the latter score on long reorderings only (Top-3-long): i.e. backward jumps with distortion D>7 and forward jumps with D>6. In Table 6.4 results are compared with the ranking produced by standard distortion, which always favors shorter jumps. Two conditions are considered: DL=10 corresponding to the sampling window δ used to produce the training samples, and DL=18 corresponding to the maximum distortion of jumps that will be considered in our early-pruning SMT experiment.

In Arabic-English, the WaW reordering model outperforms standard distortion by a large margin (about 10% absolute) in terms of overall accuracies. This is an important result, considering that the jump length, strongly correlating with the jump likelihood, is not directly known to our model. As regards the DL, the higher limit naturally results in a lower DL-error rate (percentage of correct jumps beyond DL): namely 0.76% instead of 2.44%. However, jump prediction becomes much harder: Top-3 accuracy of long jumps by

(a) Arabic-English results on tides-mt04.

Model	**DL**	**DL-err**	**Top-1**	**Top-3**	**Top-3-long**	
					back	**forw.**
Distortion	10	2.44	61.75	79.63	50.66	65.96
	18	0.76	61.98	80.00	18.85	52.28
WaW	10	2.44	71.22	91.16	76.35	69.30
	18	0.76	71.24	91.76	67.95	51.77

(b) German-English results on test08.

Model	**DL**	**DL-err**	**Top-1**	**Top-3**	**Top-3-long**	
					back	**forw.**
Distortion	10	6.54	61.65	73.79	44.48	54.03
	18	1.96	62.02	74.53	17.32	34.66
WaW	10	6.54	61.49	80.34	72.70	67.34
	18	1.96	61.70	81.41	68.02	46.25

Table 6.4: Word-to-word jump ranking accuracies (%) of standard distortion and WaW reordering model, in different DL conditions. DL-err is the percentage of correct jumps lying beyond the DL. The test sets consist of 40K and 51K reordering decisions: one for each source word in tides-mt04 and test08, respectively.

distortion drops from 50.66% to 18.85% (backward) and from 65.95% to 52.28% (forward). Our model is remarkably robust to this effect on backward jumps, where it achieves 67.95% accuracy. Given the syntactic characteristics of Arabic and English, the typical long reordering pattern in this language pair consists in (i) skipping a clause-initial Arabic verb, (ii) covering a long subject, then finally (iii) jumping back to translate the verb and (iv) jumping forward to continue translating the rest of the sentence (see Figure 6.8 for an example). Deciding when to jump back to cover the verb (iii) is the hardest part of this process, and that is precisely where our model seems more helpful, while distortion always prefers to proceed monotonically achieving a very low accuracy of 18.85%. In the case of long forward jumps (iv), instead, distortion is advantaged as the correct choice typically corresponds to translating the first uncovered position, that is the *shortest* jump available from the last translated word. Even here, our model achieves a reasonable accuracy of 51.77%, only slightly lower than that of distortion (52.28%).

In German-English, figures are somewhat different: the WaW reordering model is slightly weaker than distortion in terms of Top-1 accuracy (61.70% versus 62.02% with DL=18), but much stronger than it in terms of Top-3 (81.41% versus 74.53% with DL=18). As regards long reorderings, the performance of distortion degrades greatly when the DL is raised, while the WaW model is robust to this effect. More precisely, with DL=18, the WaW model achieves a Top-3 accuracy of 68.02% versus 17.32% by distortion on long backward jumps. Interestingly, the WaW model outperforms distortion also on long forward jumps (46.25% versus 34.66%), which was not the case in Arabic-English. This can be explained by the fact that long reordering patterns are more mixed in German-English than in Arabic-English.

In summary, the WaW reordering model significantly outperforms distortion in the ranking of long jumps in both language pairs. In the large majority of cases, it is able to rank a correct long jump in the top 3 reordering options, which suggests that it can be effectively used for early reordering pruning.

6.4.2 SMT experiments

Our SMT systems are built with the Moses toolkit, while word alignment is produced by the Berkeley Aligner [Liang et al., 2006]. For each language pair, the baseline decoder includes a phrase translation model, a lexicalized reordering model, a 6-gram target language model, distortion cost, word and phrase penalties. More specifically, the baseline reordering model is a **hierarchical phrase orientation model** [Tillmann, 2004, Koehn et al., 2005, Galley and Manning, 2008] trained on all the available parallel data. The hierarchical variant [Galley and Manning, 2008] was shown to outperform the default word-based on an Arabic-English task. To make our baseline even more competitive, we apply **early distortion cost**, as proposed by Moore and Quirk [2007]. As explained in Section 2.2.2, this distortion function has the same value as the standard one over a complete translation hypothesis, but it anticipates the gradual accumulation of the cost, making hypotheses of the same length more comparable to one another. Note that this option has no effect on the distortion *limit*, but only on the distortion *feature function*. The language model is estimated by the IRSTLM toolkit [Federico et al., 2008] with modified Kneser-Ney smoothing [Chen and Goodman, 1999]. As proposed by Johnson et al. [2007], statistically improbable phrase pairs are removed from the translation model.

Feature weights are optimized by minimum BLEU-error training [Och, 2003] on dev06-nw and test08). To reduce the effects of the optimizer instability, we tune each configuration four times and use the average of the resulting weight vectors to translate the test sets, as suggested by Cettolo et al. [2011]. Moreover, eval08-nw and test09 are used to select the early pruning parameters for the last experiment, while eval09-nw and test10 are used as blind tests.

We recall that reo09 and reo10 are the reordering-specific test sets obtained by extracting from eval09-nw and test10, respectively, only those sentences that got reordered by our chunk-based rules (see Section 5.2). Thus, the Arabic subset (reo09: 299 sentences) contains verb-subject (VS) sentences, representing about a half of all sentences. The German subset (reo10: 885 sentences) contains sentences where the verb lies in a different position with respect to the canonical SVO order of English. More precisely, the rules recognize three patterns: clause-final infinitives, verbs placed at the end of a subordinate clause, and 'broken' verb chunks where the finite verb is separate from the non-finite verb. According to our rule set, these patterns occur in about a third of the German sentences.

We evaluate global translation quality by **BLEU** and **METEOR**, and reordering accuracy by Kendall Reordering Score (**KRS**).[8] As in Chapter 5, source-output word alignments are produced by the decoder, while source-reference word alignments are generated by the Berkeley Aligner trained on the training data. Statistical significance is assessed by approximate randomization as in Riezler and Maxwell [2005].

Fine-grained evaluation

Our work specifically addresses long-range reordering phenomena in language pairs where these are quite rare, although crucial for preserving the source text meaning. Hence, an improvement at this level may not be detected by the general-purpose metrics used so far.

A better way to automatically evaluate our systems would be to use syntax- or semantics-based metrics, as the impact of long reordering errors is particularly important at these levels. As a light-weight alternative, we propose instead to use word classes (i. e. Part-of-Speech) and to concentrate the evaluation on those classes that are typically crucial to guess the general structure of a sentence.

We then develop a KRS variant that is only sensitive to the positioning of specific input words. As explained in Section 2.5, the standard KRS is computed as follows:

$$K(\pi,\sigma) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{d}(i,j)}{\frac{1}{2}n(n-1)}$$

$$\mathbf{d}(i,j) = \begin{cases} 1 & \text{if } \pi_i < \pi_j \text{ and } \sigma_i > \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

To obtain a word-weighted KRS, we assume that each input word f_i is assigned a weight λ_i, and modify the formula above as follows:

$$\mathbf{d}_\lambda(i,j) = \begin{cases} \lambda_i + \lambda_j & \text{if } \pi_i < \pi_j \text{ and } \sigma_i > \sigma_j \\ 0 & \text{otherwise} \end{cases}$$

[8] See Section 2.5 for details on these metrics.

A similar element-weighted version of Kendall Tau was proposed by Kumar and Vassilvitskii [2010] to evaluate document rankings in information retrieval. Because long reordering errors in Arabic-English and German-English mostly affect verbs, we set the weights to 1 for verbs and 0 for all other words to only capture verb reordering errors, and refer to the resulting metric as **KRS-V**.

Results and discussion

To motivate the choice of our baseline setup (early distortion cost and DL=8), we first compare the performance of *standard* and *early* distortion costs under various DL conditions. This analysis was performed only on the Arabic-English pair.

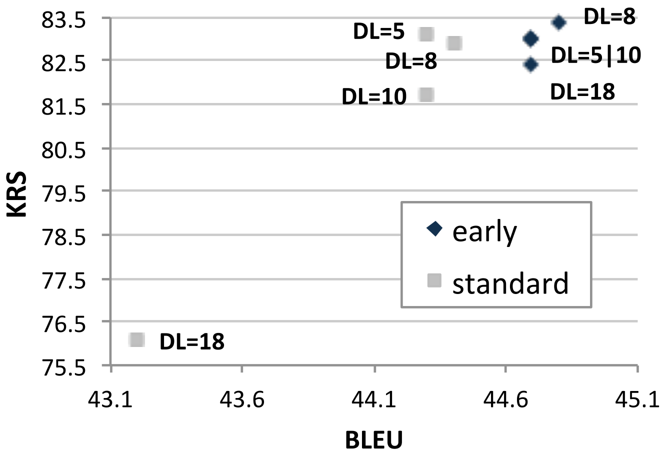


Figure 6.2: Standard *versus* early distortion cost results on the Arabic-English eval08-nw, under different distortion limits (DL). Best scores are on top-right corner.

As shown in Figure 6.2, most results are close to each other in terms of BLEU and KRS, but early distortion consistently outperforms the standard one (differences are statistically significant). The most striking difference appears at a very high distortion limit (18), where standard distortion scores drop by more than 1 BLEU point and almost 7 KRS points! Early distortion is much more robust (only -1 KRS when passing from DL=8 to DL=18), which makes our baselines especially strong from the reordering point of view.

Table 6.5 presents the results obtained by integrating the WaW reordering model as an additional feature function, and by applying our technique of early reordering pruning. Note that statistical significance is always computed against the baseline [B]. Run times refer to the translation (model loading times included) of the first 100 sentences of eval08-nw and test09 by an Intel Xeon X5650 processor.

(a) Arabic-English

		eval08-nw				eval09-nw				reo09	ms/
DL	**Reo. models**	**bleu**	**met**	**krs**	**krs-v**	**bleu**	**met**	**krs**	**krs-v**	**krs-v**	**word**
8	hier.lexreo, early disto.[**B**]	44.8	35.2	83.4	85.6	50.6	38.1	84.7	86.2	84.8	98
	+ WaW model	45.0	35.2	83.7$^{\vartriangle}$	85.9	51.1$^{\blacktriangle}$	38.3$^{\blacktriangle}$	85.1$^{\blacktriangle}$	86.8$^{\blacktriangle}$	85.8$^{\blacktriangle}$	102
18	hier.lexreo, early disto.	44.7	34.9$^{\blacktriangledown}$	82.4$^{\blacktriangledown}$	84.9$^{\blacktriangledown}$	50.3	38.0$^{\triangledown}$	83.9$^{\blacktriangledown}$	85.8$^{\triangledown}$	84.3$^{\triangledown}$	164
	+ WaW model	44.8	35.2	82.7$^{\blacktriangledown}$	85.5	51.0$^{\vartriangle}$	38.3$^{\blacktriangle}$	84.2$^{\triangledown}$	86.2	85.2	172
	+ early reo.pruning(ϑ=5)	45.0	35.3	83.7$^{\vartriangle}$	86.3$^{\blacktriangle}$	50.9	38.3$^{\blacktriangle}$	84.9	87.0$^{\blacktriangle}$	86.2$^{\blacktriangle}$	79

(b) German-English

		test09				test10				reo10	ms/
DL	**Reo. models**	**bleu**	**met**	**krs**	**krs-v**	**bleu**	**met**	**krs**	**krs-v**	**krs-v**	**word**
8	hier.lexreo, early disto.[**B**]	19.0	27.4	66.1	64.2	20.4	29.2	69.2	67.1	63.9	347
	+ WaW model	19.3$^{\blacktriangle}$	27.5	66.1	64.3	20.7$^{\blacktriangle}$	29.4$^{\blacktriangle}$	69.4$^{\blacktriangle}$	67.3	64.3	361
18	hier.lexreo, early disto.	18.0$^{\blacktriangledown}$	27.3$^{\triangledown}$	61.7$^{\blacktriangledown}$	61.0$^{\blacktriangledown}$	19.3$^{\blacktriangledown}$	29.1$^{\blacktriangledown}$	64.4$^{\blacktriangledown}$	63.8$^{\blacktriangledown}$	61.2$^{\blacktriangledown}$	680
	+ WaW model	18.1$^{\blacktriangledown}$	27.3$^{\triangledown}$	60.6$^{\blacktriangledown}$	60.3$^{\blacktriangledown}$	19.6$^{\blacktriangledown}$	29.2	63.7$^{\blacktriangledown}$	63.4$^{\blacktriangledown}$	60.8$^{\blacktriangledown}$	703
	+ early reo.pruning(ϑ=5)	19.4$^{\blacktriangle}$	27.7$^{\blacktriangle}$	66.5$^{\blacktriangle}$	64.9$^{\blacktriangle}$	20.6$^{\blacktriangle}$	29.5$^{\blacktriangle}$	69.5$^{\blacktriangle}$	67.8$^{\blacktriangle}$	65.9$^{\blacktriangle}$	240

Table 6.5: Effects of WaW reordering modeling and early reordering pruning on translation quality, measured with % BLEU, METEOR, and KRS: regular (KRS) and verb-specific (KRS-V). Statistically significant differences with respect to the baseline [**B**] are marked with $^{\blacktriangle\blacktriangledown}$ at the $p \leq .05$ level and $^{\vartriangle\triangledown}$ at the $p \leq .10$ level. Decoding time is measured in milliseconds per input word.

In both language pairs, integrating the WaW model as an additional feature function results in small but consistent improvements (second row of each table), showing that this type of model conveys at least some information that is missing from the state-of-the-art reordering models. Some of these gains, though, are not statistically significant. Concerning the run time, we notice just a small overload of about 4%: that is, from 98 to 102 ms/word in Arabic-English and from 347 to 361 ms/word in German-English.

As expected, raising the DL to 18 with no special pruning (third row) has a negative impact on both translation quality and efficiency. This effect is especially visible on the reordering scores: from 84.7 KRS to 83.9 KRS on the Arabic-English eval09-nw, and from 69.2 to 64.4 KRS on the German-English test10. Run times increase by 67% in Arabic-English (from 98 to 164 ms/word), and by 96% in German-English (from 347 to 680 ms/word).

Adding the WaW model under the high DL condition (third row) is beneficial according to all scores in Arabic-English, however the low-distortion baseline [**B**] remains unbeaten. In German-English, the addition of the WaW model has inconsistent effects under the high DL condition: the BLEU scores increase slightly (e. g. from 19.3 to 19.6 BLEU on test10), but the reordering scores decrease (e. g. from 64.4 to 63.7 KRS on test10).

We then proceed to the last experiment where the reordering space is dynamically pruned based on the WaW model scores (fifth row of each table). As explained in Section 6.3.1, a non-prunable-zone of width ϑ=5 is set around the last covered position. To set the early pruning parameters, we perform a grid search over the values (1, 2, 3, 4, 5) for histogram and (0.5, 0.25, 0.1) for relative threshold pruning, and select the values that achieve the best BLEU and KRS on eval08-nw and test09. The optimal values in Arabic-English are 3 (histogram) and 0.1 (threshold). This pruning setting implies that, at a given point of decoding where i is the last covered position, a new word can be translated only if:

- it lies within a DL of 5 from i, or
- it lies within a DL of 18 from i and its WaW reordering score is among the top 3 and at least equal to 1/10 of the best score (in linear space).

The optimal values in German-English are instead 2 and 0.25. The resulting configurations are re-optimized by MERT on dev06-nw and test08 before the final experiment.

As shown in the last row of Tables 6.5(a) and 6.5(b), early pruning achieves the best results overall: despite the high DL, we report no loss in BLEU, METEOR and KRS, but we actually see several improvements. On the Arabic-English blind test (eval09-nw), the improvements are +0.3 BLEU, +0.2 METEOR and +0.2 KRS (only METEOR is significant). On the German-English blind test (test10), the improvements are: +0.2 BLEU, +0.3 METEOR, +0.3 KRS (all are significant). While these gains are indeed small, we recall that our techniques affect rather rare and isolated events which can hardly emerge from the general purpose evaluation metrics. Moreover, to our knowledge, this is the first time that a PSMT system is able to maintain a good performance on these language pairs while admitting very long-range reordering.

Finally, and more importantly, the verb-specific KRS-V improves significantly on both the generic benchmarks and the reordering-specific subsets. In Arabic-English, we achieve a notable gain of +0.8 KRS-V on eval09-nw and +1.4 KRS-V on reo09. In German-English, we report a gain of +0.7 KRS-V on test10 and +2.0 KRS-V on reo10. All these results validate our hypothesis on the importance of refining the reordering space.

Efficiency is also largely improved by our early reordering pruning technique. Translation time is reduced from 98 to 79 ms/word in Arabic-English and from 347 to 240 ms/word in German-English, which corresponds to a speed-up of 19% and 31% over the baseline, respectively.

Interaction with beam-search pruning

During the beam-search decoding process, early reordering pruning interacts with regular hypothesis pruning based on the weighted sum of all model scores. In particular, all the systems presented in this thesis apply a default histogram threshold of 200 to each stack of hypotheses that cover the same number of input words (cf. Section 2.2.1). Given this setting, one could argue that the positive effect of our approach is mainly due to the reduction of search error. In other words, reordering quality may be improved by simply relaxing the standard pruning parameters, in which case our approach should be considered as an optimization technique rather than an actual model improvement.

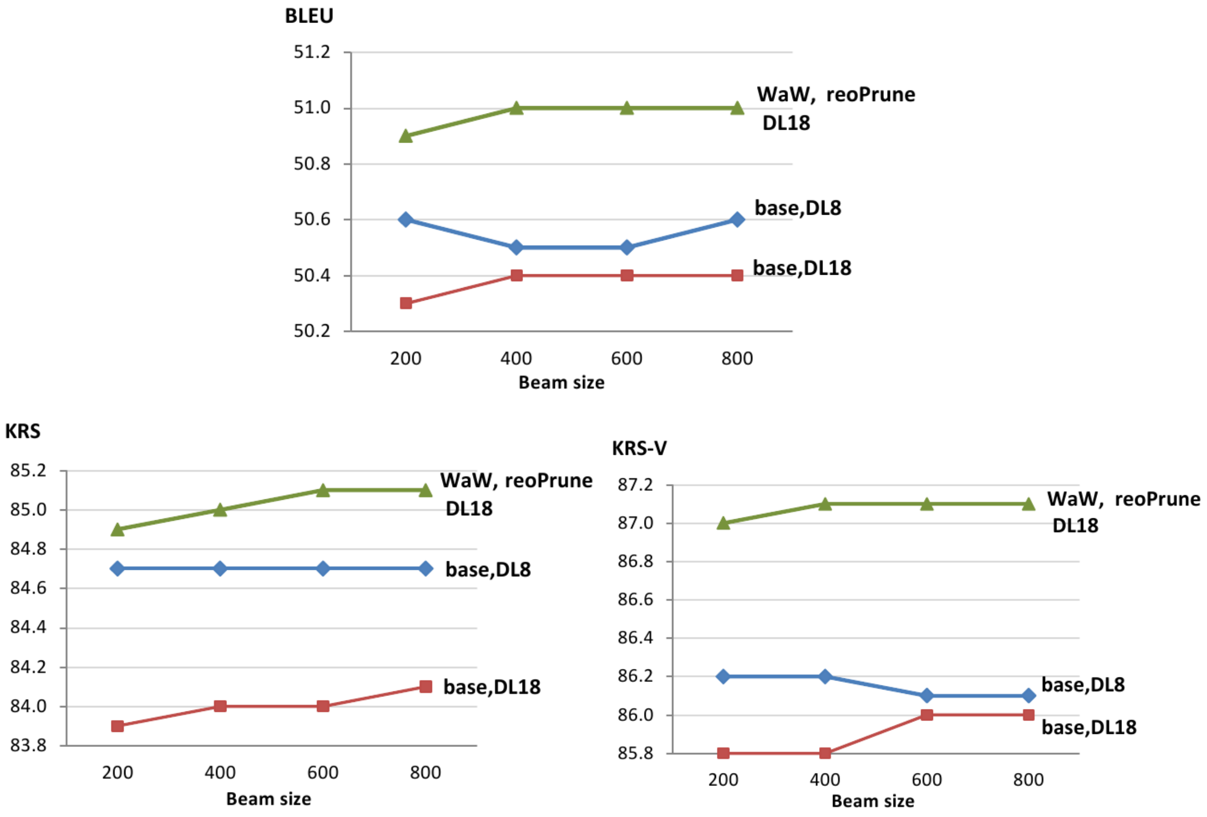


Table 6.6: Effect of beam size on translation quality measured by BLEU, KRS and KRS-V, in two baseline systems (DL=8 and DL=18) and in the WaW early-pruning system (Arabic-English task).

To answer this question, we perform another series of experiments, where we vary the histogram threshold (beam size) from the default value of 200 up to 800, while keeping all other parameters and feature weights fixed. The results in terms of BLEU, KRS and KRS-V are plotted against the beam size and reported in Table 6.6. Three Arabic-English systems are represented: baseline with a DL=8, baseline with DL=18 and our enhanced system that includes the WaW model and early reordering pruning with DL=18.

We can see that increasing the beam size has inconsistent effects on the low-DL baseline. The large-DL baseline, instead, appears to benefit from the increased beam size, nevertheless its performance remains the worst overall. Indeed, the superiority of the early-pruning system is maintained according to all metrics in all the tested beam settings, which shows that the success of our approach is due in large part to the reduction of model errors. These results confirm the usefulness of our method not only as an optimization technique, but also as a way to improve translation quality on top of a strong baseline regardless of efficiency.

Long-range reordering statistics and examples

To better understand the behavior of the early-pruning system, we extract phrase-to-phrase jump statistics from the decoder log file and count the number of long jumps that were performed to produce the 1-best translation. More precisely, we count a long jump for each pair of consecutively translated source phrases $(\tilde{f}_{i-1}, \tilde{f}_i)$ such that:

$$\mathrm{D} = |\mathrm{start}(\tilde{f}_i) - \mathrm{end}(\tilde{f}_{i-1}) - 1| > 5$$

Results are reported in Table 6.7, along with the average number of partial translation hypotheses considered per test sentence.

We can see that, in both languages, the early-pruning system performed several long jumps while exploring a much smaller search space compared to the high-distortion baseline. In Arabic-English, the high-distortion system performed almost the same number of long jumps when early pruning was enabled, but it considered four times less partial hypotheses (from 2424K to 642K per sentence). In German-English, instead, early pruning resulted in a smaller number of performed long jumps (e. g. from 48 to 19 per 100 sentences with D in [9..18]). At the same time, the higher translation quality scores reported in Table 6.5 suggest that the early-pruning system is more precise, while the high-distortion baseline is over-reordering. In German-English too, early pruning had the beneficial effect of shrinking the search space by a factor of four.

	System	**DL**	**#hyp/sent**	**(#jumps/sent)×100**		
				D: [6..8]	**[9..12]**	**[13..18]**
Arabic-English eval09-nw	baseline	8	1119K	14	–	–
	baseline	18	2424K	19	5	2
	+ WaW model and early reo.pruning (ϑ=5)	18	642K	16	5	2
German-English test10	baseline	8	634K	89	–	–
	baseline	18	1349K	92	66	48
	+ WaW model and early reo.pruning (ϑ=5)	18	385K	52	32	19

Table 6.7: Decoding statistics of the baseline and the new system: **#hyp/sent** is the average number of partial translation hypotheses considered per test sentence; **(#jumps/sent)×100** is the average number of phrase-to-phrase jumps included in the 1-best translation per 100 test sentences. Only long jumps are shown, divided into three distortion buckets.

Finally, Table 6.8 shows some examples of test sentences that were erroneuously reordered by the baseline systems. The systems including the WaW model and early pruning of reordering steps, instead, produced the correct translation.

The first Arabic sentence is a typical example of VSO order with a long subject. While the baseline system left the verb in its Arabic position, producing an incomprehensible translation, the new system placed it rightly between the subject and the object. This reordering involved two long jumps: one with D=9 backward and one with D=8 forward. The second sentence displays another, less common, Arabic construction: namely VOS, with the object realized by a personal pronoun. In this case, a backward jump with D=10 and a forward jump with D=8 were necessary to achieve the correct reordering.

The first German sentence contains a broken verb chunk: that is, the auxiliary verb (*hat*) is separated from the past participle (*geeinigt*) by the object and a very long complement. The new system was able to correctly translate and reorder the verb by performing a backward jump with D=15 and a forward jump with D=16. Finally, in the last sentence, the modal verb (*konnten*) is separate from the infinitive verb (*erreichen*) by the subject, the object and a complement. This example is further complicated by the presence of the negation (*nicht*). To produce the correct translation, the new system had to jump forward by D=12 and backward by D=14.

These examples show that our early reordering pruning technique can successfully handle very complex and diverse reordering patterns.

يواصل سفير المملكة العربة السعودية لدى لبنان عبدالعزيز خوجة تحرك ـه في اتجاه …

SRC	verb	subj.								obj.		compl.	
(ar)	**ywASl**	sfyr	Almmlkp	AlErbyp	AlsEwdyp	ldY	lbnAn	EbdAlEzyz	xwjp	tHrk	-h	fy	AtjAh...
	continues	*ambassador*	*Kingdom*	*Arabian*	*Saudi*	*to*	*Lebanon*	*Abdulaziz*	*Khawja*	*move*	*his*	*in*	*direction*

REF	The Kingdom of Saudi Arabia 's ambassador to Lebanon Abdulaziz Khawja **continues** his moves towards ...
BASE	**continue** to Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja its move in the direction of ...
NEW	The Kingdom of Saudi Arabia 's ambassador to Lebanon , Abdulaziz Khwja **continue** its move in the direction of ...

فيما دعا ـهم رئيس المكتب السياسي لـ حركة حماس خالد مشعل الى التزام الحياد

SRC	adv.	verb	obj.	subj.									compl.		
(ar)	fymA	**dEA**	**-hm**	r}ys	Almktb	AlsyAsy	l-	Hrkp	HmAs	xAld	m$El	AlY	AltzAm	AlHyAd
	meanwhile	*called*	*them*	*head*	*bureau*	*political*	*of*	*movement*	*Hamas*	*Khaled*	*Mashal*	*to*	*necessity*	*neutrality*

REF	Meanwhile, the Head of the Political Bureau of the Hamas movement , Khaled Mashal , **called upon them** to remain neutral .
BASE	The **called them** , head of Hamas' political bureau , Khalid Mashal , to remain neutral .
NEW	The head of Hamas' political bureau , Khalid Mashal , **called on them** to remain neutral .

SRC	subj.						verb$_{\text{aux}}$	obj.	compl.			
(de)	Die	obersten	Vertreter	des	amerikanischen	Kongresses	**haben**	sich	auf	eine	breitere	Form
	the	*top*	*representatives*	*of-the*	*American*	*Congress*	*have*	*themselves*	*on*	*a*	*broader*	*form*

	compl.(cont'd)											verb$_{\text{pp}}$	
	eines	Abkommens	über	eine	Finanz	hilfe	für	das	amerikanische	Finanz	system	**geeinigt**	.
	of-an	*agreement*	*about*	*a*	*financial*	*aid*	*for*	*the*	*American*	*financial*	*system*	*agreed*	

REF	The top representatives of the American Congress **have agreed** upon a broader form of the agreement on financial assistance for the American financial system .
BASE	The top representatives of the American Congress **has** on a broader form of an agreement on financial assistance for the American financial system , **agreed** .
NEW	The top representatives of the American Congress **have agreed** on a broader form of an agreement on financial assistance to the American financial system .

SRC	adv.	verb$_{\text{mod}}$	subj.	obj.									compl.		
(de)	Jedoch	**konnten**	sie	Kinder	in	Teilen	von	Helmand	und	Kandahar	im	Süden	aus	Sicherheit	grund
	however	*could*	*they*	*children*	*in*	*parts*	*of*	*Helmand*	*and*	*Kandahar*	*in*	*South*	*for*	*security*	*reasons*

	neg	verb$_{\text{inf}}$	
	nicht	**erreichen**	.
	not	*reach*	

REF	But they **could not reach** children in parts of Helmand and Kandahar in the south for security reasons.
BASE	However , they **were** children in parts of Helmand and Kandahar in the south, for security reasons .
NEW	However, they **could not reach** children in parts of Helmand and Kandahar in the south for security reasons.

Table 6.8: Long-range reordering examples showing improvements over the baseline system when the DL is raised to 18 and early pruning based on WaW reordering scores is enabled (NEW). The Arabic sentences are taken from eval09-nw and the German ones from test09.

6.5 Conclusions

In this chapter, we have presented a fully data-driven approach to improve the performance of a PSMT system on long reordering. We have trained a discriminative model to predict likely reordering steps in a way that is complementary to state-of-the-art PSMT reordering models. We have effectively integrated it into a PSMT decoder as an additional feature, ensuring that its total score over a complete translation hypothesis is consistent across different phrase segmentations. Lastly, we have proposed early pruning of reordering steps as a novel method to dynamically refine the input permutation space defined by standard reordering constraints. The core idea of this technique is to let the decoder alternate between hard reordering decisions (early pruning only based on the reordering model) and soft reordering decisions (regular pruning based on the combination all SMT models). In this way, we combine the benefits of fully integrated reordering models with those of monolingual pre-ordering methods.

The approach is easily portable to other language pairs, because it does not rely on language-specific rules, but only on widely available linguistic resources, such as POS taggers.[9]

Evaluated in Arabic-English and German-English against two strong news translation baselines, our approach leads to similar or even higher BLEU, METEOR and KRS scores at a very high distortion limit (18), which is by itself an important achievement. At the same time, the reordering of verbs, measured with a novel version of the KRS, is significantly improved, while decoding gets between 19% and 31% faster than the baseline.

[9] According to our experiments, shallow syntax annotation was beneficial but not essential for the success of the approach.

Chapter 7

Comparative Evaluation and Conclusions

Throughout this thesis, we have proposed various methods to address the problem of long-range reordering in phrase-based SMT, obtaining consistent improvements over the state of the art in two language pairs that display important reordering phenomena.

Our methods differ primarily in the kind of language-specific resources that they require: chunk-based reordering lattices (Chapter 4) and modified distortion matrices (Chapter 5) need a POS-tagger, a shallow syntax chunker and a set of hand-written reordering rules to be constructed. On the contrary, early reordering pruning based on the Word-after-Word (WaW) reordering model (Chapter 6) does not require rules. It only employs POS and, if available, chunking annotation.

From the point of view of time efficiency, the lattice solution appears to be the most costly: decoding becomes three times slower than the baseline when unpruned lattices are used. To limit this effect, lattices can be pruned by means of discriminative methods, but the pruning phase itself can also be expensive (cf. Section 4.6). On the other hand, distortion matrices and WaW-based early pruning are both very competitive in terms of run times.

To fairly compare the proposed methods at the level of translation quality and efficiency, we will now present a final series of experiments performed with exactly the same training and decoding conditions. Moreover, to position our work in the broader field of SMT, we will also evaluate our methods against the hierarchical SMT (HSMT) approach [Chiang, 2005].

7.1 Experimental setup

The comparative evaluation is carried out on the Arabic-English NIST-MT09 task and the German-English WMT10 task. The SMT training, development and test corpora used in this chapter, as well as the pre-processing pipelines, are the same as those used in the evaluations of Chapters 5 and 6. The baseline PSMT setting corresponds to that of Chapter 6, that is our strongest PSMT system including hierarchical phrase orientation models [Galley and Manning, 2008] and early distortion cost [Moore and Quirk, 2007]. The contrastive experiments are set up as follows:

Lattices. The chunk-based reordering rules described in Sections 4.2 and 5.2 are applied deterministically to the training data before phrase extraction and scoring. The same rules are then applied non-deterministically to the test sets, and the resulting reorderings are represented explicitly in the form of word reordering lattices. Translation is performed by non-monotonic lattice decoding [Dyer et al., 2008]. Before translation, the lattices are pruned according to the scores of a chunk-based reordered source LM, as explained in Sections 5.3 and 5.5.1.[1] We choose this pruning technique because it is much faster than the one based on the SVM classifier proposed in Section 4.6. Lattice edges are assigned a score of 1 if they belong to the original order path, or 0.25 otherwise. The lattice path feature weight is then tuned by MERT along with the other feature weights. Lattice decoding is not compatible with early distortion cost,[2] therefore we use standard distortion in this experiment only.

Matrices. The same chunk-based reordering rules are applied to the test sets, but the resulting reorderings are represented implicitly by means of modified distortion matrices. Before computing the matrices, reorderings are pruned with a chunk-based reordered LM as in the lattice experiment. Thus, the reorderings encoded by the matrices correspond to those encoded by the lattices. In this experiment we enable early distortion cost, therefore the modified distortion cost is not used as a feature function but only as a constraint: that is, to select the set of positions allowed for hypothesis expansion.

Because the input sentence is presented to the decoder in its original order, the training data is not reordered in this and the following experiments.

[1] This technique was designed for modified distortion matrix, but can be applied to lattices as well.

[2] This is because exact distortion between pairs of word nodes is not well defined in non-linear lattices. The approach proposed by Dyer et al. [2008], and implemented in Moses, is to use the shortest possible path pre-computed using an all-pairs shortest path algorithm.

WaW early reordering pruning. This system includes the WaW reordering model as an additional feature function. Besides, early pruning of reordering steps based on WaW scores is applied to each hypothesis expansion. All settings (pruning parameters etc.) coincide with those of the early-pruning experiments presented in Table 6.5.

Hierarchical SMT. For each language pair, we build a HSMT system on the same training data, using the tree-based implementation of Moses [Hoang et al., 2009]. As explained in Section 2.3, the number of words that may be covered by non-terminal symbols has to be limited for efficiency reasons (span constraint). We set this constraint to the default value of 10 words for rule extraction, while for decoding we consider two settings: the default 10 words and a large value of 20 to enable very long-range reorderings.

The feature weights of all our systems are optimized by MERT [Och, 2003] on the Arabic-English dev06-nw and the German-English test08. Each configuration is tuned four times and the average of the resulting weight vectors is used to translate the test sets, as suggested by Cettolo et al. [2011]. We evaluate global translation quality by **BLEU** and **METEOR**, and reordering accuracy by generic and verb-specific Kendall Reordering Scores (**KRS** and **KRS-V**).[3] The source-reference word alignments needed to compute the reordering scores are generated by the alignment models previously trained on the training data. The source-output word alignments are obtained from the decoder's trace in all experiments except the one involving reordering lattices, for which we have to make use of the pre-trained alignment models. Statistical significance is assessed by approximate randomization as in Riezler and Maxwell [2005].

The results of the comparative evaluation are presented in Tables 7.1 and 7.3. In the upper part of each table, statistical significance is computed against the baseline PSMT system (DL=8). In the lower part, instead, our best reordering method is individually compared against both the PSMT and the HSMT baselines. Run times refer to the translation of the first 500 sentences of eval08-nw and test09 by an Intel Xeon X5650 processor. To allow for a finer evaluation of decoding efficiency, model loading times are subtracted from the total translation time.

Note that the German-English scores are overall lower because only one translation reference is available, as opposed to four in Arabic-English. As for run times, they are overall higher because of the larger language model.

[3] See Section 2.5 for details on all evaluation metrics except the KRS-V, which is introduced in Section 6.4.2.

7.2 Arabic-English results

The results in the first block of Table 7.1 prove once again the coarseness of distance-based reordering constraints: that is, when the distortion limit is raised to 18 words, translation quality decreases and decoding becomes almost two times slower than the baseline.

Looking at the second block, we see that reordering **lattices** fail to improve the PSMT baseline in this experimental set up. These results are in contrast with the findings of Chapter 4, where both the unpruned and pruned lattices were consistently outperforming the baseline. This is partly explained by the fact that our latest baseline includes early distortion cost but lattice decoding does not support it, which makes the comparison somewhat unfair. Another difference with respect to our earlier evaluation is the use of hierarchical phrase orientation models. These are particularly beneficial in Arabic-English, which makes the baseline harder to beat. As regards efficiency, the lattice solution – even when the lattices are pruned – is very expensive. This is because the lattice representation implies the multiplication of input word nodes: that is, the same word has to be decoded multiple times if it appears in different positions of the lattice.

Next in the comparison are modified distortion **matrices**. Results by this method are slightly better than those achieved by the lattice solution, while decoding is almost four times faster. In particular, the reordering of verbs measured by KRS-V improves on all test sets. With respect to the baseline, most differences are not statistically significant, however we report a significant gain on the KRS-V of the reordering-specific subset reo09 (from 84.8 to 85.2), which is where improvements by our methods are mostly expected.

WaW early reordering pruning is our last proposed technique – fully data-driven and integrated into the decoding process. Results by this method are the same as those reported in Table 6.5(b) except for the run time, which is recomputed on a larger sentence sample excluding model loading time. As observed in Chapter 6, early reordering pruning makes it possible to preserve or even increase performances when raising the DL to a very high value. We report small gains in terms of BLEU and METEOR (but only the METEOR gain on eval09-nw is statistically significant). But more importantly, the reordering scores increase on all test sets, with larger improvements concentrating on the reordering of verbs: that is a gain of +0.7, +0.8 and +1.4 KRS-V on eval08-nw, eval09-nw and reo09 respectively. Notice that these scores are also significantly higher than the low-distortion baseline, proving that jumps longer than 5 are effectively performed by the early-pruning system. This system is also very efficient: that is 22% faster than the DL8-baseline.

The following block shows the results achieved by the **hierarchical SMT** system,

Arabic-English	eval08-nw				eval09-nw				reo09	**ms/**
	bleu	**met**	**krs**	**krs-v**	**bleu**	**met**	**krs**	**krs-v**	**krs-v**	**word**
All systems against the PSMT baseline:										
Phrase-based SMT										
DL=5	44.7	35.1▼	82.9▼	84.7▼	50.3▽	38.1	84.6	85.9	84.7	59
[**baseline**] DL=8	44.8	35.2	83.4	85.6	50.6	38.1	84.7	86.2	84.8	87
DL=18	44.7	34.9▼	82.4▼	84.9▼	50.3	38.0▽	83.9▼	85.8▽	84.3	164
Reordering lattices										
DL=5	44.9	35.0▽	83.1	85.1	50.4	38.1	84.4▽	85.9	84.7	229
Modif. disto. matrices										
DL=5	44.8	35.1	83.4	85.8	50.7	38.1	84.8	86.3	85.2△	60
WaW early reo. pruning										
ϑ=5 | DL=18	45.0	35.3	83.8△	86.3▲	50.9	38.3▲	84.9	87.0▲	86.2▲	68
Hierarchical SMT										
max.span=10	44.2▼	35.0▽	81.9▼	84.7▼	49.9▼	38.1	83.7▼	86.5	85.9▲	137
max.span=20	44.0▼	35.1	82.7▼	85.3	50.2	38.2	84.2▼	86.8	85.8▲	325
System-to-system comparisons:										
WaW early reo. pruning										
◇ *versus PSMT base.DL=8*	+0.2	+0.1	+0.4△	+0.7▲	+0.3	+0.2▲	+0.2	+0.8▲	+1.4▲	-22%
◇ *versus HSMT m.span=10*	+0.8▲	+0.3▲	+1.9▲	+1.6▲	+1.0▲	+0.2▲	+1.2▲	+0.5	+0.3	-50%

Table 7.1: Comparison of the proposed reordering techniques against a baseline phrase-based system and a tree-based system, in Arabic-English. Translation quality is measured with % BLEU, METEOR, and KRS: regular (KRS) and verb-specific (KRS-V). Statistically significant differences with respect to the baseline are marked with ▲▼ at the $p \leq .05$ level and △▽ at the $p \leq .10$ level. Decoding time is measured in milliseconds per input word.

which has a totally different approach to word reordering. In fact, as explained in Section 2.3, the HSMT decoder builds the target sentence in a bottom-up fashion rather than from left to right. Moreover, all reordering information is embedded in the translation rules. At the level of translation modeling, the HSMT approach is advantaged by the ability to learn many more translation units from the same training data and to reuse them in a variety of contexts (i. e. discontinuos phrases). Nevertheless, the results of our experiments confirm previous evidence [Zollmann et al., 2008, Birch et al., 2009] on the superiority of PSMT over HSMT in Arabic-English. The only exception to this trend is a significantly higher verb reordering accuracy achieved by the HSMT system on reo09: that is, 85.9 versus 84.8 KRS-V by the PSMT baseline. This is an important result from the point of view of reordering, however the low reordering scores reported on the generic test sets suggest that the HSMT system tends to perform excessive reordering. If we relax the span constraint from 10 to 20 to enable long-range reorderings comparable to those performed by a PSMT system with DL=18, the scores appear to change inconsistently. Overall, HSMT performance remains slightly worse than that of the PSMT baseline. Notice moreover the dramatic increase of translation time when a larger reordering space is explored by the HSMT system (from 137 to 325 ms/word).

In the last block of the table, we directly compare our best method – WaW-based early reordering pruning – against the PSMT and HSMT baselines, and find that our system achieves consistently higher translation quality while being significantly faster: namely, decoding time is 22% and 50% shorter, respectively.

<table>
<tr><td></td><td colspan="14">يواصل سفير المملكة العربية السعودية لدى لبنان عبدالعزيز خوجة تحرك ـه في اتجاه ...</td></tr>
<tr><td>SRC</td><td>verb</td><td colspan="8">subj.</td><td>obj.</td><td colspan="4">compl.</td></tr>
<tr><td>(ar)</td><td>**ywASl**</td><td>sfyr</td><td>Almmlkp</td><td>AlErbyp</td><td>AlsEwdyp</td><td>ldY</td><td>lbnAn</td><td>EbdAlEzyz</td><td>xwjp</td><td>tHrk -h</td><td colspan="4">fy AtjAh...</td></tr>
<tr><td></td><td>*continues*</td><td>*ambassador*</td><td>*Kingdom*</td><td>*Arabian*</td><td>*Saudi*</td><td>*to*</td><td>*Lebanon*</td><td>*Abdulaziz*</td><td>*Khawja*</td><td>*move his*</td><td colspan="4">*in direction*</td></tr>
<tr><td>REF</td><td colspan="14">The Kingdom of Saudi Arabia 's ambassador to Lebanon Abdulaziz Khawja **continues** his moves towards ...</td></tr>
<tr><td>BASE</td><td colspan="14">**continue** to Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja its move in the direction of ...</td></tr>
<tr><td>LAT</td><td colspan="14">Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja **continue** to move in the direction of ...</td></tr>
<tr><td>MAT</td><td colspan="14">The Kingdom of Saudi Arabia 's ambassador to Lebanon , Abdulaziz Khwja **continue** to move in the direction of ...</td></tr>
<tr><td>WRP</td><td colspan="14">The Kingdom of Saudi Arabia 's ambassador to Lebanon , Abdulaziz Khwja **continue** its move in the direction of ...</td></tr>
<tr><td>H10</td><td colspan="14">**continue** to Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja its move in the direction of ...</td></tr>
<tr><td>H20</td><td colspan="14">**continue** to Saudi Arabian ambassador to Lebanon , Abdulaziz Khwja its move in the direction of ...</td></tr>
<tr><td></td><td colspan="14">فيما دعا ـهم رئيس المكتب السياسي لـ حركة حماس خالد مشعل الى التزام الحياد</td></tr>
<tr><td>SRC</td><td>adv.</td><td>verb</td><td>obj.</td><td colspan="8">subj.</td><td colspan="3">compl.</td></tr>
<tr><td>(ar)</td><td>fymA</td><td>**dEA**</td><td>**-hm**</td><td>r}ys</td><td>Almktb</td><td>AlsyAsy</td><td>l-</td><td>Hrkp</td><td>HmAs</td><td>xAld</td><td>m$El</td><td>AlY</td><td>AltzAm</td><td>AlHyAd</td></tr>
<tr><td></td><td>*meanwhile*</td><td>*called*</td><td>*them*</td><td>*head*</td><td>*bureau*</td><td>*political*</td><td>*of*</td><td>*movement*</td><td>*Hamas*</td><td>*Khaled*</td><td>*Mashal*</td><td>*to*</td><td>*necessity*</td><td>*neutrality*</td></tr>
<tr><td>REF</td><td colspan="14">Meanwhile, the Head of the Political Bureau of the Hamas movement , Khaled Mashal , **called upon them** to remain neutral .</td></tr>
<tr><td>BASE</td><td colspan="14">The **called them** , head of Hamas' political bureau , Khalid Mashal , to remain neutral .</td></tr>
<tr><td>LAT</td><td colspan="14">The head of Hamas' political bureau , Khalid Mashal **called on them** to remain neutral .</td></tr>
<tr><td>MAT</td><td colspan="14">The head of Hamas' political bureau , Khalid Mashal , **called on them** to remain neutral .</td></tr>
<tr><td>WRP</td><td colspan="14">The head of Hamas' political bureau , Khalid Mashal , **called on them** to remain neutral .</td></tr>
<tr><td>H10</td><td colspan="14">**called them** , head of Hamas' political bureau , Khalid Mashal , commitment to neutrality .</td></tr>
<tr><td>H20</td><td colspan="14">The head of Hamas ' political bureau , Khalid Mashal , **called them** to remain neutral .</td></tr>
</table>

Table 7.2: Long-range reordering examples showing the behavior of different systems on Arabic-English: [BASE] is the baseline PSMT system; [LAT] refers to the lattice experiment; [MAT] refers to the modified distortion matrix experiment; [WRP] refers to the WaW early reordering pruning experiment; [H10] and [H20] are HSMT systems with a span constraint of 10 and 20 words respectively.

To conclude, we consider again the examples of Chapter 6 and report them in Table 7.2 including also the output of the systems evaluated in this chapter. In both sentences, the long-range reordering of the verb was missed by the PSMT baseline but captured by all the proposed reordering methods (LAT, MAT and WRP). As for the HSMT systems (H10 and H20), both of them failed to reorder the verb in the first sentence, whereas in the second sentence only the system with a large span constraint (H20) produced the correct word order.

7.3 German-English results

We start by observing the behavior of the PSMT system in different DL conditions. Although some scores increase when the DL is reduced to 5, we decide to consider DL=8 as our baseline setting because of the higher BLEU scores on test10 and a much higher KRS-V on reo10. On the other hand, raising the DL to a high value (18) has a very bad impact on both efficiency and translation quality.

We then examine the performance of our reordering techniques. Differently from Arabic-English, all the proposed techniques – lattices and matrices included – appear to outperform the PSMT baseline, even if this is especially strong from the reordering point of view (i. e. including hierarchical phrase orientation models and early distortion cost). In particular, **lattices** and **matrices** achieve statistically significant improvements according to all metrics. While the gains in the generic metrics are admittedly small (ranging between +0.2 and +0.7 BLEU and between +0.2 and +0.4 METEOR), the improvement is clearly visible in the reordering scores (ranging between +0.6 and +1.3 KRS and between +0.5 and +3.7 KRS-V). In this regard, we recall that the main goal of our work is to specifically improve the word reordering aspect of translation without introducing other errors, therefore our measure of success is precisely to obtain higher reordering scores with no loss in the generic scores.

The fact that lattices and matrices are more beneficial in German-English than in Arabic-English can be partly explained by the more precise reordering rule set,[4] which reduces the risk of discarding correct reorderings before lattice or matrix construction. Among our three reordering techniques, the lattices appear as the most accurate but also as the most costly in terms of decoding time. Where efficiency is paramount, the matrices or the WaW-pruning technique may be used with slightly lower performances.

Lastly, we directly compare our best method – lattices – against both the PSMT

[4] We recall that our chunk-based rules generate 3 reorderings per sentence in German-English versus 22 in Arabic-English (cf. Section 5.2).

German-English	test09				test10				reo10	**ms/**
	bleu	**met**	**krs**	**krs-v**	**bleu**	**met**	**krs**	**krs-v**	**krs-v**	**word**
All systems against the PSMT baseline:										
Phrase-based SMT										
DL=5	19.0	27.5	66.7▲	64.6▲	20.1▼	29.1▼	70.0▲	67.2	62.8▼	155
[**baseline**] DL=8	19.0	27.4	66.1	64.2	20.4	29.2	69.2	67.1	63.9	202
DL=18	18.0▼	27.3▽	61.7▼	61.0▼	19.3▼	29.1▼	64.4▼	63.8▼	61.2▼	408
Reordering lattices										
DL=5	19.3▲	27.6▲	67.2▲	65.3▲	21.1▲	29.6▲	70.5▲	68.3▲	67.6▲	260
Modif. disto. matrices										
DL=5	19.2	27.6▲	66.7▲	64.7▲	20.8▲	29.4▲	69.9▲	68.0▲	66.8▲	143
WaW early reo. pruning										
ϑ=5 | DL=18	19.4▲	27.7▲	66.5▲	64.9▲	20.6▲	29.5▲	69.5▲	67.8▲	65.9▲	142
Hierarchical SMT										
max-span=10	19.7▲	27.7▲	67.0▲	65.6▲	21.4▲	29.7▲	70.1▲	68.2▲	64.8▲	406
max-span=20	19.8▲	27.8▲	66.7▲	65.4▲	21.3▲	29.7▲	69.8▲	68.3▲	65.8▲	706
System-to-system comparisons:										
Reordering lattices										
◇ *versus PSMT base.DL=8*	+0.3▲	+0.2▲	+1.1▲	+1.1▲	+0.7▲	+0.4▲	+1.3▲	+1.2▲	+3.7▲	+29%
◇ *versus HSMT m.span=10*	-0.4▼	-0.1▽	+0.2	-0.3	-0.3▽	-0.1	+0.4▲	+0.1	+2.8▲	-36%

Table 7.3: Comparison of the proposed reordering techniques against a baseline phrase-based system and a tree-based system, in German-English. Translation quality is measured with % BLEU, METEOR, and KRS: regular (KRS) and verb-specific (KRS-V). Statistically significant differences with respect to the baseline are marked with ▲▼ at the $p \leq .05$ level and △▽ at the $p \leq .10$ level. Decoding time is measured in milliseconds per input word.

and the HSMT baseline. The improvement over the PSMT baseline is reflected by all evaluation metrics in all test sets. In particular, we report a great increase in verb reordering accuracy on the reordering-specific test set reo09 (+3.7 KRS-V). We recall that the lattice system does not use early distortion cost, thus we would get even higher gains if we compared it with a standard-distortion baseline.

Looking at the very last row of the table, we see that the lattice system achieves lower BLEU scores than the HSMT sysyem, but slightly higher KRS and a much higher KRS-V on the reordering-specific subset. This last result is probably due to the fact that the span constraint of 10 prevents the HSMT system from capturing very long-range reorderings. If we relax the span constraint to 20, the KRS-V on reo09 increases from 64.8 to 65.8, but remains much lower than the one achieved by the lattice system (67.6). Notice, moreover, that the larger span constraint results in a critical slowdown of the decoding process.

To sum up, these results challenge the idea that HSMT is more suitable than PSMT for

SRC (de)	subj. verb$_{\text{aux}}$ obj. compl. Die obersten Vertreter des amerikanischen Kongresses **haben** sich auf eine breitere Form *the top representatives of-the American Congress have themselves on a broader form* compl.(cont'd) verb$_{\text{pp}}$ eines Abkommens über eine Finanz hilfe für das amerikanische Finanz system **geeinigt** . *of-an agreement about a financial aid for the American financial system agreed*
REF	The top representatives of the American Congress **have agreed** upon a broader form of the agreement on financial assistance for the American financial system .
BASE	The top representatives of the American Congress **has** on a broader form of an agreement on financial assistance for the American financial system , **agreed** .
LAT	The top representatives of the American Congress **have agreed** on a broader form of an agreement on financial assistance to the American financial system .
MAT	The top representatives of the American Congress **have agreed** on a broader form of an agreement on financial assistance to the American financial system .
WRP	The top representatives of the American Congress **have agreed** on a broader form of an agreement on financial assistance to the American financial system .
H10	The top representatives of the American Congress , **have** an agreement on a form on a broader financial aid for the American financial system , **agreed** .
H20	The top representatives of the American Congress **have agreed** on an agreement on a form a broader financial support for the US financial system .
SRC (de)	adv. verb$_{\text{mod}}$ subj. obj. compl. Jedoch **konnten** sie Kinder in Teilen von Helmand und Kandahar im Süden aus Sicherheit grund *however could they children in parts of Helmand and Kandahar in South for security reasons* neg verb$_{\text{inf}}$ **nicht** **erreichen** . *not reach*
REF	But they **could not reach** children in parts of Helm. and Kand. in the south for security reasons.
BASE	However, they **were** children in parts of Helm. and Kand. in the south, for security reasons.
LAT	However, they **have not been able to reach** children in parts of Helm. and Kand. in the south for security reasons.
MAT	However, they **could not reach** the children in parts of Helm.and Kand. in the south for security reasons.
WRP	However, they **could not reach** children in parts of Helm. and Kand. in the south for security reasons.
H10	However, they **were** children in parts of Helm. and Kand. in the south **not reach** for security reasons.
H20	However, they **were** children in parts of Helm. and Kand. in the south **not reach** for security reasons.

Table 7.4: Long-range reordering examples showing the behavior of different systems on German-English: [BASE] is the baseline PSMT system; [LAT] refers to the lattice experiment; [MAT] refers to the modified distortion matrix experiment; [WRP] refers to the WaW early reordering pruning experiment; [H10] and [H20] are HSMT systems with a span constraint of 10 and 20 words respectively.

the German-English language pair. In fact, HSMT appears to incur similar problems as PSMT as far as long-range reordering is concerned: that is, existing reordering constraints are simply too coarse-grained to define an adequate reordering search space. On the other hand, the proposed PSMT enhancements appear as a valuable way to improve the handling of long-range reordering phenomena without sacrificing efficiency.

The German-English translation examples are reported in Table 7.4. In both sentences, our three reordering methods (LAT, MAT and WRP) were able to capture the very long-range reordering of the verb, as opposed to the baseline. Among the HSMT systems, only the one with a large span constraint (20) could correctly reorder the first sentence, whereas both failed in the second sentence.

7.4 Conclusions and future research directions

Natural languages vary greatly in how they arrange sentence constituents. Since the emergence of the first statistical machine translation methods, researchers have tried to solve this problem with various modeling strategies, and by heuristically restricting the possible word reordering operations. Still up to date, no method appears to be dominant across different language pairs.

In this thesis, we have proposed a number of techniques to advance the state of the art in reordering modeling within the phrase-based SMT framework. Our techniques differ primarily in the kind of language-specific resources that they require. All, however, share the goal of improving the definition of the reordering search space based on the characteristics of a specific language pair. In fact, in the absence of perfect reordering models, effectively restraining the set of explorable reordering hypotheses is key to the success of SMT. Being mostly complementary to the design of better reordering feature functions, our work can take advantage of the most recent advances in reordering modeling and improve SMT performances on top of them.

To guide our research, we have first examined the reordering characteristics of various language pairs, from a qualitative perspective. We have then chosen to focus specifically on language pairs with uneven distributions of reordering phenomena – that is, where reordering is predominantly local with the exception of few isolated long-range reordering patterns that are crucial to preserve the general meaning of a sentence.

Evaluated in large-scale news translation tasks, our techniques have proven successful for two very different language pairs: namely, Arabic-English and German-English. In particular, we were able to obtain significant improvements in the reordering-specific metrics while preserving – or sometimes even increasing – the generic translation quality scores. As illustrated by the examples, these results are due to very targeted changes, which are nevertheless essential for understanding the translated text.

Our best PSMT systems also appear to compare favorably with a competitive tree-based SMT approach, in terms of both quality and efficiency.

While we have clearly proved the importance of refining the reordering search space, there are still several ways to improve and extend our work. For instance, the techniques that make use of language-specific fuzzy reordering rules – lattices and matrices – could benefit from the development of more precise rules that exploit POS and lexical clues, especially in Arabic-English.

As for the early reordering pruning technique, it could profit from more accurate reordering scores. To this end, the WaW model could be improved by using different feature templates and granularities, such as automatically learnt word classes. Additionally, other kinds of reordering model scores (e. g. phrase orientation [Koehn et al., 2005] or pairwise word order [Tromble and Eisner, 2009]) may be combined to the WaW model score for the purpose of early pruning.

Concerning the language choice, we would like to apply our data-driven methods to language pairs with similar reordering characteristics, such as Arabic-French or Dutch-English, but also to language pairs with global reordering phenomena, such as Chinese-English or Turkish-English, to find out whether the gap between PSMT and HSMT can at least be narrowed.

As a concrete application of our work, we plan to integrate the proposed methods to an online PSMT system with high efficiency requirements. In particular, we would like to explore how the post-editing effort of human translators could be reduced by improving the word reordering accuracy of an SMT component included in a computer assisted translation tool.

We conclude with a consideration that has emerged during the last stages of this thesis. In line with previous findings, our experimental results suggest that HSMT also suffers from the coarseness of reordering constraints – an issue that has begun to be studied only recently [Braune et al., 2012]. If the problem of defining the reordering search space is common to both the PSMT and HSMT approaches, we are confident that the ideas proposed in this thesis can inspire analogous solutions in the tree-based SMT field.

Bibliography

Yaser Al-Onaizan and Kishore Papineni. Distortion Models for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, July 2006. Association for Computational Linguistics.

Jacob Andreas, Nizar Habash, and Owen Rambow. Fuzzy Syntactic Reordering for Phrase-based Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 227–236, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Srinivas Bangalore and Giuseppe Riccardi. Finite-state models for lexical reordering in spoken language translation. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 422–425, Beijing, China, 2000.

A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, and R. L. Mercer. Language translation apparatus and method of using context-based translation models. United States Patent, No. 5510981, Apr 1996.

Daniel M. Bikel. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4): 479–511, 2004. ISSN 0891-2017.

Alexandra Birch, Miles Osborne, and Philipp Koehn. Predicting success in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

Alexandra Birch, Phil Blunsom, and Miles Osborne. A quantitative analysis of reordering phenomena. In *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 197–205, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Alexandra Birch, Miles Osborne, and Phil Blunsom. Metrics for MT evaluation: evaluating reordering. *Machine Translation*, 24(1):15–26, 2010.

Arianna Bisazza and Marcello Federico. Chunk-based Verb Reordering in VSO Sentences for Arabic-English Statistical Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 241–249, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Arianna Bisazza and Marcello Federico. Modified Distortion Matrices for Phrase-Based Statistical Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 478–487, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Arianna Bisazza and Marcello Federico. Dynamically Shaping the Reordering Search Space of Phrase-Based Statistical Machine Translation. *Transactions of the Association for Computational Linguistics* – under revision, 2013.

Arianna Bisazza, Daniele Pighin, and Marcello Federico. Chunk-lattices for verb reordering in Arabic-English Statistical Machine Translation. *Machine Translation, Special Issue on MT for Arabic*, 26(1-2):85–103, 2012.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual Workshop on Computational learning theory*, 1992.

Fabienne Braune, Anita Gojun, and Alexander Fraser. Long-distance reordering during search for hierarchical phrase-based SMT. In *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, pages 28–30, Trento, Italy, 2012.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer, and P. Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–312, 1993.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, Uppsala, Sweden, July 2010.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June 2012. Association for Computational Linguistics.

Michael Carl. Patterns of shallow text production in translation. In Bernadette Sharp, Michael Zock, Michael Carl, and Arnt Lykke Jakobsen, editors, *Proceedings of the 8th international NLPSC workshop. Special theme: Human-machine interaction in translation. (Copenhagen Studies in Language 41)*, pages 143–151, Copenhagen, Denmark, 2011.

Marine Carpuat, Yuval Marton, and Nizar Habash. Improving Arabic-to-English SMT by Reordering Post-Verbal Subjects for Alignment. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 178–183, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Marine Carpuat, Yuval Marton, and Nizar Habash. Improved Arabic-to-English statistical machine translation by reordering post-verbal subjects for word alignment. *Machine Translation, Special Issue on MT for Arabic*, 26(1-2):105–120, 2012.

Francisco Casacuberta, Marcello Federico, Hermann Ney, and Enrique Vidal. Recent Efforts in Spoken Language Processing. *IEEE Signal Processing Magazine*, 25(3):80–88, May 2008.

Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. Methods for smoothing the optimizer instability in SMT. In *MT Summit XIII: the Thirteenth Machine Translation Summit*, pages 32–39, Xiamen, China, 2011.

Yin-Wen Chang and Michael Collins. Exact Decoding of Phrase-Based Translation Models through Lagrangian Relaxation. In *Proceedings of the 2011 Conference on Empirical*

Methods in Natural Language Processing, pages 26–37, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Boxing Chen, Mauro Cettolo, and Marcello Federico. Reordering rules for phrase-based statistical machine translation. In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, November 2006.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393, 1999.

Colin Cherry, Robert C. Moore, and Chris Quirk. On Hierarchical Re-ordering and Permutation Parsing for Phrase-based Decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 200–209, Montréal, Canada, June 2012. Association for Computational Linguistics.

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

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2): 201–228, 2007.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the Association for Computational Lingustics*, ACL 2011, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

Michael Collins and Nigel Duffy. Convolution Kernels for Natural Language. In *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press, 2001.

Michael Collins, Philipp Koehn, and Ivona Kucerova. Clause Restructuring for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. Statistical Machine Reordering. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July 2006. Association for Computational Linguistics.

Olivia Craciunescu, Constanza Gerding-Salas, and Susan Stringer-O'Keeffe. Machine translation and computer-assisted translation: a new way of translating? *Translation Journal*, 8(3), 2004.

Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, March 2003. ISSN 1532-4435.

Josep M. Crego and Nizar Habash. Using shallow syntax information to improve word alignment and reordering for SMT. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 53–61, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1.

Hal Daumé III. Notes on CG and LM-BFGS Optimization of Logistic Regression. Paper available at `http://pub.hal3.name`, implementation available at `http://hal3.name/megam`, 2004.

John DeNero and Jakob Uszkoreit. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 193–203, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 149–152, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Matthew S. Dryer. Word order. In Timothy Shopen, editor, *Clause Structure, Language Typology and Syntactic Description*, volume 1, chapter 2, pages 61–131. Cambridge University Press, second edition, 2007.

Matthew S. Dryer. Order of Subject, Object and Verb. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich, 2011.

Matthew S. Dryer and Martin Haspelmath, editors. *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich, 2011 edition, 2011.

İlknur Durgar El-Kahlout and Kemal Oflazer. Initial Explorations in English to Turkish Statistical Machine Translation. In *Proceedings on the Workshop on Statistical Machine*

Translation, pages 7–14, New York City, June 2006. Association for Computational Linguistics.

Chris Dyer and Philip Resnik. Context-free reordering, finite-state translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 858–866, Los Angeles, California, June 2010. Association for Computational Linguistics.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. Generalizing Word Lattice Translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Jakob Elming and Nizar Habash. Syntactic Reordering for English-Arabic Phrase-Based Machine Translation. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77, Athens, Greece, March 2009. Association for Computational Linguistics.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, pages 1618–1621, Melbourne, Australia, 2008.

Minwei Feng, Arne Mauser, and Hermann Ney. A Source-side Decoding Sequence Model for Statistical Machine Translation. In *Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, Colorado, USA, 2010a.

Yang Feng, Haitao Mi, Yang Liu, and Qun Liu. An Efficient Shift-Reduce Decoding Algorithm for Phrased-Based Machine Translation. In *COLING (Posters)*, pages 285–293, 2010b.

Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. Accurate Non-Hierarchical Phrase-Based Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June 2010. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

Dmitriy Genzel. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 376–384, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Daniel Gile. *La Traduction. La comprendre, l'apprendre.* Presses Universitaires de France, 2005.

Anita Gojun and Alexander Fraser. Determining the placement of German verbs in English-to-German SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 726–735, Avignon, France, April 2012. Association for Computational Linguistics.

Spence Green, Conal Sathi, and Christopher D. Manning. NP subject detection in verb-initial Arabic clauses. In *Proceedings of the Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3)*, Ottawa, Canada, 2009.

Spence Green, Michel Galley, and Christopher D. Manning. Improved Models of Distortion Cost for Statistical Machine Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 867–875, Los Angeles, California, 2010. Association for Computational Linguistics.

Deepa Gupta, Mauro Cettolo, and Marcello Federico. POS-based reordering models for statistical machine translation. In *In Proceedings of MT Summit XI*, pages 207–213, Copenhagen, Denmark, 2007.

Nizar Habash. Syntactic preprocessing for statistical machine translation. In Bente Maegaard, editor, *Proceedings of the Machine Translation Summit XI*, pages 215–222, Copenhagen, Denmark, 2007.

Nizar Habash and Fatiha Sadat. Arabic Preprocessing Schemes for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52, New York City, USA, June 2006. Association for Computational Linguistics.

Christian Hardmeier, Arianna Bisazza, and Marcello Federico. FBK at WMT 2010: Word Lattices for Morphological Reduction and Chunk-based Reordering. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 88–92, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Saša Hasan and Hermann Ney. A multi-genre SMT system for Arabic to French. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco, 2008.

David Haussler. Convolution kernels on discrete structures. Technical report, Dept. of Computer Science, University of California at Santa Cruz, 1999.

Magnus R. Hestenes and Eduard Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

Hieu Hoang, Philipp Koehn, and Adam Lopez. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *International Workshop on Spoken Language Translation (IWSLT)*, pages 152–159, Tokyo, Japan, 2009.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA, October 2010. Association for Computational Linguistics.

H. Johnson, J. Martin, G. Foster, and R. Kuhn. Improving translation quality by discarding most of the phrasetable. In *In Proceedings of EMNLP-CoNLL 07*, pages 967–975, 2007.

Dan Klein and Christopher D. Manning. Fast Exact Inference with a Factored Model for Natural Language Parsing. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10. MIT Press, Cambridge, MA, 2003.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, 2007.

Philipp Koehn. *Statistical Machine Translation.* Cambridge University Press, 2010.

Philipp Koehn and Hieu Hoang. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133, Edmonton, Canada, 2003.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proc. of the International Workshop on Spoken Language Translation*, October 2005.

Kimmo Koskenniemi and Mariikka Haapalainen. *GERTWOL – Lingsoft Oy*, chapter 11, pages 121–140. Roland Hausser, Niemeyer, Tübingen, 1994.

Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th international conference on World Wide Web*, WWW '10, pages 571–580, New York, NY, USA, 2010. ACM.

Ying Li. Three Sensitive Positions and Chinese Complex Sentences: A Comparative Perspective. *Journal of Chinese Language and Computing*, 18(2):47–59, 2008.

Percy Liang, Ben Taskar, and Dan Klein. Alignment by Agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June 2006. Association for Computational Linguistics.

Adam Lopez and Philip Resnik. Word-Based Alignment, Phrase-Based Translation: What's the Link? In *5th Conference of the Association for Machine Translation in the Americas (AMTA)*, Boston, Massachusetts, August 2006.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Arul Menezes and Chris Quirk. Dependency Treelet Translation: The Convergence of Statistical and Example-based Machine-translation? In *Proceedings of the Workshop on Example-based Machine Translation at MT Summit X*, Phuket, Thailand, September 2005.

Robert C. Moore and Chris Quirk. Faster beam-search decoding for phrasal statistical machine translation. In *In Proceedings of MT Summit XI*, pages 321–327, Copenhagen, Denmark, 2007.

Brian Mossop. An alternative to 'Deverbalization'. Technical report, York University, 2003.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. Inducing a Discriminative Parser to Optimize Machine Translation Reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 843–853, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Jan Niehues and Muntsin Kolss. A POS-Based Model for Long-Range Reorderings in SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 206–214, Athens, Greece, March 2009. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *In Proc. of LREC-2006*, pages 2216–2219, 2006.

F. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelhpia, PA, 2002.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, Boston, MA, 2004.

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, 2003.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

Sebastian Padó. *User's guide to `sigf`: Significance testing by approximate randomisation*, 2006.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. Research Report RC22176, IBM Research Division, Thomas J. Watson Research Center, 2001.

Stefan Riezler and John T. Maxwell. On Some Pitfalls in Automatic Evaluation and Significance Testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

C Ruiz, N Paredes, P Macizo, and MT Bajo. Activation of lexical and syntactic target language properties in translation. *Acta Psychologica*, 128(3):490–500, 2008.

Nick Ruiz, Arianna Bisazza, Roldano Cattoni, and Marcello Federico. FBK's Machine Translation Systems for IWSLT 2012's TED Lectures. In *International Workshop on Spoken Language Translation (IWSLT)*, pages 61–68, Hong Kong, 2012.

Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

Holger Schwenk and Jean Senellart. Translation Model Adaptation for an Arabic/French News Translation System by Lightly-Supervised Training. In *Proceedings of the Machine Translation Summit XII*, Ottawa, Canada, 2009.

Andreas Stolcke. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of ICSLP*, Denver, Colorado, 2002.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. Post-ordering in Statistical Machine Translation. In *MT Summit XIII: the Thirteenth Machine Translation Summit*, pages 316–323, Xiamen, China, 2011.

Christoph Tillmann. A Unigram Orientation Model for Statistical Machine Translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

Christoph Tillmann and Hermann Ney. Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation. *Computational Linguistics*, 29(1):97–133, 2003.

Roy Tromble and Jason Eisner. Learning Linear Ordering Problems for Better Translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August 2009. Association for Computational Linguistics.

Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998. ISBN 0471030031.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. A Word Reordering Model for Improved Machine Translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 486–496, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Warren Weaver. Translation. Reprinted in Locke and Booth (1955), 1949.

Dekai Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Fei Xia and Michael McCord. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In *Proceedings of Coling 2004*, pages 508–514, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.

Sirvan Yahyaei and Christof Monz. Dynamic Distortion in a Discriminative Reordering Model for Statistical Machine Translation. In *International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, 2010.

Kenji Yamada. *A syntax-based translation model.* PhD thesis, Department of Computer Science, University of Southern California, Los Angeles, 2002.

R. Zens, F. J. Och, and H. Ney. Phrase-Based Statistical Machine Translation. In *25th German Conference on Artificial Intelligence (KI2002)*, pages 18–32, Aachen, Germany, 2002. Springer Verlag.

Richard Zens and Hermann Ney. A Comparative Study on Reordering Constraints in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 144–151, 2003.

Richard Zens and Hermann Ney. Discriminative Reordering Models for Statistical Machine Translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 55–63, New York City, June 2006. Association for Computational Linguistics.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *Proceedings of Coling 2004*, pages 205–211, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.

Yuqi Zhang, Richard Zens, and Hermann Ney. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8, Rochester, New York, April 2007. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June 2006. Association for Computational Linguistics.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1145–1152, Manchester, UK, August 2008. Coling 2008 Organizing Committee.