Using Linguistic Knowledge in Statistical Machine Translation

by

Rabih M. Zbib

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in the field of Information Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author . . . .
Department of Civil and Environmental Engineering
September 15, 2010

Certified by . . . .
James R. Glass
Principal Research Scientist of the Computer Science and Artificial Intelligence Laboratory
Thesis Supervisor

Certified by . . . .
Steven R. Lerman
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by . . . .
Daniele Veneziano
Chairman, Departmental Committee for Graduate Students

Using Linguistic Knowledge in Statistical Machine Translation

by

Rabih M. Zbib

Submitted to the Department of Civil and Environmental Engineering
on September 15, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the field of Information Technology

Abstract

In this thesis, we present methods for using linguistically motivated information to enhance the performance of statistical machine translation (SMT). One of the advantages of the statistical approach to machine translation is that it is largely language-agnostic. Machine learning models are used to automatically learn translation patterns from data. SMT can, however, be improved by using linguistic knowledge to address specific areas of the translation process, where translations would be hard to learn fully automatically.

We present methods that use linguistic knowledge at various levels to improve statistical machine translation, focusing on Arabic-English translation as a case study. In the first part, morphological information is used to preprocess the Arabic text for Arabic-to-English and English-to-Arabic translation, which reduces the gap in the complexity of the morphology between Arabic and English. The second method addresses the issue of long-distance reordering in translation to account for the difference in the syntax of the two languages. In the third part, we show how additional local context information on the source side is incorporated, which helps reduce lexical ambiguity. Two methods are proposed for using binary decision trees to control the amount of context information introduced. These methods are successfully applied to the use of diacritized Arabic source in Arabic-to-English translation. The final method combines the outputs of an SMT system and a Rule-based MT (RBMT) system, taking advantage of the flexibility of the statistical approach and the rich linguistic knowledge embedded in the rule-based MT system.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist of the Computer Science and Artificial Intelligence Laboratory

Thesis Supervisor: Steven R. Lerman
Title: Professor of Civil and Environmental Engineering

Acknowledgments

I am grateful for the guidance and support of my adviser Jim Glass throughout this process. Four years ago, I came to Jim with little more than a passion for language, and some hazy ideas, and he gratefully accepted me as a member of the Spoken Language Systems group.

Steve Lerman was instrumental for my coming to MIT, and for my coming back the second time, and has been extraordinarily supportive throughout the years. George Washington University will only discover how lucky they are to have him.

I am proud to have Judson Harward as a mentor and a member of my committee. I hope to have many more fascinating discussions in the future.

I am indebted to John Makhoul for giving me the unique opportunity of working with the BBN team, for his continuous support, and for taking the time to be on my dissertation committee, and to read this thesis. There is something to learn everyday watching John's brilliant leadership.

Spyros Matsoukas has taught me how to be intellectually creative, yet experimentally rigorous, two necessary ingredients for successful research in language. Many of the ideas in this thesis are the result of long discussions with him.

I am thankful to all the members of SLS for providing such a stimulating work environment. I am lucky to have had the opportunity to work with the people at BBN: Jacob Devlin, Michael Kayser, Jeff Ma, Tim Ng, Long Nguyen, Antti-Veikko Rosti, Rich Schwartz, Libin Shen, Jinxi Xu and Bing Zhang. Through their brilliance and dedication, they have formed one of the best MT research teams in the world (if you don't believe me, check the GALE results!). Much of the work in this thesis was done within that ecosystem, where everybody is willing to answer any question and help with any task.

The SLS group is a stimulating environment that I have enjoyed working in and learning from.

Section 3.4 is common work with Michael Kayser. Section 3.5 and chapter 4 are common work with Ibrahim Badr.

Throughout the years, I have been lucky to meet many exceptional people at MIT and in the Boston area, too many to list here. I am proud to count many of them as friends.

To my parents, whose children's education was always a priority, through the most difficult years, I owe much.

To my love, my partner and my best friend, for whatever life may bring.

Parts of this research were supported by DARPA/IPTO Contract No. HR0011-06-C-0022 under the GALE program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The idea of automatically translating from one human language to another using computers was proposed surprisingly early in the history of development of computers, as initial attempts at implementing a machine translation (MT) system were made in the 1950s. But it is a problem that has proved considerably harder than it was first believed to be. The quest to develop cheap, reliable, and fluent machine translation continues today, with an ever increasing level of research effort and commercial investment.

The advent of the Internet has been behind the surge of interest in machine translation on both the demand and the supply sides. The proliferation of digital information, and the continued rise of the number of people coming online in all corners of the globe has created a higher need for cheap translation between more and more languages. The availability of digital information in different languages, on the other hand, coupled with the availability of cheaper and more powerful computing hardware has enabled the development of statistical methods for machine translation. Statistical machine translation (SMT) is a general approach that uses machine learning techniques to automatically learn translations from data. Parallel corpora (i.e., text data in one language along with its translation) are used to train statistical translation models, which learn probabilistic translations between linguistic entities (e.g., words or phrases). The translation of unseen data is typically formulated as a search problem, where the probabilistic model is used to search for the best translation of the complete input sentence.

Despite its divergence from traditional linguistic theories, SMT (and more generally, the statistical approach to natural language processing), has many advantages. Translation patterns are learned directly from training data, and they can be generalized to translate new sentences. This approach is also less labor intensive than the alternative of explicitly encoding the knowledge required to handle all possible cases through deterministic rules. Models that are developed can usually be applied directly to new language pairs, as long as training data is available for those languages. These advantages, combined with the enabling factors mentioned above, have made SMT the preferred approach to research and development of machine translation over the past two decades. During this time, large advances have been made in the development of SMT models and tools for many language pairs, and the quality of the translation output has continuously improved. SMT systems that produce reliable and fluent translations exist today for some language pairs, especially in specific data domains.

But when the differences between the two languages are considerable, some of the translation patterns could be difficult, if not impossible to learn. Typically, larger amounts of data are required in order to learn these translation patterns, due to their higher complexity. A completely language-agnostic approach, therefore, would not be the optimal way in which to make use of the available parallel data. Language-specific methods can be integrated within the SMT approach to enhance system performance. This thesis proposes several methods that use linguistic information at different abstraction levels to address specific aspects of the translation problem. The work in this thesis uses Arabic-English translation, in both directions, as a case study into how this language-specific information can be used. The methods and ideas contained in this thesis are, however, applicable to other languages.

Interest in Arabic translation has risen dramatically over the past decade, driven by the rise in the general interest in the Arabic language for security, political and cultural reasons. Almost all the research has been in the Arabic-to-English direction, reflecting the interest of funding sources. This thesis partly deals with English-to-Arabic MT. This direction presents technical challenges that do not exist in the other direction, due to the difference in the properties of the two languages (keep in mind that translation is a highly asymmetric problem).

We next briefly discuss the difficulties involved in the translation problem, then give a high-level description of the different approaches to machine translation.

1.1 Why is Machine Translation Difficult?

Human language is a deceptively complex system that makes use of a finite set of primitive constructs to produce an infinite number of sentences, that express all the different aspects of human activity. This fertility of language is achieved not only through recursive construction, but also through the adaptation of words to express different meanings. The resulting lexical ambiguity is a challenge to the translation task, since the proper translation of a word depends on the meaning corresponding to the specific usage. The syntactic properties of languages also differ considerably, and the difference is usually manifested as differences in the word order of the sentences. These differences have to be handled during the translation process. A sentence in serialized form can correspond to more than one unique syntactic structure. This phenomenon of syntactic ambiguity is quite common in natural language. The correct resolution of syntactic structure is crucial to the proper interpretation of a sentence, and is, therefore important for proper translation. Languages also differ in their morphology. They have, for instance, different pronoun, verb tense, and noun case

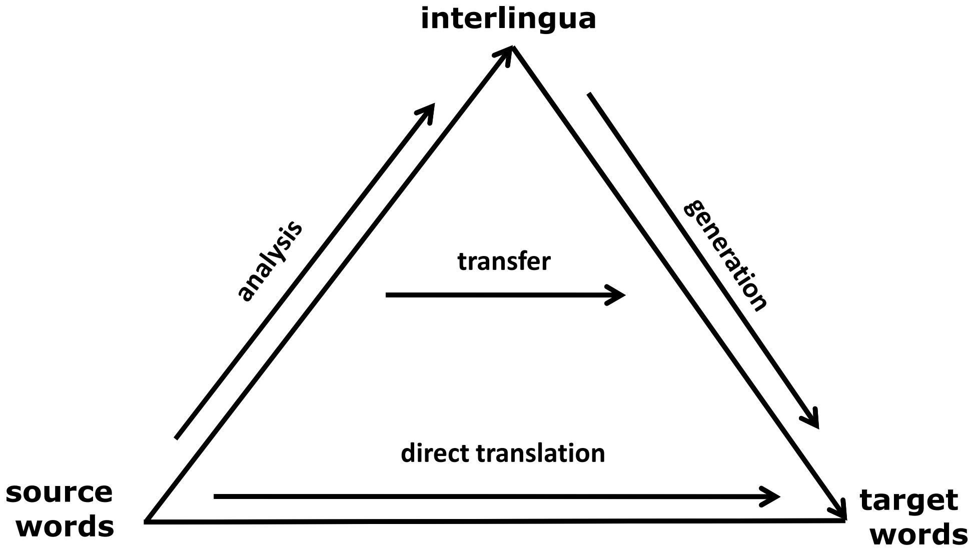


Figure 1-1: The MT Pyramid. Higher points on the pyramid edges correspond to more abstract levels of intermediary representation.

systems. The determination of which set of values of these properties in one language corresponds to which set in the other language in a specific context is a difficult task. All of these challenges assume that proper translation can be performed through word-to-word translation and reordering (an assumption made by early SMT models). But idiomatic and metaphoric use of language is very common. Also, syntactic structures are not always preserved across translation. This implies that literal translation will not give an appropriate result. The consideration of context is crucial for producing correct and fluent translation. These factors and others make machine translation a challenging problem.

1.2 Different Approaches to Machine Translation

The different approaches that have been proposed for machine translation can be broadly categorized into three classes, depending on the level of abstraction of the knowledge representation used in the translation process. Those categories are usually depicted through the machine translation triangle, shown in figure 1-1. The two sides of the pyramid correspond respectively to the analysis of the source language and the generation of the translation in the target language. At the bottom of the triangle are direct translation methods, where translation is performed from string of words to string of words, with very little analysis of the source. The second category of approaches is transfer based, where the source sentence is analyzed and transformed into a rich representation (e.g., syntactic-semantic tree). The structure is then transformed into an equivalent structure in the target language, and that, in turn, is used to generate the target sentence. At the top of the pyramid is interlingua-based translation, where the idea is that the source sentence is analyzed into a universal representation, from which the translation sentence is generated. This would reduce the number of translation systems needed to translate between n languages from $n(n-1)$ to n.

Another dimension along which MT methods differ is deterministic versus probabilistic. This, in principle, is independent of the level of abstraction used, although in practice, direct translation methods tend to be statistical, while methods that use higher abstraction levels tend to be deterministic. Source side analysis using statistical models is a recent trend that will likely continue as statistical models for analysis (e.g., parsing) and translation continue to be developed. We now briefly discuss the properties of each of these approaches.

The interlingua approach to machine translation has not found wide success, mainly due to the difficulty of specifying a universal representation of language that is comprehensive enough for the translation problem in a large domain. (Knight, 1997) illustrates the difficulties with this endeavor nicely through an example. He suggests that the reader consider what is involved in translating a seemingly simple sentence like “John saw the Grand Canyon flying to New York” into Japanese through such a process. The meaning of the sentence has to be computed first, using some precise representation. Then that representation would be used to generate the Japanese translation, using additional grammatical and lexical knowledge. Upon careful inspection, it becomes clear that resolving the ambiguity of the sentence requires knowledge of facts about the world, like "Canyons don't fly". The best that syntactic analysis can do is to produce all the different syntactic structures that could be underlying this sentence. Translation, though, requires that the correct analysis be determined. And even then, a certain level of semantic understating is needed. But a rigid semantic representation will fail to accommodate the understanding of the concept of "people flying" in the metaphoric sense, while rejecting "canyons fly". Then, trying to clearly specify the reasons why a canyon cannot fly, even though a plane, another inanimate object, can, is not an easy task. Neither reducing the reasoning to a few logical principles, nor specifying a value for the attribute CAN_FLY to every object is feasible. This approach seemly requires the solution to the artificial intelligence problem in the strict sense as a prerequisite to solving the machine translation problem.

Using deterministic transfer rules, which was the dominant approach until relatively recently, and continues to be used in many commercial applications, does not require such an ambitious solution. Only the knowledge that is needed to perform the translation has to be directly encoded in the source analysis and transfer rules between the two languages. But this approach in turn suffers from drawbacks. Specifying explicit rules of analysis and transfer that take into account all the cases that might arise, and dealing with all the subtleties mentioned in the previous section is also a daunting task. Another difficulty with the rule-based approach to machine translation is that even if one succeeds in the task of specifying correct and complete rules for translating from one given language to another, a different set of rules would have to be specified for a new language pair. This approach suffers from scalability issues both in terms of translating data from a large domain, and generalizing to new languages.

Statistical machine translation uses statistical models to learn translations automatically, usually directly between words, or using syntactic knowledge on the source and/or target sides. Parameters of the statistical model are learned from parallel data using machine learning techniques. The SMT paradigm has many advantages, which has made it the subject of the majority of the research in the field of machine translation over the past couple of decades. The most obvious advantage is that models and techniques are independent of a specific language. Software resources are also, to a large extent, portable across languages. The other advantage is that the models can learn to automatically translate between a large number of linguistic patterns, as long as these patterns occur in the training data. No explicit encoding of the knowledge is needed to handle the different cases. Another, more subtle advantage, is that SMT systems avoid making hard decisions at any of the intermediate stages of the translation process. Decisions are deferred to the end of the process, and only probability scores are carried through. This prevents intermediate mistakes from dooming the final translation result. Chapter 2 contains a review of the history and current research in SMT.

1.3 The GALE Project

All the work in this thesis, except for the parts on English-to-Arabic in chapters 3 and 4, was conducted as part of DARPA's GALE (Global Autonomous Language Exploitation) program (Olive, 2005).

GALE is a 5-year program whose goal is to develop technologies to eliminate the need for linguists and analysts in the translation and analysis of large amounts of information in foreign languages. The technologies developed under GALE include robust automatic recognition of continuous speech, including both read speech and spontaneous speech. Most relevant to this thesis, the DARPA program aims to develop machine translation technology to translate automatically transcribed speech, as well as text into English. Relevant text genres include news broadcast, newswire, and the less structured weblog data. Finally, GALE includes the development of information distillation technology to provide automatic question answering functionality based on information extracted from raw text, both original and automatically translated.

The GALE program mainly focuses on two languages: Arabic and Mandarin Chinese.

1.4 Contributions of this Thesis

This thesis offers the following specific contributions:

- A method for using morphological information for English-to-Arabic SMT.
- A method for using syntactic information for English-to-Arabic SMT.
- A model for integrating explicit contextual information into the SMT framework.
- Two methods for successfully using diacritized Arabic source.
- A method for the combination of an SMT system and a rule-based MT system.
- Suggestions on research directions for other methods of combining an SMT and a rule-based MT systems.

1.5 Summary of Each Chapter

The rest of this thesis is organized into 6 chapters. The content of each of the remaining chapters is summarized next.

- **Chapter 2: Background**

Chapter 2 starts with a review of the major advances in statistical machine translation over the past two decades. It then presents a brief description of different aspects of the Arabic language with the purpose of providing the reader with a basic insight into the language, which would facilitate understanding the work in the rest of the thesis.

- **Chapter 3: Morphological Preprocessing for SMT**

  This chapter discusses the effect on machine translation of the difference of the morphology of the source and target languages. It presents experiments on using morphology-based splitting of the Arabic source for Arabic-to-English MT, comparing the use of a rule-based and a statistical morphological analyzer on MT performance.

  The second part of chapter 3 describes how the morphological splitting can be used in the less explored direction of English-to-Arabic SMT. It discusses why recombining the segmented Arabic output of the MT system is not a trivial task, and suggests methods to perform this combination.

- **Chapter 4: Syntax-based Reordering for SMT**

  Chapter 4 presents another preprocessing method, which in this case is targeted at dealing with the difference in the syntax of Arabic and English. Reordering rules defined on the English source for English-to-Arabic SMT are suggested, and experimental results showing the advantage of using these rules are presented.

- **Chapter 5: Source Context using Binary Decision Trees**

  Chapter 5 tackles the problem of lexical ambiguity, and its effect on machine translation. It shows how additional context information can be used with beneficial results, by controlling the amount of context information through the use of binary decision trees.

- **Chapter 6: System Combination of Statistical MT and Rule-based MT**

This chapter discusses in more detail the respective advantages of the rule based and statistical approaches to machine translation. It then introduces a new method for combining the output of two systems from the two different paradigms, taking advantage of the benefits of each.

- **Chapter 7: Conclusion and Future Work**

This chapter concludes the thesis, and suggests some ideas for future research.

Figure 1-2 shows the structure of the thesis, with the type of linguistic information used in each chapter.

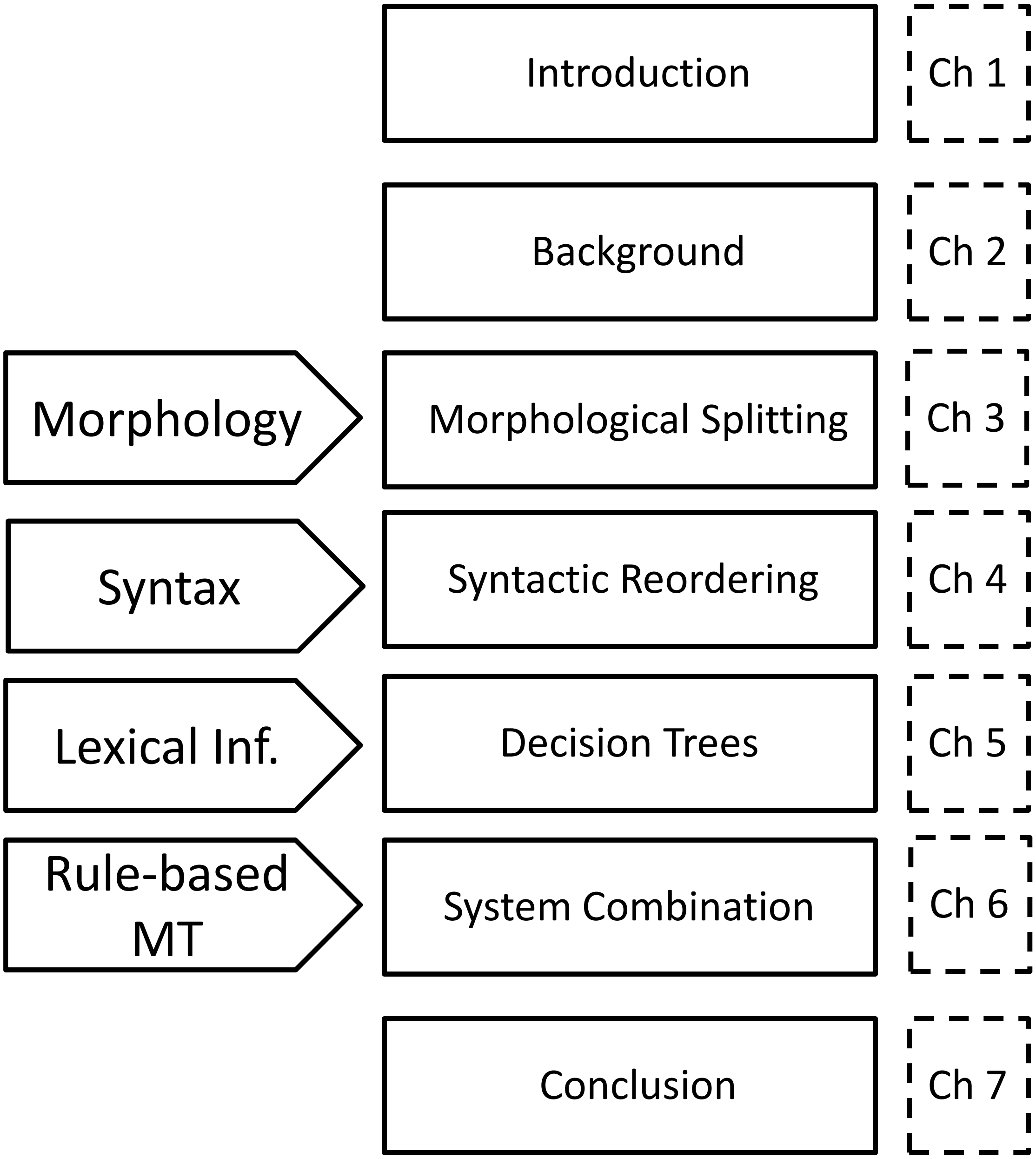


Figure 1-2: Thesis structure, showing the kind of language-dependent information used in each chapter.

Chapter 2

Background

This chapter reviews the major advances in statistical machine translation through the past two decades. Over that period, SMT has become the preferred approach for research and development of machine translation systems. New models and techniques continue to be proposed with an ever increasing pace. The first section of this chapter reviews some of the landmark advances in that area. We then discuss the problem of automatic evaluation of machine translation quality, and describe a few of the most popular evaluation methods. The last part of the chapter contains an overview of different aspects of the Arabic language: the orthography, morphology and syntax of Arabic, with the aim of providing the non-Arabic reader with enough linguistic background to discern the work in this thesis.

2.1 Previous Work on Statistical Machine Translation

This section briefly reviews the general development of statistical machine translation over the past two decades. The most significant developments in the field are mentioned here, while the previous work that is more relevant to the material in each chapter is reviewed in that chapter.

The prevalent paradigm to machine translation until the 1990's was through deterministic transfer rules. In the past two decades, the research on statistical methods for machine translation has advanced in strides, and statistical methods have also started to be adopted in commercial MT systems.

Statistical machine translation has become the prevalent approach to MT over the last two decades, especially among the reseach community. This growth has been driven by the advantages of the statistical approach mentioned in chapters 1 and 6. Two additional factors have contibuted significantly to this development. The first is the dramatic increase in the computational power and storage capacity and the decrease in cost of hardware, which permitted for increasingly computation-intensive methods to be developed using larger amounts of data. The second factor is the advent of the Internet, and of digital content in general, which has made multi-language data resources available to be used in developing SMT systems, and on the other hand has increased the demand for fast, cheap and reliable translation. The scaling of data corpora is key for the success of the complex translation models. Multi-word models like phrase-based MT and syntax-based models require large amounts of training data to overcome issues of sparsity, which result in unreliable parameter estimates and consequently, unreliable translation results. With enough training data, and for limited domains, the quality of some SMT systems is quite reliable, especially for language pairs that have been extensively researched and developed. But the problem of machine translation, in its general formulation, is still far from solved. (Church and Hovy, 1993) suggests that “crummy” machine translation output can still be useful with the right user expectations, and in the right application context.

The remainder of this section is a brief review of the development of statistical machine translation. (Lopez, 2008) is a recent comprehensive survey of the field. (Dorr et al., 1999) is an older survey. In book form, both (Manning and Schütze, 1999) and (Jurafsky and Martin, 2000) touch on the subject of machine translation. The recently published (Koehn, 2010) is the first book dedicated to SMT.

2.1.1 The Noisy Channel Model

An important property of any specific SMT method is its modeling unit. Early SMT models were based on modeling word-to-word translations. The noisy channel model, first proposed in (Brown et al., 1990), was the basis for a sequence of increasingly complex word translation models known as the IBM models (Brown et al., 1993), which have had a large influence on the development of the field. The noisy channel model assumes that the source (foreign or French) sentence f is a distorted version of the target (or English sentence) e[1]. The translation process consists in the recovery of the most probable target sentence given the source sentence. In principle, a source sentence f can be translated into any target sentence, an assumption that is shared by all SMT methods. One translation is chosen over another because it has a higher probability score, and the most probable target sentence under the model is chosen as the translation output. Bayes rule is used to rewrite the probability maximization criterion:

$$\begin{aligned} \hat{e} &= \underset{e}{\operatorname{argmax}} \Pr(e|f) \\ &= \underset{e}{\operatorname{argmax}} \frac{\Pr(f|e)\Pr(e)}{\Pr(f)} \\ &= \underset{e}{\operatorname{argmax}} \Pr(f|e)\Pr(e) \end{aligned} \tag{2.1}$$

The term $\Pr(f)$ can be eliminated from the maximization term since it does not affect the choice of e. This formulation would be familiar to readers with knowledge of speech recognition. The term $\Pr(f|e)$ is called the *translation model,* and $\Pr(e)$ is the a priori *language model* probability. It might seem more obvious and straightforward to directly search for the most probable target sentence, but the advantage of the decomposition of equation 2.1 is that the estimation of $\Pr(e|f)$ requires that every possible target sentence be assigned a non-zero probability, while most of the strings

[1] Consistent with the general practice in the literature, we use f for "foreign", the source language, and e for "English", the target language. This notation was used in the original IBM papers on word-based SMT (Brown et al., 1993)

composed of the vocabulary of the target language are non-grammatical. Estimating $\Pr(f|e)$ instead requires that only the probability of grammatical target sentences be estimated. Also, the explicit incorporation of the language model term allows non-grammatical sentences to be penalized during the search.

2.1.2 Language model

The language model has to capture information about the allowable sequences of words in the target language, as well as the frequency of occurrence of these sequences. This information guides the search process during decoding to favor more common sentences. The language model has to strike a balance between the expressiveness of the model and its flexibility. A model that defines grammaticality too tightly can prove to be limited and brittle. The most common type of language models define the probability of a given word in terms of its preceding words in the sentence. Thus, the probability of a sentence $e = e_1, \ldots, e_K$ is:

$$\Pr(e) = \prod_{i=1}^{K} \Pr(e_i|e_1, ..., e_{i-1})$$

An *n-gram* language model limits the history of a certain word to its preceding $n-1$ words, by making the following independence assumption:

$$\Pr(e_i|e_1, ..., e_{i-1}) = \Pr(e_i|e_{i-n}, ..., e_{i-1})$$

This assumption reduces problems of data sparsity, since the probabilities are estimated from counts of word string occurrences in a data corpus. *n-gram* models are widely used in Speech Recognition, Natural Language Generation, and other Natural Language Processing applications. Despite their simplicity, *n-gram* language models have proved to be surprisingly effective (Manning and Schütze, 1999; Jelinek, 1997; Rosenfeld, 2000). Until recently, language models that use syntactic and other complex information usually provide little advantage for the complexity they introduce

(Filimonov and Harper, 2009). Techniques have been proposed to smooth the probability estimates of sparse language models, and to estimate probabilities of unseen *n-grams* (Chen et al., 1998; Katz, 1987).

The explicit use of a language model as a separate factor in equation 2.1 has another advantage. The language model is estimated using a monolingual corpus, which can be obtained cheaply, as opposed to the bilingual corpus needed to train a translation model. The language model can be trained using very large amounts of data, without being restrained by the relatively smaller size of the bilingual corpora typically available.

2.1.3 Word Alignment Models

We describe the IBM word alignment models (Brown et al., 1993) in some detail next. Although these models were proposed some 15 years ago, they still form the basis for some SMT models of the current state-of-the-art. These consist of five models for word-to-word alignment, called IBM Models 1-5, each with an increasing complexity over the preceding model. They are all estimated from training data that consists of bilingual sentence pairs. The estimation procedure assumes that an alignment exists between the words of the two sentences, but that the alignment is unknown. Any word in the source sentence f can in principle be aligned to any word in the target sentence e. If the alignments were known, the word translation probabilities can be estimated accordingly:

$$\Pr(f|e) = \sum_{a \in A} \Pr(f, a|e) \tag{2.2}$$

where A is the set of alignments. If the translation probabilities were known, on the other hand, probabilistic (or partial) word alignments could be determined. The Expectation Maximization or EM algorithm (Dempster et al., 1977) is used to estimate both the alignments and translation probabilities iteratively.

The five models are of increasing complexity, as they account for more translation phenomena, such as many-to-one translation and reordering. The generative story[2] of Model 5 is:

1. Each target word e_i picks the number of source words that it will generate, ϕ_i. This is called the *fertility* of e_i. The target sentence is extended with a special *null* word, allowing source words to be translated to the empty string.

2. Each of the ϕ_i copies of the target word e_i is translated into a source word.

3. The source words are reordered according to a distortion model that depends on the lexical value and position of the target word.

This account means that the IBM alignment models are asymmetric. They allow many-to-one alignments from the source to the target, but not in the other direction. Symmetrization methods will be described in the section on phrase-based SMT. Also, word-based models translate word for word, without taking into account the surrounding source words, which often results in a *word salad,* that is, incoherent word sequences. They do not consider the fact that certain word sequences constitute phrases that should be translated together. Phrase-based SMT (section 2.1.4) addresses this issue as well. Finally, despite the inclusion of a distortion model that theoretically permits arbitrary reordering, in practice reordering is often limited to a distance of a few words to keep the models tractable, meaning that these models cannot handle long distance reorderings that the difference in the syntax of the two languages necessitates. Hierarchical and syntax-based MT attempt to address this problem by allowing word chunks to be moved jointly.

GIZA++ (Och and Ney, 2003) is a widely used implementation of the IBM model 4.

[2] A generative story is a hypothetical account of how certain data is generated. It forms the basis for determining the mathematical models that model that data.

Word alignment models besides the IBM models have been proposed, including HMM-based word alignment (Och and Ney, 2000; Lopez and Resnik, 2005; DeNero and Klein, 2007).

2.1.4 Phrase-based SMT

Phrase-based machine translation aims to improve over word-based MT by using a chunk of words, or a phrase, as the modeling unit, instead of a single word. This allows word sequences that occur together frequently to be translated jointly, avoiding word-for-word translations, which might be inadequate or wrong.

Alignment templates (Och et al., 1999) constituted a transitional step from word-based alignment models. These templates are generalized phrase pairs, consisting of word classes with internal alignments. (Och, 1999) suggests how word classes on the source and target side can be estimated from monolingual and bilingual data.

Phrase-based models (Zens et al., 2002; Marcu and Wong, 2002; Koehn et al., 2003) translate between phrases consisting of words instead of word classes. Recall that word-based alignments are asymmetric. Many-to-one alignments are only possible from the source to the target. But many-to-many alignments are needed to extract phrase pairs. Alignments are symmetrized by, for instance, running a word-based alignment procedure such as GIZA++ in both directions, and combining them by intersection, union or other heuristics. Internal alignments between words in the phrase pair are usually ignored. The phrases used in these models are not linguistically based (i.e., syntactic constituents). The phrase probabilities are estimated from phrase counts:

$$\Pr(\widetilde{f}_k|\widetilde{e}_k) = \frac{N(\widetilde{f}_k, \widetilde{e}_k)}{N(\widetilde{e}_k)} \tag{2.3}$$

The decoding consists in splitting the source sentence into phrases, translating each phrase, and then reordering the phrases. A beam-search based decoder is used to prune the search space, making the search sub-optimal, yet tractable. Ordering models that are a function of the lexical values of the phrase pairs are sometimes used. Phrase pairs successfully model local reordering within the phrase, but they are inadequate at modeling long-distance reordering, since performing arbitrary reorderings without a recursive mechanism is computationally prohibitive.

The increase in the size of available training data is a key factor in making phrase-based models usable, since large amounts of data are required for the reliable estimation of phrase pair probabilities. The widely used Pharaoh decoder (Koehn, 2004a) and the more recent open source toolkit MOSES (Koehn et al., 2007) are based on the phrase-based model in (Koehn et al., 2003).

2.1.4.1 Maximum Entropy Model

Log-linear models correspond to the maximum entropy solution for parameter estimation (Berger et al., 1996). They have been used widely in NLP applications. (Och and Ney, 2002) model the forward translation probability directly as a log-linear combination of weighted model components, corresponding to observed features as such:

$$\hat{e} = \underset{e}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e,f) \right\} \tag{2.4}$$

where $h_m(e,f)$ is the value of feature m, a function of both the source and target sentences; λ_m is the weight for feature m. The weights $(\lambda_1, \ldots, \lambda_m)$ are estimated by optimizing an objective function (typically a translation metric) using a tuning data set.

The advantage of the log-linear model over the earlier noisy channel model is that multiple model components can be added, rather than restricting the model to the translation model and language model components. Log-linear models are considered discriminative in that they directly model the target translation conditioned on the observed feature values. Features typically used in phrase-based systems include:

- Phrase probabilities, in both directions; that is $\Pr(\widetilde{f}_k|\widetilde{e}_k)$ and $\Pr(\widetilde{e}_k|\widetilde{f}_k)$.
- An *n-gram* target language model.
- Lexical probabilities; that is probabilities of translation of single word pairs, also in both directions. These probabilities work as a smoothing for the phrase pair probabilities, which are estimated from sparser counts. Chapter 5 describes the lexical smoothing feature in more detail.
- A hypothesis length penalty that allows for the length of the hypotheses to be controlled.

(Och et al., 2004) reports on the use of a wide variety of features, some of which were found to be useful, and others were not.

2.1.4.2 Parameter Tuning

The values of the feature components are estimated from (possibly different) data corpora. They are combined through a weighted interpolation, as equation 2.4 shows. The weights determine the relative effect of each feature on the overall score. As mentioned above, these are determined by optimizing some objective function over an unseen data set. This process is called parameter tuning. The optimization function is typically a (combination of) translation metric(s), that is optimized iteratively by searching through the weight space (Och, 2003). A non-gradient descent method, such as Powell's method (Brent, 1973) is used when the objective function is not differentiable. A smoothed objective function can be used instead, which can be optimized using gradient descent methods.

2.1.5 Hierarchical SMT

As (Lopez, 2008) notes, the models described so far correspond to finite-state transducers, and can be equivalently described as regular languages (Sipser, 2005). Natural language, however, is better modeled as a context-free grammar (CFG). Modeling arbitrary reorderings using regular finite-state transducers is an awkward and expensive operation. CFGs by contrast are suitable to model the recursive structure of natural language syntax. Synchronous context-free grammars (SCFG) (Aho and Ullman, 1969) are an extension of CFGs to bilingual rules of the form

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle$$

where γ is a sequence of terminals (i.e., lexical items) and non-terminals in the source language, α is the corresponding sequence of terminals and non-terminals in the target language, and $\sim$ is a one-to-one alignment between the non-terminals of γ and α. SCFGs are suitable for modeling translations between structures (as opposed to strings). The aligned non-terminals in the two right-hand sides of the bilingual rule are considered corresponding translations, and they can be recursively expanded through the application of further rules. The reordering of word chunks can be easily modeled through the reordering of aligned non-terminals in the parallel rule.

Bracketing grammars (Wu, 1996; Wu, 1997) are an early attempt at efficiently modeling translation reorderings on top of word alignments. They use a single non-terminal and a simple grammar of three rules to represent binary bracketing reorderings:

$$\begin{aligned} X &\rightarrow X_{\boxed{1}} X_{\boxed{2}} / X_{\boxed{1}} X_{\boxed{2}} \\ X &\rightarrow X_{\boxed{1}} X_{\boxed{2}} / X_{\boxed{2}} X_{\boxed{1}} \\ X &\rightarrow e/f \end{aligned} \tag{2.5}$$

Hierarchical phrase-based MT (Chiang, 2005; Chiang, 2007) combines the advantages of both SCFGs and phrase-based MT. It allows up to two non-terminal variables on the right-hand side of the rule, providing a more powerful reordering mechanism than binary bracketing grammars. The similarity with phrase-based translation is that the rules can also contain non-terminals. Hierarchical rules can be looked at as generalizations of phrase pairs where the phrases are recursively substituted instead of being concatenated. The rules are extracted from phrase pairs by substituting a non-terminal variable in place of a phrase pair that occurs within it. This grammar is thus not syntactically constrained, in that the substituted phrases do not correspond to linguistic constituents. Rule probabilities are estimated from joint and marginal counts of the source and target sequences, similar to how phrase translations are estimated.

The decoding consists in parsing the source sentence and generating the source side simultaneously. The parsing can be done in $O(n^3)$ time using dynamic programming, and *N-best* translations can be generated efficiently (Klein and Manning, 2001; Huang and Chiang, 2005).

A log-linear model with many features, similar to section 2.1.4.1 is usually used with Hierarchical SMT.

2.1.5.1 The Hierdec Decoder

The baseline systems described in section 3.4 and in chapters 5 and 6 are based on the Hierdec decoder (Shen et al., 2008), a hierarchical decoder similar to (Chiang, 2005; Chiang, 2007). In addition to the *n-gram* language model, Hierdec uses a dependency tree on the target side that is extracted from target parses.

An N-best list of translations is usually produced using a 3-gram language model for efficient decoding, and is then rescored with a 5-gram language model and other features.

(Shen et al., 2009) explores the use of additional features within the Hierdec framework. Those include part-of-speech tags on the target side and terminal length distributions.

2.1.6 Syntax-based Statistical Machine Translation

A wide range of approaches can be categorized under the label of syntax-based SMT. They share the common factor of incorporating knowledge based on natural language syntax into the translation process. The syntactic information is typically derived from statistical parsers (Collins, 1997; Collins, 1999; Charniak, 2000), which implies that the successful use of syntax in MT depends on the availability of reliable parsing. Sometimes reliable parses are available for only one of the two languages, forming the motivation for using syntax on one side only. This side is usually the English side, since the most mature and reliable statistical parsers are for English.

One approach is to use syntactic information at the source side, as a separate step, where the source sentence is reordered according to a set of rules defined on the parse tree. (Collins et al., 2005; Nießen and Ney, 2004; Wang et al., 2007; Xia and McCord, 2004) are examples of such an approach. The work in chapter 4 falls under this category. Others, such as (Huang et al., 2006), integrate the parse information more tightly by using a tree-to-string model, where the parse tree of the input is converted into a tree in the target language. The system in (Huang et al., 2006) is implemented on English-to-Chinese translation.

The more common approach is to use syntax on the target side, which is driven by the emphasis of many research programs on the translation into English, combined with the availability of reliable English parsers. (Yamada and Knight, 2001) generalize the noisy-channel framework to model tree-to-string probabilities. The trees are subjected to reordering, insertion and translation operations. A variant of the EM algorithm is used to train the model.

(Och et al., 2004) incorporate syntactic information by using syntax-based features of the target language in the re-ranking of the N-best translation list. (Gildea, 2003) proposes a more general model for tree transformation, such as node cloning and deletion. Gildea tests the models on Korean-to-English translation. (Lin, 2004) presents another string-to-tree model which maps from source strings to target dependency trees. (Marcu et al., 2006) augment the target side of the phrase pairs of phrase-based SMT. They define multiple variants of this class of models.

Most research on syntax-based MT recognizes the inadequacy of modeling full tree-to-tree isomorphism for machine translation. (Gildea, 2003) notes that syntactic divergences between translation pairs are quite common, either because of genuine differences in the syntax of the two languages, or because the translations used in training are not as parallel as they can be. Tree-to-tree models, therefore, usually concentrate on modeling non-isomorphic mappings. (Eisner, 2003) uses a variant of the EM algorithm to derive the best syntactic mapping between two trees. (Cowan et al., 2006) learns mappings from the source parse tree to a tree-like structure, called *Aligned Extended Projection,* inspired by the Tree Adjoining Grammar (TAG) mechanism.

Finally, we note that, despite the general recognition of the importance of syntactic structure for the translation process, fully incorporating syntactic information into the SMT framework has not yet delivered convincingly significant improvements. Syntax-based SMT remains a highly active field of research.

2.2 Translation Evaluation

Translation evaluation is complicated by the fact that there is no single correct answer for the translation task. A sentence can be translated in many valid, yet different ways. Variations in choice of words, word order and style can occur between different valid translations of the same sentence.

The most reliable method for evaluating translation adequacy and fluency is through human evaluation. But human evaluation is a slow and expensive process. And even human evaluation is prone to inconsistent subjectivity, especially when ranking the quality of multiple translations. The judgments of more than one human evaluator are usually averaged for this reason. To compare the results of more than two systems, human judges have to either perform 2-way comparisons between the different translations, or assign a numerical score to each translation, which adds more subjectivity to the process.

A quick, cheap and consistent method is needed to judge the effects of incremental improvements made to MT systems during day-to-day development. A precise automated evaluation method would require linguistic understanding, a problem that is, arguably, at least as difficult as machine translation. Methods for automatic evaluation usually rely on a measure of superficial similarity between the translation output and one or more translation references. Correlation studies between automatic evaluation methods (usually called translation metrics), and human judgments on translation quality are used to evaluate the reliability of these methods.

Three automatic translation metrics used to report results in this thesis are described next: BLEU, TER and METEOR.

2.2.1 BLEU

The BLEU score (Papineni et al., 2002) is one of the first automatic evaluation metrics to show a high level of correlation with human judgments, and it remains one of the most widely used. The metric is based on n-gram precision, that is, the fraction of the n-grams in the MT output hypothesis that are also found in the reference. Multiple reference translations can be used to compute BLEU. Evaluating translation hypotheses against multiple references provides a more robust assessment of the translation quality.

Using the precision measure directly is problematic, because it rewards superfluously generated n-grams. In the following example from (Papineni et al., 2002), the unigram precision is 7/7:

Example 1.

Hypothesis: the the the the the the the.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

The BLEU score deals with this issue by using a *modified precision* score, where the count of matching n-grams is capped at the maximum number of occurrences of that unigram in the same reference. In the previous example, the modified unigram precision is 2/7.

The use of precision alone means that artificially short hypotheses will get a higher score. Usually, this problem would be handled by the use of recall, that is, the fraction of the reference n-grams that are produced in the MT hypothesis. However, since more than one reference translation are used to compute the score, a bad hypothesis that recalls n-grams from different references can get a high recall score, as the following example, also from (Papineni et al., 2002), shows:

Example 2.

Hypothesis 1: I always invariably perpetually do.

Hypothesis 2: I always do.

Reference 1: I always do.

Reference 2: I invariably do.

Reference 3: I perpetually do.

The first hypothesis recalls more n-grams from the references than the second, even though it is a worse translation.

The BLEU score deals with the problem of short sentences by including a *brevity penalty:*

$$BP = \begin{cases} 1 & \text{If } c > r \\ e^{(1-r/c)} & \text{If } c \leq r \end{cases} \tag{2.6}$$

where c is the total length of the translation corpus and r is the total length of the reference corpus.

The BLEU score is computed as the geometric mean of modified n-gram precision counts, up to an n-gram order of N.

$$BLEU = BC \times \left(\sum_{n=1}^{N} w_n \log p_n \right) \tag{2.7}$$

Typically, a value of $N = 4$ and uniform weight $w_n = \frac{1}{4}$ are used, resulting in the following equation for the computation of the score:

$$BLEU = BP \times (p_1 p_2 p_3 p_4)^{1/4} \tag{2.8}$$

A higher BLEU score indicates a better translation result.

2.2.2 METEOR

METEOR (Banerjee and Lavie, 2005) is another metric for evaluating machine translation systems, also based on word matching. METEOR considers unigram precision and recall. It uses recall directly, unlike BLEU. METEOR does not use higher-order n-grams, but measures translation grammaticality instead by penalizing hypotheses based on how many chunks in the hypothesis would have to be reordered to match the reference(s). Matches between morphological variants of a word (e.g. *computer* vs. *computers*) and between word synonyms are also allowed.

A higher METEOR score generally indicates a better translation result.

2.2.3 Translation Edit Rate (TER)

Another translation evaluation metric is Translated Edit Rate (TER) (Snover et al., 2006). TER measures the number of edits needed to transform the translation hypothesis into the reference translations. The number of edits is normalized by the number of hypothesis words. TER can also use multiple translations, by considering the number of edits with the closest references, and normalizing with the average number of word references. TER is thus defined as:

$$\text{TER} = \frac{\text{\# of edits to closest reference}}{\text{average \# of reference words}}$$

The edits that TER considers are insertion, deletion and substitution of individual words, as well as shifts of contiguous words. TER has also been shown to correlate well with human judgment.

TER is a edit distance measure, so a lower TER score signifies a better translation

2.2.4 Human Translation Edit Rate (HTER)

The GALE project uses a translation distance metric similar to TER called HTER (Human-targeted Translation Edit Rate) (Snover et al., 2006) as the official metric of evaluation. To measure the HTER of the output of a translation system, human annotators perform the minimal edits required to correct the output. The edit distance between the original output and the corrected output is measured. Determining HTER is an expensive exercise, which in the GALE project is only performed during the official evaluation at the end of each phase.

2.2.5 A Note on Translation Scores

The possibility of translating a given text in many valid ways means that a correct translation would still be expected to differ from the reference translations used to score it. This means that there is a floor to the TER score (or ceiling in the case of BLEU) below which even correct translations are unlikely to be. This should be taken into consideration when the quality of an MT system is assessed based on the absolute values of the MT scores.

2.2.6 True Casing

The Arabic-to-English and English-to-Arabic systems in this thesis are trained with lower-case English. The English output of Arabic-to-English is therefore in lower case, and the MT scores for the Arabic-to-English results is presented for the lower case

		TER	**BLEU**
Test1.newswire	Lower	39.54	52.77
	Mixed	41.78	50.58
Test2.newswire	Lower	41.31	50.64
	Mixed	43.52	48.65
Tune.newswire	Lower	39.29	54.22
	Mixed	41.52	52.03

Table 2.1: Difference in scores between lower case English and mixed case English

output. The output can be converted to true case using a statistical *n-gram* based caser. Table 2.1 shows the effect of casing on the scores of a typical Arabic-to-English experiment. The scores usually deteriorate by around two points. The reporting of the lower case scores does not affect the evaluation of the effect of experiments, since that effect is judged relative to a baseline, rather than in absolute terms.

2.3 The Arabic Language

The work in this thesis concentrates on Arabic-to-English and English-to-Arabic machine translation, as a case study into how linguistically motivated techniques can be used to improve the largely language-independent statistical machine translation approach. This section presents a brief introduction to the properties of the Arabic language, with the goal of presenting enough information to allow the non-Arabic speaking reader to get an insight into how the specific linguistic characteristics of Arabic are exploited.

2.3.1 The Arabic Language Family

Typologically, Arabic is a Semitic Language sharing a common origin and high level grammatical features with Hebrew, Aramaic and some of the languages of Ethiopia, among others. It is the official language of 26 countries, and is spoken by 250 million people, making it the 5th most popular language in the world.

The state of spoken Arabic today is better described as a family of languages/dialects rather than a single language, where different varieties have emerged over a vast geographic area, influenced by language substrata and contact with surrounding languages. The spoken varieties have considerable differences among each other, and some of them are not mutually intelligible. They form a dialect continuum, but can be classified into four main groups: North African, Egyptian, Levantine and Gulf/Iraqi.

2.3.2 Modern Standard Arabic

None of the spoken dialects of Arabic has a standardized written form. Instead, the Arabic speaking world uses a written variety of the language called Modern Standard Arabic (الفُصحَى, *AlfuSHA*). Based on the Classical Arabic of the Qur'an, this literary form of Arabic is shared throughout the Arabic speaking region, with little local variation. It is almost always used for writing, to the exclusion of the spoken dialects, and is also spoken in formal situations, such as religious ceremonies, political speeches or news broadcast. It is not, however, spoken natively, in daily life in any region of the Arabic speaking world. It is formally taught to children in school. This, together with the fact that the differences between Modern Standard Arabic (MSA) and the spoken dialects can be considerable, creates an interesting case of Diglossia ((Ferguson, 1959) reproduced in (Belnap and Haeri, 1997)), a sociolinguistic phenomenon where the literary standard differs considerably from the vernacular varieties. Far from being a purely linguistic phenomenon, the diglossic situation of Arabic is one that is laden with social, political and religious issues (Haeri, 2000).

The differences between MSA and the various dialects are large enough to render the translation of dialect input using MT systems trained on MSA data non-usable. The vast majority of linguistic data resources are in MSA; and this thesis, like almost all other research on Arabic MT, focuses on translation to and from MSA. Chapter 7 briefly discusses some of the issues with data gathering and translation of Arabic dialects. The remainder of this section will describe the orthography, morphology and syntax of MSA.

2.3.3 Arabic Orthography and Encoding

The Arabic script is an alphabet consisting of 28 basic letters and a few additional variants. It is written from right to left. Letter shapes change depending on whether they are connected to the adjacent letters. Arabic has three vowel (a, u, i). Both vowel and consonant duration are phonemic. Short vowels and consonant doubling are written using small marks, called diacritics, placed above or below the preceding consonant[3]. Diacritics are usually omitted from Arabic text, except in special cases, such as religious texts, children's books and language learning material. This creates an ambiguity of the pronunciation, as well as lexical ambiguity since two different words can have the same non-diacritized spelling. Readers of Arabic rely on context and on their knowledge of the lexicon to resolve this ambiguity. Chapter 5 discusses the implications of the resulting lexical ambiguity on SMT, and proposes methods for using diacritized source for translation. When diacritics are supplied, Arabic orthography is mostly phonemic.

Several encoding systems for representing Arabic in digital form exist. Those include Apple's MacArabic, and Microsoft's Windows-1256. Modern applications tend to use the Arabic Unicode system, usually encoded using UTF-8. It is worth noting that software applications usually distinguish between representation for storage,

[3] Such writing systems are referred to as *Abjads* (See Daniels, Peter T., et al. eds. The World's Writing Systems Oxford. (1996), p.4.)

where the isolated form of the characters is stored, and rendering, where a different shape of the character might be used, depending on whether it is connected or not.

Several systems for transliterating Arabic in the Roman alphabet also exist. The advantage of transliteration is that no special resources are needed, and it avoids problems of display. The Buckwalter Arabic Transliteration is commonly used in NLP. It is a one-to-one representation of Arabic characters that uses only ASCII. Its advantage is that it can be used by machines and also easily learned by humans. It can be also converted to and from proper Arabic encodings, provided that the source is monolingual. Table 2.2 lists the Buckwalter transliteration of the alphabet, and the International Phonetic Alphabet (IPA) corresponding to each chatacter. The examples in this thesis are written using the Buckwalter system.

2.3.4 Arabic Morphology

Morphology is the branch of linguistics that studies and describes the structure of morphemes, or meaning-carrying units, like words and other particles. Two types of morphology are usually distinguished: derivational morphology and inflectional morphology. Derivational morphology is the process of deriving a new word by combining different morphemes. The addition of the *-er* affix in English, for instance, produces a noun from a verb stem (e.g., *teach+er→teacher, write+er →writer*). The second type of morphology is inflectional, where the form of the word is changed to indicate its grammatical features, without changing the meaning of the word or its part of speech. An example would be the formation of the past verb in English (e.g., *teach→taught, write→wrote*).

Morphology is relevant to machine translation, since words are the basic unit of input to the translation process. But languages differ surprisingly widely in what they consider a word to be. Analytic (or isolating) languages, like Chinese and other East Asian languages, typically use independent words to represent each morpheme. Function morphemes such as tense, gender or number are either unmarked or repre-

Arabic	Buckwalter	IPA	Arabic	Buckwalter	IPA
ء	’	ʔ	ض	D	dˤ
آ	|	aː	ط	T	tˤ
أ	>	ʔ	ظ	Z	zˤ
ؤ	&	ʔ	ع	E	ʕ
إ	<	ʔ	غ	g	ɣ
ئ	}	ʔ	ف	f	f
ا	A	aː	ق	q	q
ب	b	b	ك	k	k
ة	p	h/t	ل	l	l
ت	t	t	م	m	m
ث	v	θ	ن	n	n
ج	j	ʒ	ه	h	h
ح	H	ħ	و	w	w
خ	x	x	ي	y	j
د	d	d	ـً	F	an
ذ	*	θ	ـٌ	N	un
ر	r	r	ـٍ	K	in
ز	z	z	ـَ	a	a
س	s	s	ـُ	u	u
ش	$	ʃ	ـِ	i	i
ص	S	sˤ	ـْ	o	-

Table 2.2: The Buckwalter Transliteration System.

sented as separate words. The other end of the scale are what is called polysynthetic languages, which can pack in a single word the equivalent of a long English sentence. Many Native North American languages are polysynthetic. In the middle of this scale are synthetic languages, such as Arabic, which combine derivational or inflectional morphemes in different ways and to varying degrees. Given the potentially large difference in the morphology of the translation language pair, it is very important

Pattern	Word	Translation
$C_1aC_2aC_3a$	kataba	*he wrote*
$C_1aC_2C_2aC_3a$	kattaba	*he dictated*
$taC_1AC_2aC_3a$	takAtaba	*he corresponded*
$< inC_1aC_2aC_3a$	<inkataba	*it was written*
$C_1iC_2AC_3$	kitAb	*book*
$C_1uC_2uC_3$	kutub	*books*
$C_1AC_2iC_3$	kAtib	*writer*
$C_1uC_2C_2AC_3$	kuttAb	*writers*
$maC_1C_2wC_3$	maktwb	*written*
$maC_1C_2wC_3$	maktwb	*letter*
$C_1AC_2iC_3$	kAtib	*writer*
$maC_1C_2aC_3$	maktab	*office/desk*
$maC_1C_2aC_3ap$	maktabap	*library/bookstore*

Table 2.3: Examples of words derived from the root $k-t-b$ and their corresponding patterns.

to consider what is the best modeling unit for the translation process. Chapter 3 discusses this issue in more detail. We next describe Arabic morphology briefly.

The derivational morphology of Arabic, like that of other Semitic languages, is based on a root+pattern structure. Arabic words, except for loan words, are derived from an abstract 3-consonant root (2-consonant and 4-consonant roots also exit, although they are rare), by filling specific spots of a template (or pattern) of consonants and vowels with the consonants of the root. The root is usually associated with a semantic notion that relates all the words derived from it. The patterns can also have an associated semantic category, although that is less regular than in the case of the root. The classical example of the Arabic root is $k-t-b$, which designates the notion of writing. The template $C_1AC_2iC_3$ is a nominal template that usually designates an actor. Substituting the root consonants in the slots C_1, C_2, C_3 yields the word *kAtib 'writer'*. Substituting the root $t-j-r$ yields *tAjir 'merchant'*. Table 2.3 shows more examples of words derived from the root $k-t-b$.

A fair amount of work has been done on the morphological analysis and generation

	Singular	**Dual**	**Plural**
	Perfect		
1st Person	**katab**tu	**katab**nA	
2nd Person	**katab**ta/**katab**ti	**katab**tumA	**katab**tum/**katab**tun~a
3rd Person	**katab**a/**katab**at	**katab**A	**katab**w/**katab**na
	Imperfect		
1st Person	>a**ktub**u	na**ktub**u	
2nd Person	ta**ktub**u/ta**ktub**yn	ta**ktub**An	ta**ktub**wn/ta**ktub**na
3rd Person	ya**ktub**u	ya**ktub**An	ya**ktub**wn/ya**ktub**na

Table 2.4: Conjugation of verb *ktb* ‘to write‘. The common verb stem for each of the two conjugations is highlighted.

of Arabic root+pattern morphology, mostly using finite state techniques (Koskenniemi, 1983; Kiraz, 2001; Beesley, 2001). Work on the analysis of concatenative morphology of Arabic, which is more relevant to this thesis, is described in chapter 3.

Arabic has a rich inflectional morphology, where open class words are marked for many grammatical features. Nouns and adjectives are inflected for number (singular, dual, plural), gender (masculine, feminine), case (nominative, accusative, genitive). Verbs are inflected for aspect (perfect, imperfect, imperative), voice (active, passive), mood (indicative, subjunctive, jussive). The subject features (person, number, gender) are also marked on the verb. Inflectional morphology is mostly concatenative. The conjugation of verbs in the perfect tense, for instance, is done with a suffix and the imperfect tense is conjugated with a suffix+prefix, as table 2.4 shows. A notable exception is the formation of irregular plurals (also called *broken plurals*), which is template based. Table 2.3 shows a few examples of irregular plurals.

In addition, meaning-bearing particles, called clitics in the linguistic literature, are also concatenated to verbs and nouns. Clitics that attach to verbs include conjunction, the future tense prefix, and object pronouns. Clitics that attach to nouns include conjunctions, some prepositions, the definite article and possessive pronouns. The order in which these clitics attach to the verb or noun stem depends on their category:

[CONJ+ [PART+ [Al+ STEM +PRON]]]

The following examples show a verb and a noun with attached clitics.

(2.9) wsnqAblhA

w+ s+ nqAbl +hA

and+ will+ we-meet +her

'And we will meet her'

(2.10) wbydk

w+ b+ yd +k

and+ with+ hand +your

'And with your hand'

Chapter 3 describes how these affixed clitics can be detached, and the resulting benefit to SMT.

2.3.5 Arabic Syntax

The syntax of Arabic differs from that of English in many respects. A comprehensive exposition of Arabic syntax is obviously outside the scope of this thesis. This section will describe a few relevant syntactic properties.

The neutral word order of the Arabic sentence is Verb Subject Object (VSO):

(2.11) (a) *rkl* *Alwld* *Alkrp*
kicked-3SM the-boy the-ball
'the boy kicked the ball'

SVO order can also be used with a topicalized subject:

(2.12) (a) *Alwld* *rkl* *Alkrp*
the-boy kicked-3SM the-ball
'the boy kicked the ball'

Recall from section 2.3.4 that the personal object pronoun is a suffix that is attached to the verb, which results in the order $\mathrm{VO}_{pron}\mathrm{S}$ (the subscript is to indicate that this sentence is only valid when the object is a pronoun):

(2.13) (a) *rklhA* *Alwld*
kicked-3SM-it the-boy
'the boy kicked it'

Subject-verb agreement is an interesting pattern. In the VSO order, the verb agrees with the subject in gender and person, while in the SVO order it agrees in gender, number and person:

(2.14) (a) *rklt* *AlbnAt* *Alkrp*
kicked-3SF the-girls the-ball
'the girls kicked the ball'

(b) *AlbnAt* *rkln* *Alkrp*
the-girls kicked-3PF the-ball
'the girls kicked the ball'

Arabic is a *zero copula* language, meaning that the verb *to be* is not overtly expressed:

(2.15) (a) *Alkrp* *mstdyrp*
the-ball round
'the ball is round'

Adjectives follow the nouns they modify, and they agree with them in number, gender and definiteness:

(2.16) (a) *Alkrp* *almstdyrp*
the-ball the-round-fem
'the round ball'

Contrasting example (2.15 a) with example (2.16 a) shows the importance of the definite article in distinguishing a sentence from a noun phrase.

The genitive construction, called *idafa,* is also interesting. A noun is place before another noun to modify it:

(2.17) (a) *krp* *Alwld*
ball the-boy
'the boy's ball'

(b) *krp* *Alqdm*
ball the-foot
'football'

idafa can be constructed hierarchically. The whole phrase is made definite by adding the definite article to the last term:

(2.18) (a) *krp* *wld* *AljyrAn*
ball boy the-neighbors
'the ball of the neighbor's boy'

(b) *krAt* *wld* *AljyrAn*
balls boy the-neighbors
'the balls of the neighbor's boy'

Chapter 3

Morphological Preprocessing for SMT

Statistical machine translation methods aim at learning translation patterns that explicitly take into account the differences in morphology and sentence structure between the two languages. This task is even more challenging when either of the languages has a rich morphology, where the morphemes and grammatical features are exhibited as a change in the surface form of the word. It has been shown that reduction of the sparsity of the vocabulary through morphological preprocessing can lead to improvements in machine translation quality. This chapter presents experiments on the morphological splitting of Arabic for Arabic-to-English translation, providing further evidence for the usefulness of morphological preprocessing techniques on the source side, and comparing different morphological analyzers in terms of their effect on MT performance. The second part of the chapter shows that the same technique of morphological splitting is also beneficial when performed on the target side in the case of English-to-Arabic MT. It also suggests methods for recombining the output of the MT system, which is segmented due to the use of segmented Arabic for training.

Section 3.5 on English-to-Arabic translation is based on work published in (Badr et al., 2008).

3.1 Introduction

One of the advantages of the statistical approach to machine translation is that statistical translation models learn translation patterns directly from training data, and generalize them to handle new data, without explicitly encoding the knowledge required for the different cases. This allows the SMT approach to be language-independent to a large extent. Differences in morphology and sentence structure between the two languages are learned automatically. On the other hand, such translation patterns can be hard to learn completely automatically when the differences in the linguistic characteristics of the two languages are considerable. Another strong point of SMT is its robustness when it comes to the representation of the input, where the only requirement is that the input is consistently represented during training and decoding. Any transformation can be performed on the source language, as a preprocessing step, as long as it is done consistently on the training and test data. Such preprocessing modifications are usually applied on the raw source to make it more suitable for the translation process. This robustness in terms of input representation has been exploited by applying one or more preprocessing steps on the source side to bring it closer to the target language in terms of sentence structure and/or token representation. Such steps include stemming, tokenization, part-of-speech (POS) tagging, and syntactic reordering. They are usually based on specific characteristics of the language pair. When done correctly, the source side preprocessing makes the task of automatically learning translation patterns simpler, which reflects on the quality of the translation output.

The Arabic language is characterized by its complex morphology, compared to English. As section 2.3.4 explained, Arabic verbs and nouns are inflected for gender and number. Morphemes for prepositions, conjunctions, personal and possessive pronouns are also attached to the Arabic words. An Arabic word can thus correspond to an English phrase consisting of multiple words. This means that the number of words in an Arabic sentence is smaller than the number of English words in the corresponding translation. For example, in the parallel corpus used in the section on Arabic-to-English below, the average number of Arabic words per sentence is 23, compared to 32 words on the English side. This discrepancy, sometimes referred to as a *morphological gap,* has two effects on machine translation: First, the number of out-of-vocabulary (OOV) words for Arabic is higher for a given amount of data, since an Arabic word might occur in one inflection (or with one set of affixes) but not another in the data. If the original Arabic source is used directly as MT input, the system will have to learn how to translate an Arabic word separately for all its inflections. If a certain inflected form of a word is not seen in the training data, the system will fail to translate that word all together. The second effect of the morphological gap is that aligning the Arabic and English sides is harder since more one-to-many alignments need to be learned, due to the difference in the number of words between the two sides of the parallel sentence.

In this chapter, we focus on one preprocessing technique that has been used to bridge the morphological gap between Arabic and English, namely morphological segmentation. Affixed morphemes are separated from the stem of the word, based on a morphological analysis that determines the morphemes contained in each word. As will be further explained later, morphological analysis cannot be performed simply based on pattern matching. The morphological analyzer has to draw on morphological, lexical and contextual knowledge to determine the morphemes that constitute a word in context, or in other words whether a given string of characters constitutes an affixational morpheme, or part of the word stem itself. The effect of segmentation on MT is studied in this chapter for both translation directions: Arabic-to-English and English-to-Arabic. Morphological segmentation of the Arabic source for Arabic-to-English MT has been successfully applied before. This work presents a comparison of the use of a rule-based and a statistical morphological analyzer in terms of machine translation performance. The rule-based analyzer used is from Sakhr Software, and the statistical analyzer is MADA (Habash and Rambow, 2005). The section on Arabic-to-English also presents a new morphological splitting scheme called *verb canonicalization,* where the subject pronoun morpheme is split from the verb, and shows additional gains from its use on data from the web domain. The second part of the chapter shows how morphological analysis can also be used when translating into a morphologically complex language. It shows that producing segmented Arabic for English-to-Arabic translation is better than non-segmented Arabic, and suggests different schemes for recombining the segmented output into normal Arabic. The reasons for why this is not a trivial task are also explained.

3.2 Related Work

Most of the previous work on morphological preprocessing of Arabic for SMT has been for Arabic-to-English translation. In one of the earlier works in this area, (Lee et al., 2003) present a morphological segmenter for Arabic based on a trigram language model. (Lee, 2004) uses that segmenter for Arabic-to-English MT, deleting or merging some of the segmented morphemes to make the Arabic align better with the English target.

The only work previously published on English-to-Arabic SMT is (Sarikaya and Deng, 2007). It uses shallow segmentation, and does not make use of contextual information. The emphasis of that work is on using Joint Morphological-Lexical Language Models to re-rank the output. (Habash and Sadat, 2006) and (Sadat and Habash, 2006) use the morphological analyzer MADA, which will be described in further detail later, to segment the Arabic source. They propose various segmentation schemes and study their effect on MT. Both (Lee, 2004) and (Sarikaya and Deng, 2007) show that the improvement obtained from the morphological segmentation decreases with the increase in the size of the training corpus. The same trend is observed in this work, as will be discussed later. The reduction in the gain is due to the fact that, as the size of the training corpus increases, the model becomes less sparse, and the segmentation thus becomes less important.

There has also been work published on translating between English and other morphologically complex languages. Morphological analysis on the source side has been shown to improve results in other language pairs. (Nießen and Ney, 2004) do morphological and syntactic restructuring on German for German-to-English translation. They, for example, attach German verbs to their prefixes, transform the structure of German question sentences to be similar to English, and augment ambiguous German words with their POS. (de Gispert et al., 2006) show that POS tagging, stemming and lemmatization on the source side improve Spanish-to-English translation. (Hakkani-Tür et al., 2000) also preprocess Turkish, an agglutinative language, by splitting complex words based on morphological disambiguation. (Goldwater and McClosky, 2005) perform morphological analysis on the source for Czech-to-English SMT. They replace Czech words with lemmas and abstract morphemes to reduce the source word sparsity. (Popović and Ney, 2004) also separate affix morphemes from source words in Spanish, Catalan and Serbian.

Factored Translation Models (Koehn and Hoang, 2007) is one approach to model morphology explicitly. It is an extension of phrase-based statistical machine translation that allows the integration of additional morphological and lexical features, such as POS, word class, gender, number, etc., at the word level on both the source and the target sides. These features are integrated as additional models at either the source or the target side. A generation model is required on the target side, to generate the surface form from the word factors. The authors claim that the tighter integration is better than either using preprocessing and post-processing, or directly using the word surface form in the translation. The paper shows improvements for translations from English to German and Czech. (Avramidis and Koehn, 2008) enrich the English side by adding a feature to the Factored Translation Model framework that models noun case agreement and verb person conjugation, thus emulating languages with more complex morphology. The paper shows that these features result in more grammatically correct output for English-to-German and English-to-Czech translation.

We next describe in some detail the two morphological analyzers used in this work.

3.2.1 The Sakhr Morphological Analyzer

We briefly describe the Sakhr morphological analyzer, used in section 3.4. The Sakhr morphological analyzer consists of a large base of linguistic knowledge, and a set of rules to decide on a morphological analysis of Arabic words in context. It uses an Arabic lexicon that contains valid stems along with their part of speech (POS), root and pattern, applicable prefixes and suffixes, morphological features (e.g. gender, number, person), syntactic features (e.g. transitivity, agreement), and semantic features (e.g. senses, taxonomies). For each Arabic token, the analyzer generates a list of valid analyses. The correct analysis is determined according to context, using additional information from databases of proper names, idioms, adverbs, and word collocations, as well as rules that use all information contained in the lexicon. The Analyzer uses other resources: a statistical POS tagger and Named-Entity recognizer as well as a database of common spelling mistakes and an Arabic language model for text verification and name detection. The output of the morphological analyzer is also used in subsequent steps of the Sakhr MT process.

3.2.2 The MADA Morphological Analyzer

This section briefly describes the Morphological Analysis and Disambiguation for Arabic (MADA) tool (Habash and Rambow, 2005), used in sections 3.4 and 3.5. MADA is itself based on the output of the Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004), which is a deterministic morphological analyzer that produces a set of morphological analyses of Arabic text using a database of stems, prefixes and suffixes. BAMA produces all possible analyses of a given Arabic word, where each analysis consists of the morphemes that constitute the word, and associated grammatical features (e.g. tense, case, number, gender). MADA is a disambiguation tool that uses Support Vector Machines (SVM) based classifiers trained on the Penn Arabic Treebank (Maamouri et al., 2004). The classifiers are based on 10 morphological features such as POS, gender, number, person, etc. There output is used to decide on the best analysis of the Arabic input from the list of analyses produced by BAMA. The output of the combined classifier can be used for POS tagging as well as Arabic Tokenization.

3.3 Morphological Segmentation of Arabic

As explained in section 2.3.4, Arabic morphemes can be combined in two ways. Certain morphemes are concatenated to the word stem, i.e., they are attached to the word as a suffix or a prefix, with no change or limited change to the form of the stem. A limited change in the orthography of the stem is sometimes needed to reflect certain morpho-phonological rules that the morpheme combination triggers. Other morphemes are combined with the stem non-linearly, where rather than concatenating the morpheme, the stem is rewritten, usually according to a template. Non-linear morphemes include the subject pronoun and the tense in verbs, as well as root+template combinations that produce new words (derivational morphology). The concatenative morphemes, on the other hand, can be further subdivided into two categories: morphemes that represent inflectional features, such as gender, number and person, and "detachable" morphemes, usually called *clitics* in the linguistic literature. These are independent meaning-carrying particles that are attached to the main word in the Arabic orthography. These clitics are attachable to the stem of the word in a particular order, which is shown here:

[CONJUNTION+ [PARTICLE/PREP+ [DEF-ARTICLE+ STEM +PRONOUN]]]

Following is a list of the detachable clitics:

1. Conjunction: *w+*

2. Prepositions: *b+, f+, k+*

3. Future particle (modifies verbs): *s+*

4. Definite article: *Al+*

5. Possessive pronouns (modify nouns): *+(t)y, +k, +h, +hA, +nA, +kmA, +hmA, +nA, +km, +kn, +hm, +hn*

6. Object pronouns (modify verbs): *+(t)y, +k, +h, +hA, +nA, +kmA, +hmA, +nA, +km, +kn, +hm, +hn*

The splitting of the morphemes is based on the morphological analysis (or morphological disambiguation), which determines the morphemes that constitute a given Arabic word from the surface form of that word. Given the morphological analysis, the separation of the morphemes becomes a deterministic, but still non-trivial task. Separating non-linear morphemes requires a lookup in a stem dictionary. Separating concatenated affixes also requires some processing. The separated stem has to be normalized to account for the effect of morpho-phonological rules mentioned above. Example (3.1) shows the splitting of affixational morphemes from a verbal complex. Example (3.2) shows the splitting of morphemes from a noun. Note that the standalone stem is *syArp, 'car',* but the last character, which is a feminine marker, becomes *+t,* when it is followed by a suffix. It has to be normalized back to *+p* when the morpheme is split.

(3.1) wsnsAEdhm

w+ s+ nsAEd +hm

and+ will+ we-help +them

'And we will help them'

(3.2) wbsyArtnA

w+ b+ syArp +nA

and+ with+ car +our

'And with our car'

Given the number of different Arabic morpheme categories, there are many choices as to which morphemes to separate, and whether/how the separated morphemes are combined together. Each of these choices is called a morphological splitting scheme. (Habash and Sadat, 2006) and (Sadat and Habash, 2006) proposed many schemes for the segmentation of the Arabic source for Arabic-to-English translation, with varying degrees of splitting aggressiveness. Those range from simple tokenization of punctuations, to the separation of root and pattern morphemes. They found that separating the affixed clitics listed above results in the most gain, especially with larger amounts of training data, where the problem of vocabulary sparsity and out-of-vocabulary is less severe. It is important to separate the notion of a splitting scheme from the methods and tool for this separation. The same splitting scheme can be implemented using different morphological analyzers.

The following section presents a comparison of two morphological analyzers, the Sakhr rule-based analyzer and the MADA statistical analyzer, in terms of their effect on MT performance. It also presents an experiment in separating non-linear subject pronoun morphemes. In section 3.5, we investigate the use of affix segmentation of Arabic for English-to-Arabic translation, and propose different recombination schemes.

3.4 Morphological Preprocessing for Arabic-to-English SMT

The preprocessing scheme used in this section consists of splitting the affixes listed on page 60, then combining all the split prefixes into one, and the split suffixes into one, so that each word consists of at most three parts: *prefix+ stem +suffix.* The same scheme is implemented using both the Sakhr and MADA morphological analyzers. This section also presents a new preprocessing scheme called *verb canonicalization.* We mentioned above that the Arabic verb is inflected for the gender, number and person of its subject pronoun. These features modify the verb in a non-linear way, rather than being affixed as the object pronouns are. The modification of the verb is according to a pattern that depends on the subject pronoun as well as the triliteral root of the verb and its tense. Verb canonicalization separates the subject pronoun by writing the verb stem in a canonical form, defined in this case to be the 3rd masculine singular form of the verb, while keeping the verb tense. An artificial token is also inserted before the stem to represent the subject pronoun. The motivation here is the usual one of bringing the form of the structure of the Arabic verbal complex even closer to English, where the subject pronoun is a separate token, and the verbal person inflection is minimal (±s). The Sakhr tagger is used to perform this splitting. Table 3.1 shows two example phrases, and their resulting form under the different splitting schemes. The examples show that the resulting canonical form of the verb can look significantly different from the inflected verb. The canonical form can be easily recovered, though, using the lexical information that the morphological analyzer depends on.

3.4.1 Experimental Setup

Before presenting the experimental results for Arabic-to-English with morphological splitting, we describe the experimental setup and data used. We test on two data genres: newswire and web log data. For the newswire genre we use a test set consisting

Orig. Text	<ltqynA	bmhndsyhm	
Affix Splitting	<ltqynA	b+ mhndsyn +hm	
Verb Canon.	<ltqY +SUBJ_1P	b+ mhndsyn +hm	
Translation	*We met their engineers*		
Orig. Text	nzwr	mhndsy	$rkthm
Affix Splitting	nzwr	mhndsy	$rkp +hm
Verb Canon.	yzwr +SUBJ_1P	mhndswn	$rkt +hm
Translation	*We visit the engineers of their company*		

Table 3.1: Examples of Morphological Splitting.

	Newswire OOV	**Web OOV**
Simple Tokenization	0.29%	1.41%
MADA Affix Splitting	0.12%	0.45%
Sakhr Affix Splitting	0.12%	0.43%
Sakhr Canonicalization	0.11%	0.34%

Table 3.2: OOV rate of the different segmentation schemes.

of 3223 sentences, and a tuning set of 2963 sentences. For the web data, the test set consists of 4589 sentences, and the tuning set of 4524 sentences. These data sets were constructed from the following collections: NIST MT04-08 evaluation sets, the GALE Phase 1 (P1) , Phase 2 (P2) and Phase 3 (P3) evaluation sets, and the GALE P2, P3 and P4 development sets. The average length of a sentence is 35 words for newswire and 29 words for web.

The training data consists of around 150 million words of Arabic-English parallel data, aligned using GIZA++ (Och and Ney, 2003), and 7 billion words for the English Gigaword corpus to train the language model. The Hierdec decoder described in section 2.1.5.1 is used.

The baseline in this section uses a simple tokenization scheme where punctuation marks are separated, and certain characters are normalized (final 'y' to 'Y' and all forms of *alif-hamza* to *alif* at the beginning of the words).

3.4.2 Experimental Results

We start by observing the effect of the different splitting schemes on the out-of-vocabulary (OOV) rate, in table 3.2. The OOV rate is calculated as the percentage of words in the test set that do not occur at all in the training corpus. Note first that the OOV rate for the web data is higher than the rate for newswire for all segmentation schemes, which is due to the higher variability for the web domain. Note also that, as expected, the segmentation of affixes reduces OOV for both genres, with very close rates resulting from the use of MADA or the Sakhr morphological analyzer. Verb canonicalization reduces the OOV further, especially for web data.

Table 3.3 contains the translation results for the newswire genre, and table 3.4 contains the results for the web data. In the results tables,BL-Pr is the BLEU precision score, MET is the METEOR score, and Len is the length of the output relative to the reference.The differences in scores of each experiment relative to the baseline are shown immediately below the scores for that experiment. Affix splitting with MADA gives a gain of about 1.5 BLEU points on newswire and 1.9 BLEU points on web. When using the Sakhr morphological analyzer to split the affixes, the gain increases to 1.7 and 2.1 points on newswire and web respectively. Verb canonicalization shows no gain on the newswire set, and a small gain on the web set. Note that this is consistent with the OOV rates where canonicalization reduced the OOV rate on the web data.

3.5 Morphological Preprocessing for English-to-Arabic SMT

3.5.1 Segmentation of Arabic Text

Almost all research in the area of Arabic statistical machine translation has concentrated on the Arabic-to-English direction. As we have seen in the first part of this chapter, the characteristic challenge for that direction is the reduction of the vocabulary sparsity on the source side, which could be mitigated through source-side

	TER	**BLEU**	**BL-Pr**	**MET**	**Len**
	lc	**lc**	**lc**		
	Test.ara.text.nw.v2				
Simple Tokenization	**39.52**	**49.70**	**50.32**	**68.60**	**98.78**
MADA Affix Splitting	38.16	51.19	51.67	69.20	99.08
	-1.36	+1.49	+1.35	+0.60	+0.30
Sakhr Affix Splitting	37.95	51.38	51.96	69.30	98.89
	-1.57	+1.68	+1.64	+0.70	+0.11
Sakhr Verb Canonicalization	38.12	51.35	51.87	69.35	99.00
	-1.40	+1.65	+1.55	+0.75	+0.22
	Tune.ara.text.nw.v2				
Simple Tokenization	**37.25**	**54.37**	**54.37**	**70.19**	**100.04**
MADA Affix Splitting	35.99	55.69	55.69	70.80	100.08
	-1.26	+1.32	+1.32	+0.61	+0.04
Sakhr Affix Splitting	35.24	56.93	56.93	71.30	100.06
	-2.01	+2.56	+2.56	+1.11	+0.02
Sakhr Verb Canonicalization	35.21	56.97	56.97	71.31	100.01
	-2.04	+2.60	+2.60	+1.12	-0.03

Table 3.3: Arabic to English MT results for Arabic morphological segmentation, measured on newswire test data.

	TER	**BLEU**	**BL-Pr**	**MET**	**Len**
	lc	**lc**	**lc**		
	Test.ara.text.web.v2				
Simple Tokenization	**57.43**	**25.77**	**27.26**	**53.18**	**94.66**
MADA Affix Splitting	54.75	27.65	29.20	54.81	94.82
	-2.68	+1.88	+1.94	+1.63	+0.16
Sakhr Affix Splitting	54.48	27.91	29.40	55.04	95.06
	-2.95	+2.14	+2.14	+1.86	+0.40
Sakhr Verb Canonicalization	54.39	28.01		54.87	
	-3.04	+2.24		+1.69	
	Tune.ara.text.web.v2				
Simple Tokenization	**55.00**	**29.23**	**30.48**	**55.40**	**95.99**
MADA Affix Splitting	52.50	31.42	32.77	56.79	95.97
	-2.50	+2.19	+2.29	+1.39	-0.02
Sakhr Affix Splitting	51.77	31.92	33.30	57.37	95.94
	-3.23	+2.69	+2.82	+1.97	-0.05
Sakhr Verb Canonicalization	51.80	31.96		57.22	
	-3.20	+2.73		+1.82	

Table 3.4: Arabic to English MT results for Arabic morphological segmentation, measured on web test data.

morphological splitting. The challenge for the English-to-Arabic direction is a complementary one. In this direction, the MT system is required to output words with complex inflections. The vocabulary sparsity on the target side, which is due to the morphological complexity of Arabic, has a similarly negative effect. Splitting the Arabic target affixes instead of using the raw Arabic can help reduce the vocabulary sparsity in this case as well. The target side of the training data and the Arabic language model would have to be split. The decoder will then output segmented Arabic. A final step in the translation process is, therefore, to recombine into surface form. But this proves to be a non-trivial task for a number of reasons. Before discussing these reasons and describing methods for recombining segmented Arabic, we should mention that the two affix splitting schemes used in the section are:

S1: Declitization, by splitting off the concatenative morphemes listed on page 60.

S2: Same as S1, except that the split morphemes are glued into one prefix and one suffix, such that any given word is split into at most three parts: *prefix+ stem +suffix.* This is similar to the splitting scheme in section (3.4).

An example shows how a compounded prepositional phrase is segmented according to both schemes:

(3.3) wlAwlAdh

S1: w+ l+ AwlAd +h

S2: wl+ AwlAd +h

’And for his children’

The morphological analyzer MADA is used to perform the segmentation in this section’s experiments.

3.5.2 Recombination of Segmented Arabic

As previously mentioned, the segmented output of the decoder has to be recombined to produce a correct form of Arabic as the output of the MT system. But this is not a trivial step, for the following reasons:

1. Morpho-phonological Rules: When morphemes combine, they sometimes undergo phonological modification as a result, which can be reflected in the orthography. For example, the feminine marker *'p'* at the end of a word changes to *'t'* when a suffix is attached to the word. So *syArp +y* recombines to *syArty* ('my car'). The morphological splitter MADA restores the proper form of the stem upon splitting. It is important for the segmented stem to be represented in their proper form for a couple of reasons:

    (a) If the proper form of the stem is not restored upon splitting, the data will contain an unnecessarily large vocabulary size. If *syArty* is split to *syArt +y,* then the data would contain two forms of the stem: *syArp* and *syArt,* which makes the training data unnecessarily sparser.

    (b) The decoder will produce stems in their normal form next to split morphemes, and the post-processing should be able to recombine those properly. So even if *syArt +y* is not normalized, the decoder might still produce *syArp +y,* which the post-processor should be able to combine into the proper form *syArty.*

2. Letter Ambiguity: Data sources are inconsistent in spelling. For example, the character *'y'* is often misspelled as *'Y' (Alf mqSwrp),* at the end of the word. Final *'Y'* is normalized to *'y'* to make the data more consistent. The recombination procedure needs to be able to decide whether a final *'y'* was originally a *'Y'*. For example, *mdy +h* recombines to *mdAh 'its extent'*, since the final *'y'* is actually a *'Y'* that in turn in transformed into a 'A' when attached. On the other hand, *fy +h* recombines to *fyh 'in it'*.

3. Word ambiguity: In some cases, a morpheme tuple (*prefix(es)+stem+suffix*) can be recombined into two different valid forms. One example is the optional insertion of *'n'* (called *nwn AlwqAyp, 'protective n'* in classical grammar), between the stem and the first person object pronoun. So the segmented word *lkn +y 'but I am'* can recombine to either *lknny* or *lkny,* both valid forms.

Given the above reasons, a simple concatenation of the split morphemes would not produce correct Arabic text. A number of recombination schemes are proposed to deal with these issues:

Recombination Scheme R

In this scheme, recombination rules are defined manually. To resolve word ambiguity, the grammatical form that appears most frequently in the training data is picked. To resolve letter ambiguity, we use a unigram language model trained on data where character normalization has not been performed, and choose the most frequent form.

Recombination Scheme T

This scheme uses a table derived from the training set that maps the segmented form of the word to its original form. If a segmented word has more than one original form, one of them is picked at random. The table is useful in recombining words that are split erroneously. Take for example, *qrDay,* which is a proper noun. It gets incorrectly segmented to qrDAn +P:1S, which makes its correct recombination without the table impossible.

Recombination Scheme T+R

Attempts to recombine a segmented word using scheme **T,** and defaults to scheme **R** if it fails.

Scheme	**Training Set**	**Tuning Set**
Baseline	43.6%	36.8%
R	4.04%	4.65%
T	N/A	22.1%
T+R	N/A	1.9%

Table 3.5: Recombination Results. Percentage of sentences with mis-combined words.

3.5.3 Experimental Setup

Segmentation experiments from two data domains were run: Arabic news text, and ISWL — spoken dialog from the travel domain.

For the news domain, data from LDC corpora was used. 2000 sentences were randomly selected for testing, and another 2000 were selected for tuning. The largest training size corpus used was 3 million words, but subsets of 1.6 million and 600K words were also used to measure the effect of training corpus size on the gain obtained from morphological segmentation. 20 million words from the LDC Arabic Gigaword, in addition to 3 million words from the training data were used for language modeling. Experimentation with different language model orders showed that the best results are obtained from using a 4-gram language model for the baseline system, and a 6-gram language model for segmented Arabic. The English source of the parallel data is downcased, and the punctuations are separated. The resulting average number of words per sentence on the English side is 33; for non-segmented Arabic it is 25 words, and for segmented Arabic 36 words. The average number of Arabic words per sentence becomes closer to that of English after segmentation.

For the spoken domain, the IWSLT 2007 Arabic-English corpus was used (Fordyce, 2007). The corpus consists of 200,000 words for training, 500 sentences for tuning and 500 sentences for testing. The Arabic side of the parallel data was used for language modeling, using a trigram for the baseline and a 4-gram for segmented Arabic. A lower order language model was used here because of the smaller size of the data.

The average sentence length is 9 words for English, 8 for Arabic and 10 for segmented Arabic.

GIZA++ (Och and Ney, 2003) was used for alignments, and decoding was done using MOSES . Tuning was done using minimum error training (Och, 2003) to optimize weights for the phrase translation model, distortion model, language model and word penalty for BLEU. The Arabic references of the tuning set were not segmented for the baseline experiments. Two tuning schemes were used for the segmented Arabic experiments: **T1** used segmented Arabic for the reference, and **T2** used non-segmented Arabic.

Factored Models Comparable English-to-Arabic experiments using factored translation models (Koehn and Hoang, 2007) were also performed, to provide a comparison with the preprocessing approach suggested here. These experiments used the MOSES system as well. The factors used on the English side are the POS tag and the surface word. On the Arabic side, we use the surface word, the stem and the POS tag, which is concatenated to the segmented affixes. For example, for the word *wlAwlAdh ('and for his kids')*, the factored words are *AwlAd* and *w+l+N+P:3MS*. A different language model is used for each of the two factor models: a trigram for surface words and a 7-gram for the POS+affix factor. We also use a generation model to generate the surface form form the stem and POS+affix, a translation table from POS to POS+affix and from the English surface word to the Arabic stem. If the Arabic word cannot be generated from the stem and POS+affix, we back off to translating it from the English surface word.

3.5.4 Results

This section presents and discusses results for the translation of English to morphologically segmented Arabic with recombination. It presents and discusses results for recombination accuracy and machine translation.

It is worth noting that the test sets used in these experiments have only one reference available. This negatively affects the BLEU scores, in which the outputs of these experiments are measured, since the BLEU score is a function of *n-gram* precision measured against reference(s). Standard tests set for more common translation directions, such as Arabic-to-English or Chinese-to-English typically provide multiple references (usually 4). The scores presented in this chapter should be evaluated taking this limitation in consideration.

3.5.4.1 Morphological Recombination Results

The method described in section 3.5.2 was run on the Arabic reference of the training and test data. The results for recombination are presented in table 3.5. The results indicate the percentage of sentences in the corresponding data set that contain at least one recombination error.

In table 3.5, the baseline row corresponds to the naive approach of gluing the prefixes and suffixes to the stem without any preprocessing of the stem. In this case, 34.6% of the training sentences and 36.8% of the tuning sentences contain at least one recombination error. When combination scheme **R,** with manually defined rules, is used, the percentage of sentences containing at least one error drops to 4.04% on the training set and 4.65% on the tuning set. This shows the importance of preprocessing the stem according to the ortho-phonological rules, and suggests that the application of these re-write rules for combining Arabic morphemes is relatively frequent.

As mentioned before, scheme **T** uses a table that maps specific morpheme tuples to their recombined forms. The table is derived from the training data. When the segmented tuning data set is recombined using this scheme, the number of sentences with recombination errors is 22.1%. Using the mapping table, therefore, provides less coverage than using the predefined rules. When both schemes are used together, by using the mapping table first, and backing off to using the rewrite rules if the segmented form is not found in the table, the number of sentences with recombination

Training Size	Large 3M	Medium 1.6M	Small 0.6M
Baseline	26.44	20.51	17.93
S1+T1 tuning	26.46	21.94	20.59
	+0.02	+1.43	+2.66
S1+T2 tuning	26.81	21.93	20.87
	+0.37	+1.42	+2.9
S2+T1 tuning	26.86	21.99	20.44
	+0.42	+1.48	+2.51
S2+T2 tuning	27.02	22.21	20.98
	+0.58	+1.70	+3.05
Factored Models + tuning	27.30	21.55	19.80
	+0.86	+1.04	+1.87

Table 3.6: BLEU scores for news data with one reference.

	No Tuning	T1	T2
Baseline	26.39	24.67	-
S1	29.07	29.82	-
	+2.68	+5.15	-
S2	29.11	30.10	28.94
	+2.72	+5.43	-

Table 3.7: BLEU scores for IWSLT data with 1 reference.

errors drops to 1.9%. The conclusion to be drawn is that the mapping table is more reliable than the rules, since it covers certain special cases that the rules might transform erroneously. However, the rules provide better coverage, and using them as a backoff for unseen forms results in a significant drop in the sentence error rate. The scheme **T+R** is used in the translation experiments.

3.5.4.2 Translation Results

This section presents translation results for English-to-Arabic translation on the data sets from the two domains mentioned above: Arabic news text, and ISWLT — spoken dialog from the travel domain.

The translation scores for the news data are shown in table 3.6. The scores are presented in the BLEU metric. Segmentation schemes **S1** and **S2** are defined in section 3.5.1. Two different tuning schemes are used: **T1** tunes using segmented Arabic for the reference of the tuning set, and **T2** uses non-segmented Arabic.

The first thing to note is that the range of scores is lower than that of comparable Arabic-to-English systems. This is partly due to the use of one reference translation for the computation of the BLEU scores, compared to the multiple references typically available in Arabic-to-English test sets. Another factor is that translating to Arabic is a more difficult task than translating in the opposite direction.

To quantify the effect of the training data size on the performance of the different experiments, three corpora with varying sizes are used to train the corresponding systems: a large corpus with 3M words, a medium size corpus with a subset of 1.6M million words, and a small corpus with a subset of 0.6M words. For the same system configuration, lower training data size results in lower BLEU scores, as expected. More interestingly, the gain obtained from morphological segmentation is larger when the size of the training corpus is smaller. This observation is consistent with previous work that uses morphological segmentation (e.g. (Habash and Sadat, 2006)). The reason is that, as the size of the training corpus increases, the out-of-vocabulary rate of the non-segmented corpus decreases, and the corresponding translation models become less sparse, hence reducing the benefit obtained from the segmented data. Segmentation scheme **S2** performs slightly better than **S1** in general, and **T2** is better than **T1** for the news experiments.

Concerning the scores for the IWSLT data (table 3.7), the first thing to note is that they are in the same range as those for the news data (table 3.6), despite the significantly smaller size of the training corpus (3M vs. 200K words for the language model). The reason is that the IWSLT sentences are shorter and have a simpler structure. The gain obtained from segmenting Arabic for the IWSLT data is also larger in relative terms than the gain on the news data, because of the small size of the training data.

3.6 Summary

This chapter provided further evidence of the benefit of morphological segmentation to SMT. It compared the performance of a rule-based and a statistical morphological analyzer, and their effect on the quality of machine translation. It also showed that morphological segmentation on the target side, in the case of English-to-Arabic SMT, results in improvements of MT quality as well, and presented several methods for recombining the segmented Arabic output. The next chapter builds on this with another preprocessing technique for SMT: phrase reordering based on syntactic structure.

Chapter 4

Syntax-based Reordering for SMT

The previous chapter described experiments that use morphological preprocessing for Arabic-to-English and English-to-Arabic SMT. This chapter presents another preprocessing technique, using a different type of linguistic information, namely reordering the source language to better match the phrase structure of the target language. We apply syntactic reordering of the English source for English-to-Arabic translation. The chapter first introduces reordering rules, and motivates them linguistically. It also studies the effect of combining reordering with the morphological segmentation presented in the previous chapter. Results are reported on the newswire domain, UN text data and the spoken travel domains.

This chapter is based on work originally described in (Badr et al., 2009).

4.1 Introduction

One important aspect in which languages differ is their sentence structure, which corresponds to rules of the language grammar that allow constituents to be combined in specific ways. Syntax-based SMT attempts to model these differences directly by using tree-to-tree models. For string-based models though, these differences are manifested as differences in the word order of the corresponding serialized sentences.

Local structural relationships (i.e. with respect to the tree structure of the sentence) can thus appear as long distance relationships in the serialized sentence. For this reason, string models, such as phrase-based SMT systems, have an inherently limited capability in dealing with such long distance linguistic phenomena, since they rely on word alignments that are mostly local. Automatically learned reordering models (called distortion models) that can be conditioned on lexical items from both the source and target are usually used with string-based SMT models, such as phrase-based SMT, providing limited reordering capability to string-based SMT models. But the reorderings in this case are still applied to the sentence string, rather than a representation of the deep structure of the sentence.

One approach that attempts to deal with long distance reordering, while still using string-based models is to reorder the source side to better match the word order of the target language using predefined rules. This is done as a preprocessing step before both training and decoding. The reordering rules are applied to the parse trees of the source sentences, thus indirectly incorporating information on the structure of the source language into the translation process. Despite the added complexity of parsing the data, this technique has been shown to improve on phrase-based SMT, especially when good parses of the source side exist.

This method has been applied to German-to-English and Chinese-to-English SMT (Collins et al., 2005; Wang et al., 2007). The current chapter describes the application of a similar approach to English-to-Arabic SMT. A set of syntactic reordering rules are applied on the English side to better align it to the Arabic target. The reordering rules exploit systematic differences in the sentence structures of English and Arabic. They specifically address two syntactic constructs. The first is the Subject-Verb order in independent sentences, where the preferred order in written Arabic is Verb-Subject. The second is the structure of the noun phrase, where many differences between the two languages exit, among them the order of the adjectives, compound nouns, possessive constructs, as well as the way in which definiteness is marked. These transformations are applied to the parse trees of the English source. It has been observed previously, for instance in (Habash, 2007), that the improvement in translation quality that can be obtained from syntactic reordering depends heavily on the quality of the sentence parses. Since the source language in this work is English, the parses are more reliable, and therefore, the reorderings that are applying based on the parse are more correct. The reason English parsers perform better than parsers of other languages, is that they have been in development for longer, and state-of-the-art advancements in statistical parsing techniques are usually applied to English first.

This chapter also investigates the effects of using the morphological segmentation technique presented in section 3.5 in combination with the syntactic reordering rules. In the rest of the chapter, relevant previous work is presented. A description of the linguistic motivation for this work is then provided. The translation system and data used are presented, together with experimental results on three domains: news text, UN data, and spoken dialog from the travel domain.

4.2 Related Work

This section describes previous work on syntactic preprocessing for SMT. (Habash, 2007) uses syntactic reordering rules for Arabic-to-English SMT. In that work, the rules are automatically learned using word alignments. After the sentence pairs are aligned, the Arabic side is parsed to extract the reordering rules based on how the constituents in the parse tree are reordered on the English side. No significant improvement is observed with reordering when compared to the baseline, which uses a non-lexicalized distance reordering model. This is attributed in the paper to the poor quality of the Arabic parses.

Syntax-based reordering as a preprocessing step has been applied to language pairs other than Arabic-English. Most relevant to the approach presented here are

(Collins et al., 2005) and (Wang et al., 2007). Both parse the source side sentences, and then reorder the sentence based on predefined, linguistically motivated rules. Both suggest that reordering as a preprocessing step results in better alignments, and reduces reliance on the distortion model. Significant gains are reported for both German-to-English and Chinese-to-English translation. (Popović and Ney, 2006) use similar methods to reorder German by looking at POS tags of the German source for German-to-English and German-to-Spanish translation. They show significant improvements on test set sentences that do get reordered as well as those that don't, which is attributed to the improvement of the extracted phrases. (Xia and McCord, 2004) also use reordering rules to improve the translation, but with a notable difference: the reordering rules are automatically learned from the alignment of parse trees for both the source and target sentences. They report a 10% relative gain for English-to-French translations. Although the use of target side parses in their approach is optional, it is needed if full advantage is to be taken from it. This presents a bigger issue when no reliable parses are available for the target language, as is the case with Arabic. More generally, the use of automatically-learned rules has the advantage of being readily applicable to different language pairs, since there is no need to define language-specific rules for each source language or language pair. The use of deterministic, predefined rules, however, has the advantage of being linguistically motivated, since structural differences between the two languages are addressed explicitly. Moreover, the implementation of predefined transfer rules based on source-side parses is relatively easy and cheap to implement in different language pairs.

As mentioned in the previous chapter, different approaches have been proposed for translating from English to more morphologically complex languages. These include Factored Translation Models (Koehn and Hoang, 2007), and enriching source side with morphological features (Avramidis and Koehn, 2008). Although these methods are well equipped for handling languages that differ in their morphology, they still use the same distortion models as phrase-based MT to handle structural-based reordering. (Koehn and Knight, 2003) uses syntactic features to re-rank the n-best output in German-to-English translation.

4.3 Reordering Rules

Section 3.5 showed that there is an advantage to using morphologically segmented Arabic for English-to-Arabic translation. Some of the experiments in this section use segmented Arabic, and the effect of the interaction between morphological segmentation and syntactic reordering is studied. For the experiments that use segmentation, the same segmentation and recombination procedures described in section 3.5 are used.

This section presents the syntax-based rules used to reorder the English side to better match the syntax of the Arabic target. These rules are applied to the English parse tree at the sentence level or the noun phrase level. The reader is also reminded of the relevant syntactic properties of Arabic which motivate these rules. A more comprehensive description of Arabic syntax can be found in section 2.3.5.

Verb Phrase Rules

The structure of the main sentence in Arabic is Verb-Subject-Object (VSO). The order Subject-Verb-Object is also possible, but less frequent. In the SVO order, the verb agrees with the subject in gender and number, but in the VSO order, the verb only agrees in gender with the subject, as the following examples show:

(4.1) (a) *Akl* *Alwld* *AltfAHp*
ate-3SM the-boy the-apple
’the boy ate the apple’

(b) *Alwld* *Akl* *AltfAHp*
the-boy ate-3SM the-apple
‘the boy ate the apple’

	(c)	*Akl*	*AlAwlAd*	*AltfAHAt*		
		ate-3SM	the-boys	the-apples		
		‘the boys ate the apples’				
	(d)	*AlAwlAd*	*AklwA*	*AltfAHAt*		
		the-boys	ate-3PM	the-apples		
		‘the boys ate the apples’				

When the direct object of the verb is a personal pronoun, the pronoun is attached to the verb, as described in section *2.3.5*. So when the subject follows the verb, it follows the object pronoun as well, resulting in a VOS word order. This order will be referred to as VO_{pron}S to indicate that the object has to be a personal pronoun in this case. For example:

(4.2)	(a)	*Akl*	*+hA*	*AlAwlAd*		
		ate-3SM	it	the-boys		
		‘the boys ate it’				

In a dependent clause, the order must be SVO, as illustrated by the ungrammaticality[1] of example (4.3 b).

(4.3)	(a)	*qAl*	*An*	*Alwld*	*Akl*	*AltfAHp*
		said-3SM	that	the-boy	ate	the-apple
		‘he said that the boy ate the apple’				
	(b)	*qAl	*An*	*Akl*	*Alwld*	*AltfAHp*
		said-3SM	that	ate	the-boy	the-apple
		‘he said that the boy ate the apple’				

As discussed in more detail later, this syntactic difference between dependent and independent clauses has to be taken into account when the syntactic reordering rules are applied. Another pertinent syntactic property is that the negation particle has to always precede the verb:

[1] An asterisk in front of the sentence or phrase indicates that it is ungrammatical

(4.4)	(a)	*lm*	*yAkl*	*Alwld*	*AltfAHp*
		not	eat-3SM	the-boy	the-apple

‘the boy did not eat the apple’

Based on these syntactic properties of the Arabic sentence, we define a reordering rule that transfers the English parse tree from SVO order to VSO. Verb phrases are reordered if they have an explicit subject noun phrase and their main verb is not in the participle form, since otherwise the Arabic subject occurs before the verb participle. A check is also made to make sure that the verb is not in a relative clause (example 4.3 b). The following example of a mapped sentence illustrates all these cases:

(4.5) **original:** the health minister stated that 11 police officers were wounded in clashes with the demonstrators

reordered: stated the health minister that 11 police officers were wounded in clashes with the demonstrators

The main clause verb stated is reordered, while the relative clause were wounded is not.

If the verb is negated, then the negation particle is moved together with the verb.

(4.6) **original:** click here if associated images do not appear in your mail

reordered: click here if do not appear associated images in your mail

Finally, if the object of the sentence is a pronoun, then it is moved with the verb to reflect the $\mathrm{VO}_{pron}\mathrm{S}$ structure mentioned above. For example:

(4.7) **original:** the authorities gave us all the necessary help

reordered: gave us the authorities all the necessary help

The reordering has to be applied to the parse tree rather than the sentence string because the subject might consist of a complex noun phrase as the following example shows:

(4.8) **original:** one of the Saudi business institutions, which imports "cream" products from Denmark, set out after the blessed boycott to change the cream label

reordered: set out one of the Saudi business institutions, which imports "cream" products from Denmark, after the blessed boycott to change the cream label

The parse tree, when the parse is correct, provides the boundaries of the NP constituent that forms the subject, thus making the reordering process simple. In principle it is in these situations, when the constituents are quite long, that the reordering should help the translation the most, since those long-distance reorderings would likely not be handled correctly by the lexicalized distortion models of phrase-based SMT.

Noun Phrase Rules

The structure of noun phrases in Arabic also differs from that of English in a number of ways. The adjective follows the noun it modifies rather than preceding in. When the modified noun is definite, the adjective is also marked with the definite pronoun:

(4.9) *AlbAb* *Alkbyr*
the-door the-big
'the big door'

Arabic uses a special construct called *idafa* to express the possessives, compound nouns and the equivalent of the *of*-relationship in English. Idafa compounds two or more nouns. So the English constructs N'_1sN_2 and N_2ofN_1 both correspond to the Arabic N_1N_2. As example (4.10 a) shows, this construct can be chained recursively.

(4.10) (a) *bAb* *Albyt*
door the-house
'the door of the house'

(b) *mftAh* *bAb* *Albyt*
key door the-house
'The key to the door of the house'

Example (4.10 a) also shows that the *idafa* construct is made definite by adding the definite article *Al-* to the last noun in the NP. Adjectives follow the *idafa* noun phrase regardless of which noun in the chain they modify. Thus, example (4.10 a) is ambiguous in that the adjective *kbyr (big)* can modify any of the preceding three nouns. The same is true of relative clauses that modify a noun:

(4.11) *mftAH* *bAb* *Albyt* *Alkbyr*
key door the-house the-big
'the big key to the house door'
'the key to the house's big door'
'the key to the door of the big house'

The differences in the structure of the noun phrase between the two languages are handled by the reordering rules as follows: The order of all nouns, adjectives and adverbs in the noun phrase is inverted. This addresses the difference in noun/adjective order, as well as the *idafa* construct. The following example shows the reordering of a noun phrase:

(4.12) **original:** the blank computer screen

reordered: the screen computer blank

Prepositional Phrase Rule

This rule is motivated by the correspondence between the *of*-construct in English and the *idafa* construct in Arabic. All prepositional phrases of the form $N_1 of N_2 \ldots of N_n$ are transformed to $N_1 N_2 \ldots N_n$. If the prepositional phrase is definite, all definite articles are removed, and a definite article is added to N_n, the last noun in the chain. For example, (4.13) **original:** the general chief of staff of the armed forces

reordered: general chief staff the armed forces

All adjectives in the top noun phrase are also moved to the end of the construct:

(4.14) **original:** the real value of the Egyptian pound

reordered: value the pound Egyptian real

"the" Rule Since the definite article is added to adjectives that modify a definite noun, the definite article is replicated in front of the adjectives as well. This rule is applied after the **Noun Phrase** rule described above. For example:

(4.15) **original:** the blank computer screen

reordered: the blank the computer the screen

The transformation rules **NP, PP** and **"the"** are applied in that order, since they interact, although they do not conflict. The **VP** rule is independent of them. The following example shows the application of several rules to the same phrase:

(4.16) **original:** the real value of the Egyptian pound

reordered: value the pound the Egyptian the real

4.4 Experimental Setup

This section described the experimental setup and data used in the syntactic reordering experiments.

Similar to (Wang et al., 2007), the English side of the corpora is parsed and reordered using the predefined rules. As noted before, the reordering of English can be done more reliably than other source languages, such as Arabic, Chinese and German, since the state-of-the-art statistical English parsers are noticeably better than parsers in other languages. The English source is tokenized and tagged using the Stanford

Log-linear Part-of-Speech Tagger (Toutanova et al., 2003). The data is then split into smaller sentences, and tagged using Ratnaparkhi's Maximum Entropy Tagger (Ratnaparkhi, 1996). The sentences are parsed using the Collins Parser (Collins, 1997), and then person, location and organization names are tagged using the Stanford Named Entity Recognizer (Finkel et al., 2005). On the Arabic side, the data is normalized by changing the final *'Y'* to *'y'*, and changing the various forms of *Alif hamza* to *Alif,* since these characters are written inconsistently in some Arabic sources. The data is then segmented using MADA, in the same way described in section 3.5.

The English source is aligned to the segmented Arabic target using the standard MOSES configuration of GIZA++ (Och and Ney, 2000; Och and Ney, 2003), which uses IBM Model 4 (Brown et al., 1993). Decoding is done using MOSES (Koehn et al., 2007), the same decoder used in section 3.5. A maximum phrase length of 15 is used to account for the increase in length of the segmented Arabic. The setup also uses a bidirectional reordering model conditioned on both the source and target sides, with a distortion limit of 6. The parameter tuning uses minimum error rate training (Och, 2003) to optimize the weights for the distortion model, language model, phrase translation model and word penalty over the BLEU metric (Papineni et al., 2002). For the segmented Arabic experiments, tuning with both segmented and non-segmented data as a reference is done. The recombination of segmented Arabic is done according to the procedure in 3.5.

4.4.1 Data

Experiments were done on data in three domains: newswire text, UN data and spoken dialog from the travel domain. It is important to note that the sentences in the travel domain are much shorter than in the news domain, which simplifies the alignment as well as reordering during decoding. Also, since the travel domain contains spoken Arabic, it is more biased towards the Subject-Verb-Object sentence order than the Verb-Subject-Object order, which is more common in the news domain. Since most of the data used was originally intended for Arabic-to-English translation, the test and tuning sets have only one reference, and therefore, the BLEU scores reported here are also lower than scores typically reported in the literature on Arabic-to-English MT.

The news training data consists of several LDC corpora[2]. A test set is constructed randomly by picking 2000 sentences from the training data, and the tuning set consists of another 2000 randomly picked sentences. The final training set consists of 3 million words (counted on the English side). The system was also tested on the NIST MT 05 test set, while the NIST MT 03 and 04 test sets were used for tuning. The first English reference of the NIST test sets are used as English source, and the Arabic source is used as reference. For the language model, we use 35 million words from the LDC Arabic Gigaword corpus, plus the 3 million words consisting of the Arabic side of the parallel data. Experimentation with different language model orders showed that the optimal model orders are 4-grams for the baseline system and 6-grams for the segmented Arabic. The average sentence length is 33 for English, 25 for non-segmented Arabic and 36 for segmented Arabic.

To study the effect of the amount of training data on syntactic reordering, the UN English-Arabic parallel data is used (LDC003T05). Experiments were run with two training data sizes: 30 million words and 3 million words. For these configurations, 1500 and 500 sentences chosen randomly are used for test and tuning respectively.

For the spoken dialog domain, the BTEC 2007 Arabic-English corpus is used. The training set consists of 200K words, the test set has 500 sentences, and the tuning set has 500 sentences. The language model consists of the Arabic side of the training data. Because of the significantly smaller data size, a trigram LM is used for the baseline, and a 4-gram LM is used for segmented Arabic. In this case, the average sentence length is 9 for English, 8 for Arabic, and 10 for segmented Arabic.

[2] LDC2003E05 LDC2003E09 LDC2003T18 LDC2004E07 LDC2004E08 LDC2004E11 LDC2004E72 LDC2004T18 LDC2004T17 LDC2005E46 LDC2005T05 LDC2007T24

Scheme	**RandT**		**MT 05**	
	S	**NoS**	**S**	**NoS**
Baseline	21.6	21.3	23.88	23.44
VP	21.9	21.5	23.98	23.58
	+0.30	+0.20	+0.10	+0.14
NP	21.9	21.8	–	–
	+0.30	+0.50	–	–
NP+PP	21.8	21.5	23.72	23.68
	+0.20	+0.20	-0.16	+0.24
NP+PP+VP	22.2	21.8	23.74	23.16
	+0.60	+0.50	-0.14	-0.28
NP+PP+VP+The	21.3	21.0		
	-0.30	+0.30	–	–

Table 4.1: BLEU scores for syntactic reordering of newswire data.

4.5 Experimental Results

This section describes and discusses results for the English-to-Arabic MT experiments using syntactic reordering. All results are shown in terms of the BLEU score.

The translation scores for the news domain are shown in table 4.1. The notation used in the table is as follows:

- **S:** Segmented Arabic
- **NoS:** Non-segmented Arabic
- **RandT:** Scores of the test set of sentences chosen randomly
- **MT 05:** Scores of the NIST MT 05 test set

The first column in the results table indicates which combination of reordering rules are used in each configuration. The first thing to note is that the gain obtained from the reordering of segmented and non-segmented Arabic is comparable for most of the reordering schemes. Note also that the gains achieved from reordering on NIST MT test set are smaller than those obtained on the random test set. This is likely due to the fact that the sentences in the NIST test set are longer, which adversely affects the parsing quality. The average English sentence length is 33 words in the NIST test set, while the random test set has an average of sentence length of 29 words. The reordering scheme NP+PP+VP, which applies the 3 reordering rules, shows the most gain on the random test set. The replication of the definite article (*the* rule) before the adjectives in addiction causes a degradation instead, possibly because it increases the sentence length notably, and thus deteriorates the alignment quality.

To get a better insight into the effect of sentence length on the quality of the reordering, the NIST test sets were divided into two subsets depending on the length of the source sentence, and the subsets were scored separately. Short sentences were defined as having less than 40 words on the English side, while long sentences have 40 or more words. Out of the 1055 sentences in the NIST test set, 719 (or 68%) are short and 336 (or 32%) are long. The length-dependent scores are shown in table 4.2. The results show a consistent, although varying gain from all the reordering rules for the shorter sentences. This provides further evidence of the importance of the parsing quality for reordering to be beneficial. We also report, in table 4.3, on the N-best oracle scores for combining the baseline system with the reordering systems, as well as the percentage of the oracle sentences (i.e. the sentences in the N-best list that has the highest score) that are produced by the respective reordering systems. Since the BLEU score is computed for the whole document jointly, the computation of the oracle score cannot be done in the usual way. Instead, it is computed by starting with the candidate translations from the reordered system, and iterating over all the sentences one by one, replacing each sentence with its corresponding translation from the baseline system and computing the BLEU score for the entire set. If the substitution improves the score, then the sentence in question is replaced with the baseline system translation. Otherwise, the reordered system translation is kept, and the next sentence is considered.

Scheme	**S**		**NoS**	
	Short	**Long**	**Short**	**Long**
Baseline	22.57	25.22	22.40	24.33
VP	22.95	25.05	22.95	24.02
	+0.38	-0.17	+0.55	-0.31
NP+PP	22.71	24.76	23.16	24.07
	+0.14	-0.46	+0.76	-0.26
NP+PP+VP	22.84	24.62	22.53	24.56
	+0.27	-0.60	+0.13	+0.23

Table 4.2: BLEU scores for syntactic reordering of newswire data based on sentence length.

Scheme	**Score**	**% Oracle reord**
VP	25.76 (+4.16)	59%
NP+PP	26.07 (+4.47)	58%
NP+PP+VP	26.17 (+4.57)	53%

Table 4.3: Oracle BLEU scores for combining baseline system with other reordering systems.

Table 4.4 shows the results of reordering on the UN test data for different training sizes. It is important to note that although gains from VP reordering stay constant when scaled to larger training sets, gains from NP+PP reordering diminish. This is due to the fact that NP reordering tends to be more localized than VP reorderings. Therefore, with more training data the lexicalized reordering model of the baseline phrase-based system becomes more effective in reordering noun phrases.

Finally, results for the BTEC corpus are reported in table 4.5 for different segmentation and reordering scheme combinations. The first thing to point out is that all the sentences in the BTEC corpus are shorter, simpler and can be more easily aligned than the sentences of the previous test sets. Hence, the gain introduced by reordering is not enough to offset the errors introduced by the parsing. It is also worth mentioning that the preferred sentence order for spoken Arabic is Subject-Verb-Object, rather than the Verb-Subject-Object sentence order typical of written Arabic text. This contributes to the explanation of the lack of gain when the verb

Scheme	**30M**	**3M**
Baseline	32.17	28.42
VP	32.46	28.60
	+0.29	+0.18
NP+PP	31.73	28.80
	-0.44	+0.38

Table 4.4: Oracle BLEU scores for combining baseline system with other reordering systems.

Scheme	**S**	**NoS**
Baseline	29.06	25.40
VP	26.92	23.49
	-2.14	-1.91
NP	27.94	26.83
	-1.12	+1.43
NP+PP	28.59	26.42
	-0.47	-1.02
the	29.80	25.10
	+0.74	-0.30

Table 4.5: BLEU scores for syntactic reordering of the Spoken Language Domain.

phrase is reordered. Noun phrase reordering produces a significant gain with non-segmented Arabic. Replicating the definite article *the* in the noun phrase does not create alignment problems as it does with the newswire data, since the sentences in this case are considerably shorter. A gain of 0.74 BLEU points is thus seen from the application of that rule. That gain does not translate to the non-segmented Arabic, since in that case the definite article *Al* remains attached to its head word.

4.6 Summary

This chapter presented linguistically motivated rules to reorder the English source, making it more similar in structure to Arabic. It showed that these rules produce significant gain in some configurations. The chapter also studied the effect of the interaction between morphological segmentation of Arabic and syntactic reordering on translation results, as well as the effect of the size of the training data on those results.

The effect of sentence length, which is correlated with the quality of the parsing was also described, providing further evidence that this technique depends heavily on the parse quality. This is especially true because the reordering step is applied as a separate preprocessing step. Future work, where for example a “soft reordering”, that is a set of probabilistic partial reorderings, is applied to the source sentence could be one possible way to mitigate against the sensitivity to parsing errors.

Chapter 5

Source Context using Binary Decision Trees

State-of-the-art statistical machine translation models, such as phrase-based or hierarchical SMT, incorporate source language context, by using multi-word modeling units (i.e., phrase pair, hierarchical rule). It has been shown, though, that MT systems built on such models can benefit further from the explicit incorporation of lexical, syntactic, or other kinds of context-informed word features (Vickrey et al., 2005; Gimpel and Smith, 2008; Brunning et al., 2009; Devlin, 2009). The addition of context information usually comes at the expense of increasing the size of the modeling space, which in turn results in a sparser translation model when estimated from the same data corpus. The increase in data sparsity usually has a detrimental effect on translation quality. The challenge is then to balance the advantage of explicitly incorporating more context into the translation model with the shortcomings of the increase in data sparsity.

This chapter presents a method for using context-informed word attributes on the source side, while controlling the amount of context information using binary decision trees. The decision trees decide which context information is likely to help machine translation, based on which information provides the most reduction in the entropy of the translation probability distribution. We present two explicit methods for using the decision tree mechanism proposed. The first method clusters attribute-dependent source words and uses the clusters in training and decoding. The second method uses decision trees to compute an interpolated context-dependent lexical smoothing feature that is used as an additional component of the log-linear model of the decoder. We present experiments that use part-of-speech (POS) tags and diacritics as context information in Arabic-to-English SMT, and show significant gains against a baseline consisting of a state-of-the-art SMT system.

The work in this chapter was published in (Zbib et al., 2010).

5.1 Introduction

Translation, when performed by humans, requires an interpretation of the meaning of the source before it is generated in the target language. This interpretation is highly dependent on the source context, among other things. Context in this case has to be understood in the broad sense. It includes local word context (surrounding words and their properties), as well as discourse context (information across different parts of the text), and extra-textual information, such as the translator's acquaintance with the function of the text, its social or professional context, and how it relates to a larger field of knowledge or expertise. All of these factors have a bearing on the decisions that are made in the translation process.

Lexical ambiguity, where the same word can have more than one (sometimes unrelated) meanings is a common phenomenon in natural language. The different meanings usually translate differently in another language. The Arabic translations of the two meanings of the classical example of the English word *bank* are *mSrf* for the financial institution and *Dfp* for the river bank. The determination of which meaning, and hence which translation is to be used depends on the context in which the source word occurs. But even when lexical ambiguity is not explicit, the translation has to take in more than one word on the source side to produce an appropriately correct and fluent translation. Word-for-word translation becomes worse the larger the difference in the morphology and syntax of the two languages are.

As more data and computational resources have become available, statistical machine translation has made advances in dealing with using context in the translation. State of the art SMT models use translation units that are bigger than single words. Phrase-based SMT models translate multi-word phrases that are extracted from the training data, and hierarchical SMT models use rules extracted from phrase pairs. The size of the translation units cannot be increased arbitrarily though, since that would negatively affect the ability of the model to generalize to unseen data, as the bigger translation units of the test data are less likely to be seen in the training data. Another effect is that the probability estimates that the models use to rank and choose hypotheses become less reliable because of the decrease in the number of samples of each translation unit observed in the training data. Also context for words at the edge of phrases is not taken into account except through the language model and other feature scores.

The use of additional explicit context information can benefit MT, if it can be done in a controlled way. A mechanism for deciding which context information is potentially useful for translation can use that information, but disregard useless context information, thus avoiding the unnecessary increase in the number of model parameters. This chapter proposes the use of decision trees to achieve this goal. The methods proposed can use arbitrary context-dependent or context-informed source word attributes such as POS tags, word context or diacritics. Two specific methods for using decision trees are presented: clustering of source words, and context-dependent lexical smoothing.

5.1.1 Arabic Diacritics as Context-dependent Attribute

As we noted in section 2.3.3, Arabic orthography represents short vowels and consonant doubling with small marks placed above or below the preceding letter, called diacritics. Those are usually omitted in regular Arabic text, including corpora used in Arabic MT systems, which further exacerbates the lexical ambiguity problem. The following examples shows how the name *EmAn* is erroneously translated as *Amman* instead of *Oman.* Diacritizing the name as *EumaA*n would disambiguate it from the form that corresponds to the other name, *Eam~aAn.*

(5.1) **Source:** lm t$hd **EmAn** ywmA kh*A mn* vlAvyn EAmA.

MT Output: Amman did not witness such a day for 30 years .

Ref : Oman did not witness such a day for 30 years.

The second example shows how the word *twgl* is translated by the MT system as the noun *incursion*, corresponding to the diacritization *tawag~ul,* while the right translation is the verb *made an incursion,* which corresponds to the diacritization *tawag~ala.*

(5.2) **Source:** wmydAnyA **twgl** jy$ AlAHtlAl fY jnwb qTAE gzp bEd vlAvp AsAbyE mn AlgArAt Aljwyp Alty Asfrt En Ast$hAd vlAvp wxmsyn flsTynyA

MT Output: On the ground , the occupation army **incursion** in the south of the Gaza Strip after three weeks of the air raids that resulted in the martyrdom of 55 Palestinians

Ref: On the ground , the occupation army **made an incursion** in the south of the Gaza Strip after three weeks of air raids that resulted in the martyrdom of 55 Palestinians

Arabic text can be diacritized automatically with a high degree of precision, which would decrease the lexical ambiguity in the source. Using fully diacritized Arabic source, however, has not been found to improved SMT (Diab et al., 2007). This is likely due to the increase in the size of the vocabulary, which makes the translation models sparser and less reliable. It might also be due to the errors in the output of the diacritizer. The disambiguation information contributed by the diacritics does not seem to offset the negative effect of the increase in the vocabulary size and that of diacritization errors. The work in this chapter shows how automatically diacritized Arabic source can be used beneficially in MT, by using the diacritized form of the source words as a context-dependent attribute in the two ways described above: namely by clustering the diacritized source or by computing a lexical smoothing feature from the diacritized source. Section 5.2.3 reviews the literature on automatic diacritization in some detail.

In the rest of this chapter, section 5.2 reviews previous work relevant to this chapter in several areas. Section 5.3 presents the procedure for growing the decision trees for the source words, and the two methods for using the decision trees in MT. Section 5.4 describes the experimental setup used in this chapter, and section 5.5 presents the experimental results of the various parts of this work. The chapter concludes in section 5.6, where thoughts and preliminary results on clustering rule probability counts to deal with the issue of rule sparsity are presented.

5.2 Related work

5.2.1 Lexical Smoothing

The use of lexical translation probabilities has been shown to improve the performance of machine translation systems, even those using much more sophisticated models. (Och et al., 2004) for instance found that including IBM Model 1 (Brown et al., 1993) word probabilities in their log-linear model works better than most other higher-level syntactic features at improving the baseline. (Gimpel and Smith, 2008) proposed the incorporation of source-side lexical features into phrase-based SMT by conditioning the phrase probabilities on those features. They used word context, syntactic features or positional features. The features were added as components into the log-linear decoder model, each with a tunable weight. (Devlin, 2009) used context lexical features in a hierarchical SMT system, interpolating lexical counts based on multiple contexts. It also used target-side lexical features.

(Sarikaya and Deng, 2007) used Part-of-Speech tags on the target side to model word context. They augmented the target words with POS tags of the word itself and its surrounding words, and used the augmented words in decoding and for language model rescoring. They reported gains on Iraqi-Arabic-to-English translation.

The use of Word Sense Disambiguation (WSD) has been proposed as a way to enhance machine translation also by disambiguating the source words. (Cabezas and Resnick, 2005; Carpuat and Wu, 2007; Chan et al., 2007) Using WSD, however, requires that the training data be labeled with senses, which might not be available for most languages. Also, WSD is traditionally formulated as a classification problem, and therefore does not naturally lend itself to be integrated into the generative framework of machine translation.

5.2.2 Decision Trees

Decision trees have been used extensively in various areas of machine learning, typically as a way to cluster patterns in order to improve classification (Duda et al., 2000). They have, for instance, been long used successfully in speech recognition to cluster context-dependent phoneme model states (Young et al., 1994).

Decision trees have also been used in machine translation, although to a lesser extent. In this respect, the work in this chapter is most similar to (Brunning et al., 2009), where the authors extended word alignment models for IBM Model 1 and Hidden Markov Model (HMM) alignments. They used decision trees to cluster the context-dependent source words. Contexts belonging to the same cluster were grouped together during Expectation Maximization (EM) training, thus providing a more robust probability estimate. While (Brunning et al., 2009) used the source context clusters for word alignments, the current work uses the attribute-dependent source words directly in decoding. The proposed approach can be readily used with any alignment model. This method can also be used to improve alignment quality. The attribute-augmented source words can be clustered using decision trees, then used to obtain alignments. The alignments can then be used to decode attribute-dependent or attribute-independent source sentences. Improvements were obtained when this method was used with GIZA++ alignments.

(Stroppa et al., 2007) presented a generalization of phrase-based SMT (Koehn et al., 2003) that also takes into account source-side context information. They conditioned the target phrase probability on the source phrase as well as source phrase context, such as bordering words, or part-of-speech of bordering words. They built a decision tree for each source phrase extracted from the training data. The branching of the tree nodes was based on the different context features, branching on the most class-discriminative features first. Each node is associated with the set of aligned target phrases and corresponding context-conditioned probabilities. The decision tree thus smoothes the phrase probabilities based on the different features, allowing the model to back off to less context, or no context at all depending on the presence of that context-dependent source phrase in the training data. The model, however, did not provide for a back-off mechanism if the phrase pair was not found in the extracted phrase table. The method presented in this chapter differs in various aspects. Context-dependent information is used at the source word level, rather than the phrase level, thus making it readily applicable to any translation model and not just phrase-based translation. Also, by incorporating context at the word level, decoding can be done directly with attribute-augmented source data (see section 5.3.1).

5.2.3 Arabic Diacritics

Since an important part of the experiments described in this chapter use diacritized Arabic source, this section presents previous work on automatically restoring diacritics and using them in machine translation.

Automatic diacritization of Arabic has been done with high accuracy, using various generative and discriminative modeling techniques. For example, (Ananthakrishnan et al., 2005) uses a generative model that incorporates word level *n-grams*, sub-word level *n-grams* and part-of-speech information to perform diacritization. (Nelken and Shieber, 2005) models the generative process of dropping diacritics using weighted transducers, then uses Viterbi decoding to find the most likely generator. (Zitouni et al., 2006) presents a method based on maximum entropy classifiers, using features like character n-grams, word n-grams, POS and morphological segmentation. (Habash and Rambow, 2007) determines various morpho-syntactic features of the word using SVM classifiers, then chooses the corresponding diacritization.

The experiments in this chapter use the automatic diacritizer by Sakhr Software. In addition to stem diacritization, the Sakhr automatic diacritizer assigns mood ending diacritics at the end of verbs and case endings for nouns and adjectives. The verb moods are the indicative, subjunctive, and jussive. For the nouns and adjectives, the cases are nominative, accusative, and genitive, which could be applied with or without nunation, depending on the definiteness of the noun. Nunation is the addition of a final *n* to a noun or adjective to indicate that it is not definite. The case ending diacritics are determined using rules that depend on adjacency relations with function words like prepositions, articles, demonstrative articles, pronouns, relative pronouns, etc. They also determine case endings for different syntactic structures like noun-noun, noun-adjective, and verb-subject-object relations, with the help of agreement conditions and a selection restriction database. Expressions (e.g., proper nouns, idioms, adverbs, and collocations) are saved in their fully diacritized form whenever possible, to enhance diacritization accuracy. The accuracy of the diacritizer measured on a validation set of around 2000 sentences of newswire data is 97% for stem diacritization, and 91% for full diacritization.

There has been work done on using diacritics in Automatic Speech Recognition (Vergyri and Kirchhoff, 2004). However, the only previous work on using diacritization for MT is (Diab et al., 2007), which uses the diacritization system described in (Habash and Rambow, 2007). It investigates the effect of using full diacritization as well as partial diacritization on MT results. The authors find that using full diacritics deteriorates MT performance. They use partial diacritization schemes, such as diacritizing only passive verbs, keeping the case endings diacritics, or only gemination diacritics. They also find no gain in most configurations. The authors argue that the deterioration in performance is caused by the increase in the size of the vocabulary, which in turn makes the translation model sparser, and is also caused by the errors of the diacritizer.

5.3 Procedure for Growing the Decision Trees

This section describes the procedure for growing the decision trees using the context-informed source word attributes. Details about how they are actually used in translation are described in the following subsections.

Let s be a source word, and let S be the set of attribute-dependent forms of s. So if the attribute used is the source word diacritics, then the elements of S are the diacritized forms of s; and if the attribute used is the POS tag, then the elements of S are the pairs $\langle s, pos_i \rangle$, where pos_i are all the POS tags that source word s is tagged with in the training data. If $s_i \in S$ is an attribute-qualified source word, and t_j is a target word aligned to s_i, then the forward lexical probability is defined as:

$$\begin{aligned} p\left(t_j|s_i\right) &= \Pr\left[s_i \text{ is aligned to } t_j \mid s_i\right] \\ &= \frac{\text{count}(s_i,t_j)}{\text{count}(s_i)} \end{aligned} \tag{5.3}$$

where count(s_i, t_j) is the count of alignment links between s_i and t_j.

$$\begin{aligned} h(S) &= -\text{count}(s_i)\sum_j \frac{\text{count}(s_i,t_j)}{\text{count}(s_i)} \ln p(t_j|s_i \in S) \\ &= -\sum_j \text{count}(s_i,t_j) \ln p(t_j|s_i \in S) \end{aligned} \tag{5.4}$$

is the weighted entropy of $p(t_j|s_i \in S)$, the lexical probability of the attribute-qualified source words in S. The entropy is weighted by the number of samples in the training data of the source words in S.

A separate binary decision tree is grown for each source word. The procedure starts with assigning S, the set of all the attribute-qualified forms of the source word s. At each node n, the list is split into two subsets $(S_1^\star, S_2^\star)$, each assigned to a child node of n, such that:

$$(S_1^\star, S_2^\star) = \underset{\substack{(S_1,S_2) \\ S_1 \cup S_2 = S}}{\text{argmax}} \left\{h(S) - (h(S_1) + h(S_2))\right\} \tag{5.5}$$

In other words, the two-way partitioning of the list that maximizes the reduction in entropy is chosen. This step is repeated recursively.

The weighting of the entropy by the source word counts gives more importance to the context-dependent source words with a higher number of samples in the training data, since the lexical translation probability estimates for these words can be trusted more than those with lower counts. The rationale behind the splitting criterion used here is that, at each node, the split that reduces the entropy of the lexical translation probability distribution the most is also the split that best separates the list of attribute-dependent forms of the source words in terms of the target words to which it translates. For a source word that has multiple meanings, depending on its context, the decision tree will tend to implicitly separate those meanings using information from the lexical translation probabilities.

Global optimization of decision trees is an NP-complete problem (Hyafil and Rivest, 1976). Local optimization criterion, such as this one, are commonly used to grow the tree.

5.3.1 Decision Trees for Source Word Clustering

The first method for using the decision trees is to cluster attribute-dependent source words. A decision tree is grown for each source word as described above, but a node is only split if the reduction in entropy is larger than some predefined entropy threshold θ_h. When the tree cannot be expanded anymore, its leaf nodes will contain a multi-set partitioning of the list of attribute-dependent forms of the corresponding source word. Each of the clusters can be seen as an equivalence class, where all the forms in that class are mapped to the same form (e.g., an arbitrarily chosen member of the cluster). Assuming that the source word tokens occur in the data in their attribute-dependent form, the mappings are used to map these tokens in the parallel training data before they are aligned, and also to map the training data consistently.

The experiments reported on here use diacritics as an attribute type. The various diacritized forms of a source word are thus used to train the decision trees. The resulting clusters are used to map the data into a subset of the vocabulary, which is used in training and decoding. Section 5.5.1 presents the results of these experiments. Diacritics are obviously specific to Arabic; but this method can be used with other attribute types and other languages, by first appending the source words to their context (e.g., attach to each source word its part-of-speech tag or word context), and then training decision trees and mapping the source side of the data.

Augmenting the source words to explicitly include source word attributes (diacritics or otherwise) can make the source text less ambiguous, if the attributes do

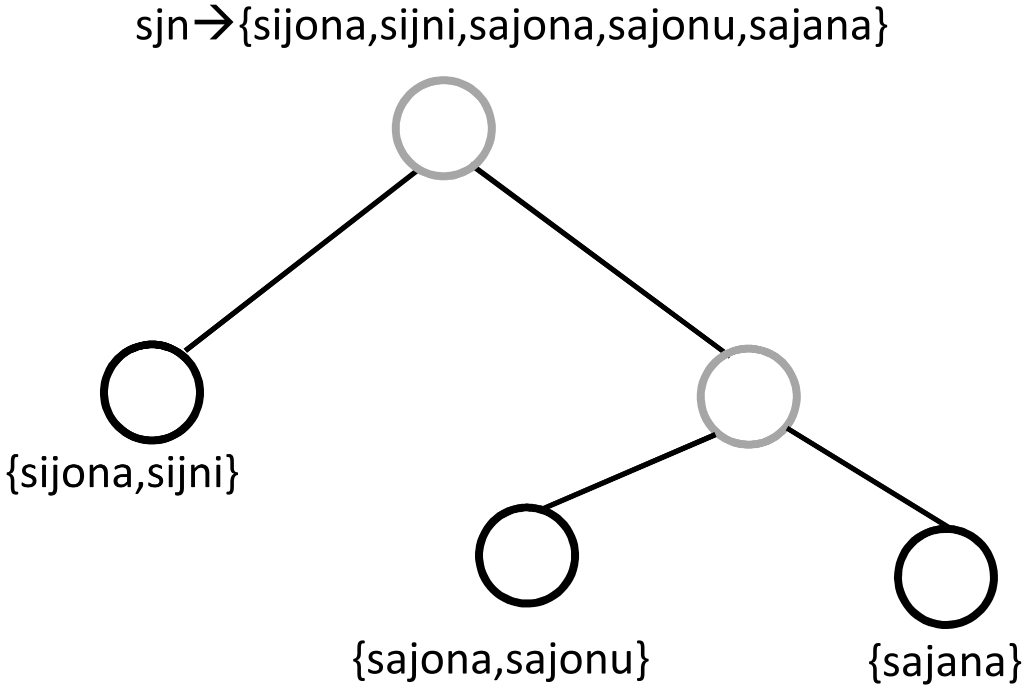


Figure 5-1: Decision tree to cluster diacritized forms of word *sjn*.

in fact contain disambiguating information, which would, in principle, help machine translation performance. The flip side is, as mentioned before, that the resulting increase in the size of the vocabulary increases the translation model sparsity, which in general has a negative effect on translation. The decision-tree based clustering procedure will only keep the attribute-dependent forms of the source words that decrease the uncertainty in the translation probabilities, and would, therefore, be helpful for translation. The sparsity side effect is mitigated by the use of count-weighted entropy in the node splitting criterion, which will tend to keep the attribute-dependent forms of a given source word that occur a sufficient number of times in the training data.

An example for the clustering of the diacritized forms of the word *sjn* is shown in figure 5-1. The root contains the various diacritized forms (*sijona 'prison AC-*

CUSATIVE', sijoni 'prison DATIVE', sajona 'imprisonment ACCUSATIVE.', sajoni 'imprisonment ACCUSATIVE.', sajana 'he imprisoned'). The leaf nodes contain the attribute-dependent clusters.

5.3.2 Decision Trees for Lexical Smoothing

As mentioned in section 5.2.1, lexical smoothing, which is computed from word-to-word translation probabilities, is a useful feature, even in SMT systems that use sophisticated translation models like phrase-based or hierarchical SMT. This is likely due to the robustness of context-free word-to-word translation probabilities of more complicated models, such as extracted phrases or hierarchical rules, which are estimated from much larger sample spaces.

(Devlin, 2009) showed a benefit from incorporating word context into lexical smoothing by interpolating the context-free and context-dependent lexical counts. The interpolation step in this case was critical, because otherwise, the context-dependent lexical probabilities suffer from the same kind of sparsity problems that phrase pair probabilities or hierarchical rule probabilities have. This section presents another method for incorporating source-word information into the lexical smoothing feature, while avoiding the disadvantage of the increased sparsity that results from the addition of the diacritics. Decision trees similar to the ones described in the first method are used to construct a hierarchy of attribute-dependent lexical probability scores, and interpolate these models to compute a new lexical smoothing score.

The lexical smoothing feature is usually computed as:

$$f(\mathbf{U})=\prod_{t_j\in T(\mathbf{U})}\left(1-\prod_{s_i\in\{S(\mathbf{U})\cup\mathrm{NULL}\}}(1-\bar{p}(t_j|s_i))\right) \tag{5.6}$$

where $\mathbf{U}$ is the modeling unit specific to the translation model used. For a phrase-based system, $\mathbf{U}$ is the phrase pair, and for a hierarchical system $\mathbf{U}$ is the translation rule. $S(\mathbf{U})$ is the set of terminals on the source side of $\mathbf{U}$, and $T(\mathbf{U})$ is the set of terminals on its target. The NULL term in the equation above is added to accounts for unaligned target words, which we found in our experiments to be beneficial. One way of interpreting equation 5.6 is that $f(\mathbf{U})$ is the probability that for each target word t_j on the target side of $\mathbf{U}$, t_j is aligned to at least one word s_i on the source side. The feature value is typically used as a component in the log-linear model with a tunable weight.

The method proposed here generalizes the lexical smoothing feature to incorporate the source word attributes. A tree is grown for each source word as described at the beginning of this section, but using an entropy of $\theta_h = 0$. In other words, the tree is grown all the way until each leaf node contains one attribute-dependent form of the source word. By the end of the tree-growing procedure, each node in the tree will contain a cluster of attribute-dependent forms of the source word and a corresponding lexical probability distribution. The lexical translation probability models at the root node are those of regular attribute-independent lexical probabilities. The models at the leaf nodes are the most fine grained, since they are conditioned on only one attribute value. But the deeper the tree level of a node, the fewer alignment link events are available to estimate the word translation probability distribution, and the less reliable these estimates are. So instead of using the leaf node probability estimates to compute the lexical smoothing feature, we perform a recursive interpolation step, where the probability distribution p_n at each node n is interpolated with the probability of its parent node as follows:

$$\overline{p}_n = \begin{cases} p_n & n \text{ is root}, \\ w_n p_n + (1 - w_n)\overline{p}_m & \text{otherwise}, \end{cases} \tag{5.7}$$
$$\text{where } m \text{ is the parent of } n$$

A fraction of the parent probability mass is thus given to the probability of the child node. If the probability estimate of an attribute-dependent form of a source word with a certain target word t is not reliable, or if the probability estimate is 0 (because the source word in this context is not align with t), then the model gracefully backs off by using the probability estimates from other attribute-dependent lexical translation probability models of the source word.

The interpolation weight is a logistic regression function of the source word count at a node n:

$$w_n = \frac{1}{1 + e^{-\alpha - \beta \log\left(\text{count}\left(S_n\right)\right)}} \tag{5.8}$$

The weight varies depending on the count of the attribute-qualified source word in each node, thus reflecting the confidence in the estimates of each node's distribution. The two global parameters of the function, a bias α and a scale β are tuned to maximize the likelihood of a set of alignment counts from a heldout data set of 179K sentences. The tuning is done using Powell's method (Brent, 1973).

During decoding, the probability distribution at the leaves is used to compute the feature value $f(\mathbf{R})$ for each hierarchical rule $\mathbf{R}$. In this method, training and decoding are done using the regular, attribute-independent source. During decoding, the source word attributes are only used to index the interpolated probability distribution needed to compute $f\left(\mathbf{R}\right)$.

Figure 5-2 shows the decision tree for the same sample example word as figure 5-1, except that the tree in this case is grown until each leaf node contains only one diacritized form of the word.

5.4 Experimental Setup

As with most other parts of this thesis, the experiments in this chapter use the string-to-dependency-tree hierarchical translation system based on the model described in (Shen et al., 2008). GIZA++ (Och and Ney, 2003) is used for word alignments. The decoder model parameters are tuned using Powell's method (Brent, 1973) to maximize

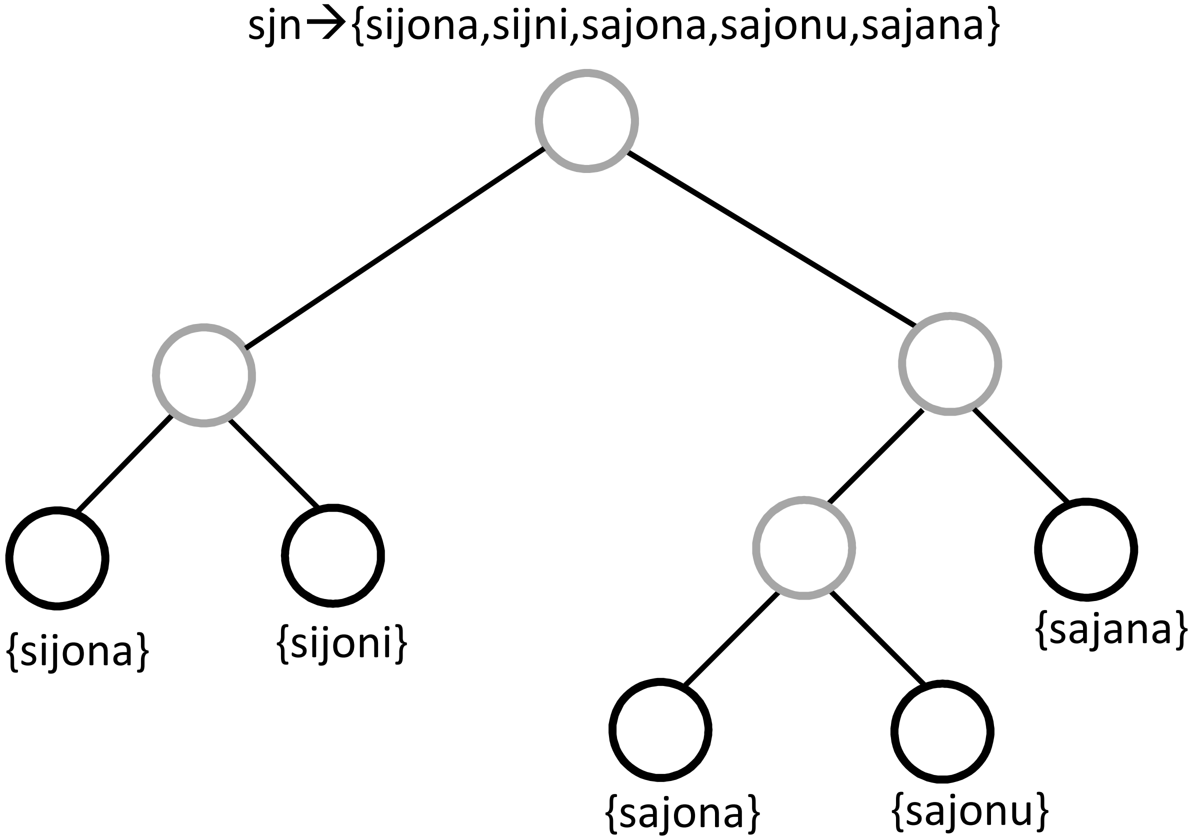


Figure 5-2: Decision tree for the diacritized forms of word *sjn*.

the IBM BLEU score (Papineni et al., 2002). (Rosti et al., 2010) contains a detailed description of the MT system setup.

27 million words from the Sakhr Arabic-English Parallel Corpus (SSUSAC27) are used to train the alignments. The language model uses 7B words consisting of the English Gigaword[1] and of additional data collected from the web.

Tuning and testing are done on two separate data sets consisting of documents from the following collections: the newswire portion of NIST MT04, MT05, MT06, and MT08 evaluation sets, the GALE Phase 1 (P1) and Phase 2 (P2) evaluation

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

sets, and the GALE P2 and P3 development sets. The tuning set contains 1994 sentences and the test set contains 3149 sentences. The average length of sentences is 36 words. Most of the documents in the two data sets have 4 reference translations, but some have only one. The average number of reference translations per sentence is 3.94 for the tuning set and 3.67 for the test set. The baseline for all experiments in this chapter uses morphologically split Arabic source, using the Sakhr morphological analyzer, and the splitting procedure described in 3.4

The next section reports on measurements of the likelihood of test data, and describes the translation experiments in detail.

5.5 Experimental Results

In order to assess whether the decision trees do in fact result in decreasing the uncertainty in the lexical probability on unseen data, we compute the likelihood of the test data with respect to the lexical probabilities with and without decision tree splitting. We align the test set with its reference, and then obtain the link count $l\_count(s_i, t_j)$ for each alignment link $i = (s_{i,}t_i) \in I$, where I is the set of alignment links. The log-likelihood is normalized by the number of links in I:

$$\begin{aligned} L &= \log\left[\left(\prod_i \bar{p}(t_i \mid s_i)^{l\_count(s_i,t_i)}\right)^{\frac{1}{|I|}}\right] \\ &= \frac{1}{|I|}\sum_{i\in I} l\_count(s_i,t_i)\log\bar{p}\left(t_i \mid s_i\right) \end{aligned} \tag{5.9}$$

where $\bar{p}\left(t_i \mid s_i\right)$ is the smoothed probability for the word pair (t_i, s_i) in equation (5.7). If the same instance of source word s_i is aligned to two target words t_i and t_j, then these two links are counted separately. If a source in the test set is out-of-vocabulary, or if a word pair (t_i, s_i) is aligned in the test alignment but not in the training alignments (and thus has no probability estimate), then it is ignored in the

	Likelihood	**%**
Baseline	-1.29	-
Decision Trees with Diacritics	-1.25	+2.98%
Decision Trees with POS	-1.24	+3.41%

Table 5.1: Normalized likelihood of the test set alignments without decision trees, and then with decision trees using diacritics and part-of-speech respectively.

calculation of the log-likelihood. The likelihood should be normalized to make the likelihoods of different test sets comparable.

Table 5.1 shows the likelihood of the baseline case, where one lexical translation probability distribution is used per source word, and compares it with the likelihoods calculated using the lexical distributions in the leaf nodes of the decision trees, for both diacritics and POS as attribute types. The table shows an increase in the likelihood of 2.98% and 3.41% corresponding to diacritics and part-of-speech respectively.

5.5.1 Results for Source Word Clustering using Decision Trees

The decision tree clustering experiment as described in section 5.3.1 depends on a global parameter, namely the entropy reduction threshold θ_h. This parameter was tuned manually. Figure 5-3 shows the BLEU scores of the tuning set as a function of the threshold value with diacritics being used as an attribute type. The most gain is obtained for an entropy value of 50.

The fully diacritized data has an average of 1.78 diacritized form per word. The occurrence-weighted average is 6.28, indicating that words with more diacritized forms tend to occur more frequently. After clustering using a threshold value of $\theta_h = 50$, the average number of diacritized forms becomes 1.11, and the occurrence-weighted average becomes 3.69. The clustering procedure thus seems to eliminate most diacritized forms, which likely do not contain helpful disambiguating information.

Table 5.2 lists the detailed results of the translation experiments that cluster

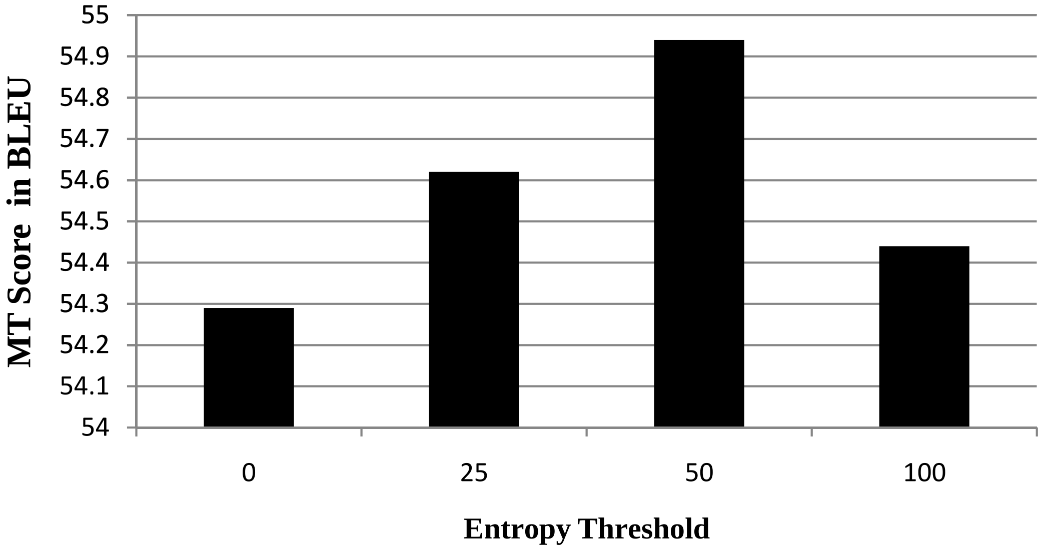


Figure 5-3: BLEU scores of the word clustering experiments as a function of the entropy threshold.

diacritics. The first experiment shows that using full diacritization results in a small gain on the BLEU score, and no gain on TER, which is somewhat consistent with the results obtained by (Diab et al., 2007). The next experiment shows the results of clustering the diacritized source words using decision trees for the entropy threshold value of 50. The TER loss of the full diacritics experiments becomes a gain, and the BLEU gain increases. This confirms the hypothesis presented earlier in this chapter that using the fully diacritized source increases the model sparsity, which undoes most of the benefit obtained from the disambiguating information that the diacritics contain. Using the decision trees to cluster the diacritized source data, on the other hand, prunes diacritized forms that do not decrease the entropy of the lexical translation probability distributions. It thus finds a sweet-spot between the negative effect of increasing the vocabulary size and the positive effect of the disambiguating information.

Recall from section 5.2.3 that grammatical case endings for nouns and adjectives

	TER	**BLEU**	**BL-Pr**	**MET**	**Len**
	lc	**lc**	**lc**		
	Test.ara.text.nw				
Baseline	**40.14**	**52.05**	**52.35**	**68.53**	**99.43**
Full Diacritics	40.31	52.39	52.52	68.25	99.75
	+0.17	+0.34	+0.17	-0.28	+0.32
Clustered Diacs ($\theta_h = 50$)	39.75	52.60	52.94	68.60	99.36
	-0.39	+0.55	+0.59	+0.07	-0.07
Lattice Decoding	39.97	52.39	52.74	68.60	99.34
	-0.17	+0.34	+0.39	+0.07	-0.09
	Tune.ara.text.nw				
Baseline	**39.29**	**54.22**	**54.22**	**69.33**	**100.03**
Full Diacritics	39.49	54.29	54.29	69.03	100.01
	+0.20	+0.07	+0.07	-0.30	-0.02
Clustered Diacs ($\theta_h = 50$)	38.76	54.94	54.94	69.41	100.04
	-0.53	+0.72	+0.72	+0.08	+0.01
Lattice Decoding	38.83	54.71	54.73	69.62	99.97
	-0.46	+0.49	+0.51	+0.29	-0.06

Table 5.2: Results of experiments using decision trees to cluster source word diacritics.

are marked using diacritics, and that the Sakhr diacritizer has the capability of outputting these case endings. In the experiments of this chapter, using diacritics with case endings gave consistently better scores than using diacritics with no case endings, despite the fact that they result in a higher vocabulary size. This suggests that diacritics not only help in lexical disambiguation, but they might also be indirectly helping in phrase reordering, since the diacritics on the final letter indicate the word's grammatical function.

5.2 also shows the results of an experiment that uses lattice decoding (Dyer et al., 2008). We first concatenate the hierarchical rule sets of the baseline and the fully diacritized data, then we construct a lattice from the input sentence by combining the diacritized and non-diacritized versions of the sentence, where each token is represented by two coinciding arcs. This allows the decoder to choose a path that mixes diacritized and non-diacritized words, using translation rules from the corresponding

	TER	**BLEU**	**BL-Pr**	**MET**	**Len**
	lc	**lc**	**lc**		
	Test.ara.text.nw				
Baseline	**40.14**	**52.05**	**52.35**	**68.53**	**99.43**
No Rule Probabilities	41.57	50.25	50.50	67.08	99.51
	+1.43	-1.80	-1.85	-1.45	+0.08
No Lexical Smoothing	42.89	48.53	49.05	66.09	98.95
	+2.75	-3.52	-3.30	-2.44	-0.48
	Tune.ara.text.nw				
Baseline	**39.29**	**54.22**	**54.22**	**69.33**	**100.03**
No Rule Probabilities	40.57	52.24	52.24	68.03	100.00
	+1.28	-1.98	-1.98	-1.30	-0.03
No Lexical Smoothing	41.97	50.28	50.55	66.73	99.47
	+2.68	-3.94	-3.67	-2.60	-0.56

Table 5.3: Effect of removing rule probabilities vs. removing lexical smoothing.

subsets of the rule base. A gain of 0.34 BLEU points results from this experiment.

5.5.2 Results for Lexical Smoothing using Decision Trees

This section presents results for using the decision trees to compute a context-dependent lexical smoothing feature.

We start by providing experimental evidence of the importance of the lexical smoothing feature. The effect of lexical smoothing is compared to that of the hierarchical rule probabilities by running two experiments. In the first one, the rule probabilities are removed, and in the second, the regular, context-independent lexical smoothing attribute is removed. It is important to keep in mind that in the first experiment, the hypotheses are still generated from the same hierarchical rule set. It is just that their probabilities are effectively not used in the ranking of the hypotheses. Table 5.3 shows that removing the rule probabilities results in a degradation of 1.8 BLEU points on the test set, while removing the lexical smoothing feature results in a degradation of around 3.5 points. Despite its being estimated from context-independent word-to-word translation probabilities, the lexical smoothing feature is

	TER	**BLEU**	**BL-Pr**	**MET**	**Len**
	lc	**lc**	**lc**		
	Test.ara.text.nw				
Baseline	**40.14**	**52.05**	**52.35**	**68.53**	**99.43**
Lexical Smooth. (diacs, no interp.)	39.98	52.09	52.56	68.29	99.11
	-0.16	+0.04	+0.21	-0.24	-0.32
Lexical Smooth. (diacs)	39.75	52.55	53.13	68.25	98.90
	-0.39	+0.50	+0.78	-0.28	-0.53
Lexical Smooth. (POS)	40.05	52.40	52.60	68.48	99.63
	-0.09	+0.35	+0.25	-0.05	+0.20
Lexical Smooth (diac, POS)	40.20	52.38	52.64	68.14	99.51
	+0.06	+0.33	+0.29	-0.39	+0.08
Lexical Smooth (diac, POS, POS-1,POS+1)	39.64	52.46	53.16	68.16	98.69
	-0.50	+0.41	+0.81	-0.37	-0.74
	Tune.ara.text.nw				
Baseline	**39.29**	**54.22**	**54.22**	**69.33**	**100.03**
Lexical Smooth. (diacs, no interp.)	39.10	54.48	54.58	69.13	99.82
	-0.19	+0.26	+0.36	-0.20	-0.21
Lexical Smooth. (diacs)	38.55	54.84	55.05	69.22	99.62
	-0.74	+0.62	+0.83	-0.11	-0.41
Lexical Smooth. (POS)	38.80	54.65	54.65	69.37	100.03
	-0.49	+0.43	+0.43	+0.04	+0.00
Lexical Smooth (diac, POS)	38.75	54.74	54.76	69.12	99.96
	-0.54	+0.52	+0.54	-0.21	-0.07
Lexical Smooth (diac, POS, POS-1,POS+1)	38.34	55.07	55.32	69.22	99.55
	-0.95	+0.85	+1.10	-0.11	-0.48

Table 5.4: Results of experiments using the attribute-dependent lexical smoothing feature.

more useful in ranking the translation hypotheses than the sparser hierarchical rule probabilities.

Table 5.4 shows the results of using the decision trees to interpolate context-dependent lexical probability models. The first result is that of a control experiment, where the diacritics-dependent lexical translation probabilities obtained from the decision trees were used, but without performing the probability interpolation step of equation 5.7. The gains mostly disappear, especially on BLEU, showing the importance of the interpolation step for the proper estimation of the lexical smoothing feature. When the interpolation step is performed, the results show a gain of 0.5 BLEU points and 0.39 TER points. Using part-of-speech as an attribute gives a smaller gain. Both of these gains are statistically significant with a confidence interval of 95%, using the random sampling with replacement test proposed in (Koehn, 2004b).

Although the discussion in section 5.3.2 was presented in terms of using one attribute type in the decision trees, extending this method to use more than one attribute type is straight-forward. Table 5.4 shows the results for using diacritics and part-of-speech tags at the same time, with no additional gains. This is likely due to the largely redundant information contained in the diacritics and POS tags. The addition of the POS tags of the previous and next words does not give an additional gain either.

5.6 Conclusion and Future Work

This chapter explored the incorporation of explicit context-informed word attributes into SMT, while controlling the amount of the increase in the number of model parameter, such as the size of the vocabulary by using binary decision trees. We reported on experiments on Arabic-to-English translation using diacritized Arabic and part-of-speech as word attributes, and showed that the use of these attributes increases the likelihood of source-target word pairs of unseen data. Two specific methods were proposed for using the results of the decision tree training process in machine translation, and showed that they both result in improvement in the translation quality. This also constitutes the first successful attempt at using diacritized Arabic source in MT.

Possible future directions include the use of multi-word tree, instead of growing a separate tree for each source word, thus providing more robust estimates of the

	TER lc	**BLEU lc**	**BL-Pr lc**	**MET**	**Len**
			Test.ara.text.nw		
Baseline	**40.14**	**52.05**	**52.35**	**68.53**	**99.43**
Rule Count Clustering	39.91	52.36	52.78	68.35	99.21
	-0.23	+0.31	+0.43	-0.18	-0.22
			Tune.ara.text.nw		
Baseline	**39.29**	**54.22**	**54.22**	**69.33**	**100.03**
Rule Count Clustering	38.92	54.47	54.57	69.27	99.81
	-0.37	+0.25	+0.35	-0.06	-0.22

Table 5.5: Results on clustering target side counts of hierarchical rules based on POS.

translation probabilities on which to grow the decision trees. Also, although the experiments presented in this paper use local word attributes, nothing in principle prevents these methods from being used with long-distance structural attributes, sentence context, or even discourse-level features.

The results in this chapter show the importance of dealing with the issue of sparsity in the estimation of the probability models. Oracle experiments on the system used here show that the same set of extracted rules can produce an output that is better by many points. This suggests that, besides developing new translation models, the pursuit of a line of research aimed at obtaining better probability model estimates on top of the existing translation models will be fruitful.

Section 5.5.2 hinted at the fact that the hierarchical rule probabilities are not as useful a feature for scoring the hypotheses as one would expect them to be. Recall that the hierarchical phrase-based grammar consists of rules of the form:

$$X \to \langle \gamma, \alpha, \sim \rangle$$

where γ is a sequence of terminals and non-terminals in the source language, α is the corresponding sequence in the target language, and $\sim$ is a one-to-one alignment between the non-terminals of γ and α. The rule probabilities are estimated from the joint and marginal counts of the sequences on the source and target sides, which is a much larger space than that of the word-to-word translation probabilities. Most rules are observed only a handful of times in the training data, and many are only seen once. Smoothing the rule probabilities in some way can provide more robust estimates for those probabilities, which would reflect on the MT output quality. Decision trees could be used for this purpose, given that an appropriate criterion for node splitting is used. Simple count clustering can also be used. Results of some preliminary experiments on using clustered rule counts on the target side are shown in table 5.5. The rule counts are clustered by replacing the terminals on the target side of the rules with their POS tags, providing a clustered joint count $c_l(s,t)$ and a clustered marginal target count $c_l(t)$. The clustered probability estimate is then interpolated with the probability estimate of the fully lexicalized rules:

$$\begin{aligned}\overline{p} &= [1-w]p(s,t) + wp_l(s,t) \\ &= [1-w]\,\tfrac{c(s,t)}{c(t)} + w\tfrac{c_l(s,t)}{c_l(t)}\end{aligned}$$

A value of $w = 0.25$ (resulting from manual tuning) gives a gain of 0.3 BLEU points as the table shows. Further work in this direction, where more sophisticated clustering criteria are used, and were the parameters are tuned discriminatively are likely to produce significantly larger gains.

Chapter 6

System Combination of Statistical MT and Rule-based MT

This chapter presents methods on the integration of Statistical Machine Translation (SMT) and Rule-based Machine Translation (RBMT) at the system level. System level integration means that two or more systems are combined in some configuration to produce an output that surpasses in quality the outputs of the individual systems, while treating these individual systems as black boxes. This chapter focuses on the combination of SMT and RBMT systems, which have different, yet complementary advantages.

The first part of the chapter presents “noun-phrase based combination”, a method that uses the RBMT system translation with SMT translations of input noun phrases, to build a word lattice that is then rescored using a number of features. In the second part, preliminary results on extending serial combination are presented. In serial combination, the output of the RBMT system is processed through a statistical post-editing module, which can learn to correct some systematic errors and also makes the output more fluent through a target language model. The preliminary results show how the output of a serial combination system can be rescored using a regular

SMT model, and presents ideas on further development of such an extension. The work in this chapter uses BBN's hierarchical decoder and Sakhr's rule-based Arabic-to-English MT system. The methods presented can however be used with other MT systems and other language pairs.

6.1 Introduction

The statistical approach to MT has some attractive qualities that have made it the preferred approach in MT research over the past two decades. Statistical translation models learn translation patterns directly from data, and generalize them to translate new data, without the need for explicitly encoding the knowledge required to handle the different cases in translation rules. They are also better at handling translations of idiomatic expressions. SMT systems avoid making hard decisions at any intermediate stage during the translation process. They defer such decisions to the end of the process, thus preventing intermediate mistakes from dooming the final result. The SMT approach is largely language-independent; the models that are developed can, in general, be applied to any language pair. Most parts of the system implementations can also be readily used for new languages.

Rule-based Machine Translation systems have other advantages. Some of these systems, especially industrial ones, have matured over decades of development. They contain a wealth of linguistic knowledge at different levels, that is used to perform detailed source-side analysis as part of the translation process, and they exploit specific properties of the two languages for that purpose.

An output analysis in (Thurmair, 2005) comparing the output of a RBMT system (Linguatec's Personal Translator) and an SMT system (Vogel et al., 2000) for German-to-English translation concludes with observations that are consistent with the above characterization of each of the two MT approaches. We next summarize some these observations.

RBMT systems, using a structural analysis of the input (e.g. parse tree) are able to handle long distance reordering better than SMT systems, which translate and reorder input chunks that do not constitute syntactic categories (syntax-based SMT being an exception). For example, (Thurmair, 2005) found that the SMT system often had trouble correctly reordering the German relative clause. RBMT systems, however, are more vulnerable to failure of the input analysis, especially for ungrammatical input. SMT systems are more robust in that they will always produce an output.

SMT systems, in general, have trouble handling the morphology on the source or the target side, especially with morphologically rich languages. Errors in morphology can have severe consequences on the sentence meaning, beyond mere aesthetics. They can change the grammatical function of words, or the interpretation of the sentence through the wrong verb tense. Some SMT approaches, such as factored translation models (Koehn and Hoang, 2007), attempt to solve this issue by explicitly handling morphology on the generation side.

Another observation is that SMT systems are usually better in terms of lexical selection. They avoid early hard decisions on word meaning and word translation. They can also translate idiomatic expressions and common phrases better, when literal translation for such expressions is not adequate, since they learn these translations from the aligned data. RBMT systems need such translations to be explicitly specified in a dictionary. SMT systems could still suffer from making the wrong lexical choice due to lexical ambiguity. Chapter 5 presents methods on how to enhance lexical disambiguation that are specific to Arabic, but can be generalized to other source languages. Also, SMT systems produce a more fluent output, because of the use of a target language model to constraint the output such that it shares statistical properties with the large LM training corpus.

RBMT and SMT systems have, therefore, complementary characteristics, and a hybrid approach could take advantage of the strengths of both. It could leverage the language specific information of RBMT systems to help SMT use such information, which it might not be able to learn automatically. At the same time, it can take advantage of the flexibility of the SMT approach, avoiding hard decisions at any intermediate stage of the translation process.

This chapter proposes a system combination architecture, called "Noun-Phrase based Combination" that attempts to take advantage of the respective strengths of the two approaches. Briefly, the input sentence is first translated via the RBMT system. Noun phrases of the input sentence are then translated using the SMT system. Finally a lattice consisting of the RBMT output, aligned with arcs corresponding to the SMT noun phrase translation is constructed, and re-scored using multiple features including an English language model to select the highest scoring path. Noun phrases are selected to be translated through the SMT system because they, more than other syntactic constituents, tend to contain expressions that cannot be appropriately translated word-for-word. If these expressions are common enough to occur multiple times in the training data, the SMT system can learn proper translations for them. The use of the RBMT translation as the skeleton for the lattice allows the high-level structure of the final output to follow that of the RBMT output, which tends to handle long distance constituent reordering better. Subsequent sections describe noun-phrase based combination in more detail, and present results of this method using the RBMT Sakhr system and the SMT Hierdec system.

The later part of this chapter presents preliminary results on extending another method of system combination between RBMT and SMT systems: serial combination. It then presents some ideas on how this method can be further developed.

6.2 Related Work

The benefit of integrating the rule-based and statistical approaches has long been recognized, and multiple approaches have been proposed for this integration at different levels. (Thurmair, 2009) is a recent comprehensive survey of these approaches. The author classifies the integration methods into 3 categories. The first category is system combination, which the author calls coupling. In this case, the outputs of two or more systems are combined together to produce an output that is better than either of the individual outputs, without introducing changes in any of the systems being combined. A statistical model for combining the outputs is usually used. The advantage of these approaches is that the individual systems are treated as black boxes, which provides a certain flexibility in terms of the types of systems that can be combined.

One of the system combination approaches is serial combination (Simard et al., 2007a; Simard et al., 2007b), where the output of the RBMT system is processed through a statistical post-processing module. Section 6.4 elaborates further on serial combination. (Rosti et al., 2007) and (Rosti et al., 2008) suggest a method for combining systems at the word level, by creating a word lattice from the aligned outputs of the individual systems, and rescoring the lattice with a language model and other features. The final output is thus constructed from chunks of the outputs of the combined systems. This combination method is general, in the sense that it can be used to combine MT systems of arbitrary types, including rule-based and statistical. (Chen et al., 2009) presents a method that uses the MOSES decoder (Koehn et al., 2007) to extract phrases from the aligned outputs of the combined systems. (Eisele et al., 2008) supplant a MOSES-based SMT system with phrase tables from a number of RBMT systems.

The second category of approaches that Thurmair identifies is “architecture extension”. In this category of methods, the translation system essentially falls into one of the two paradigms, rule-based or statistical, but is modified to include resources or models from the second paradigm. Preprocessing methods, such as the morphological and syntactic preprocessing of chapter 4 fall under this category.

The third category of approaches is what Thurmair calls "genuine hybrid architectures", where system components from both paradigms are combined to form a novel system. One example is the multi-language European MT system called METIS (Vandeghinste et al., 2006), which uses rule-based tools, such as lemmatizers, taggers, chunkers, and transfer rules, for source analysis, but a language model for target sentence generation. Another example is context-based MT (Carbonell et al., 2006), which uses a bilingual dictionary to translate words in a *n-gram* window of source words, generating a lattice of *n-gram* translations, which is then scored to generate a final translation.

6.3 Noun-Phrase Based System Combination

As described above, noun-phrase based system combination builds a word lattice from the rule-based translation of the input sentence and SMT translations of noun-phrases in that sentence. It then extends and rescores that lattice.

In the sense used here, a word lattice is an acyclic directed graph with one source node and one sink node. Each arc is labeled with one or more words as well as a vector of scores corresponding to a set of features. If needed, ϵ-arcs can be added to tie multiple source nodes or multiple sink nodes together. A word lattice is a compact representation of MT output hypotheses, where each path through the lattice corresponds to a hypothesis. New translation outputs can be produced from word chunks that belong to different hypotheses, by searching the lattice using feature scores. Efficient search algorithms can be used to extract the top N scored hypotheses.

Figure 6-1 depicts the noun-phrase system combination process as a flowchart. Step 1 correspond to the RBMT translation. Step 2 translates the noun phrases through the SMT system. The system combination is done in step 3. A detailed description each of these steps follows.

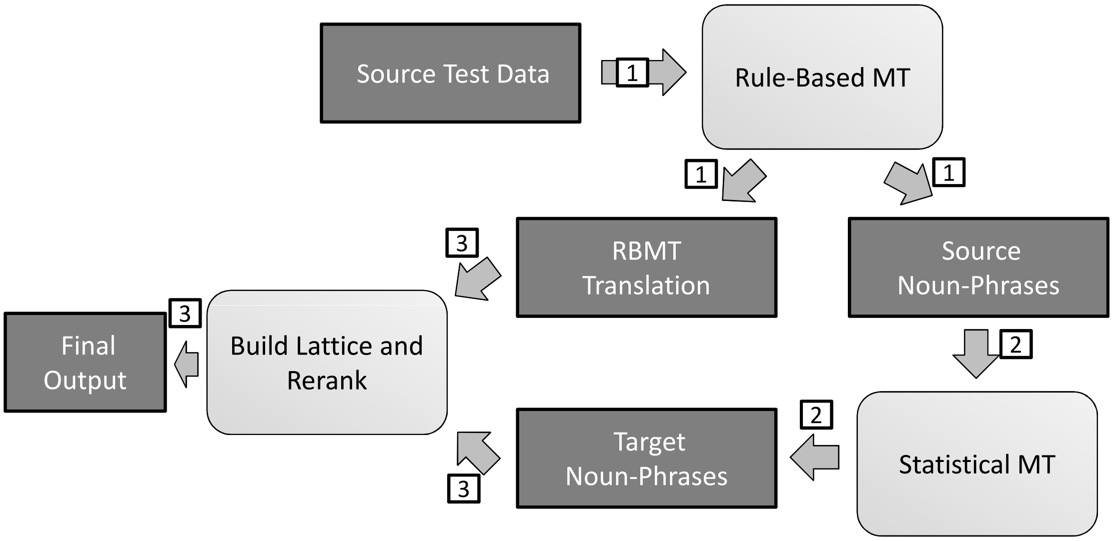


Figure 6-1: Flow Chart of the Noun Phrase Based System Combination Process.

6.3.1 Lattice Construction Procedure

We start with the following definition, which will be useful for the description of the combination method:

Definition 1. *A* base noun phrase *is a noun phrase that contains no other noun phrase within it.*

The first step in the lattice construction procedure is to translate the test data through the rule-based system. Noun phrases are then extracted from the Arabic source and each given a unique id. They are each translated through the SMT System, as separate segments. For nested noun phrases, the nested phrase and the containing phrase are translated separately. In an SMT system, the translation of a given sentence chunk is influenced by its surrounding context. Translating the nested phrases separately from their containing phrases will result in translations that are more varied. This allows the lattice that is built based on these translations to be richer, with more varied translation branches.

It is worth noting that the spans of the noun phrases in the source sentence can be easily mapped to the corresponding spans in the rule-based translations, since the rule-based translation process is deterministic. This makes the mapping of the indices of input spans to the indices of output spans a straight-forward task, unlike in statistical translation.

After the translation step, the output of the rule-based system and the output of the SMT system are combined by building a word lattice for each input sentence. Each arc in the lattice is labeled with a string in the target language (output of either of the two MT systems), and a set of feature values that will be used to select the best path through the lattice. Specifically, the lattice of a given input sentence is built as follows:

1. A base chain of arcs that partitions the RBMT translation of the sentence into consecutive base-noun-phrase and non-noun-phrase spans, such that no single noun-phrase arc contains a nested noun-phrase. Each arc is labeled with the RBMT output corresponding to its span.

2. For each non-base noun phrase in the input segment, an arc that spans that noun phrase is added to the lattice, such that the edges of the arc coincide with the start and end nodes in the base chain constructed in step 1. The arcs are labeled with the corresponding output of the RBMT system.

3. For each noun-phrase in the lattice (both base and non-base), a set of parallel arcs is added, each labeled with a one of the N-best translations of the SMT system.

4. In addition, each are is also labeled with a set of word-level and arc-level features.

The addition of the arcs in step 2 might seem redundant. But having a single arc in the lattice for each noun phrase allows separate feature values to be assigned to those noun phrases.

Section 6.3.1.1 enumerates the types of features used. Each feature is assigned a global weight. If $\mathbf{x_i} = [x_{i1} \ldots x_{im}]$ is the feature vector for arc $\mathbf{i}$, and $\mathbf{w} = [w_1 \ldots w_m]$ is the feature weight vector, then the total score for arc $\mathbf{i}$ is the dot product:

$$\mathbf{w}.\mathbf{x_i} = \sum_{\mathbf{j}} w_i x_{ij}$$

The score of a path in the lattice is the sum of the scores of its arcs.

The weights of the features are tuned to maximize the BLEU score. The objective function is optimized using Powell's method (Brent, 1973), since it is not directly differentiable. A list of N-best translations are extracted from the lattices by searching for the N-top scoring paths in the lattice using the A* algorithm (Hart et al., 1968).

A 3-gram language model is used to score the lattice paths and extract an N-best list of hypotheses. The N-best list is then re-ranked with a 5-gram language model to select the top candidate translation. The complete combination process is summarized in figure 6-1.

Figure 6-2 shows the lattice construction process through an example. In the figure, the base chain is constructed from the rule-based translation, namely the string *abcd*. In this sentence, *b* is a base noun phrase and *bcd* is a non-base noun phrase. As mentioned above, the rule-based translation for noun phrase *bcd* is added as one arc, in order to assign arc-level features to it. The SMT translations for the two noun phrases are then added. For the noun phrase *c*, the translations $c_1, c_2, \ldots, c_i$ are added in parallel, between nodes 2 and 3. For noun phrase *bcd*, the translation $bcd_1, bcd_2, \ldots, bcd_i$ are added between nodes 1 and 4.

6.3.1.1 Lattice Features

The following features are used to calculate the arc scores:

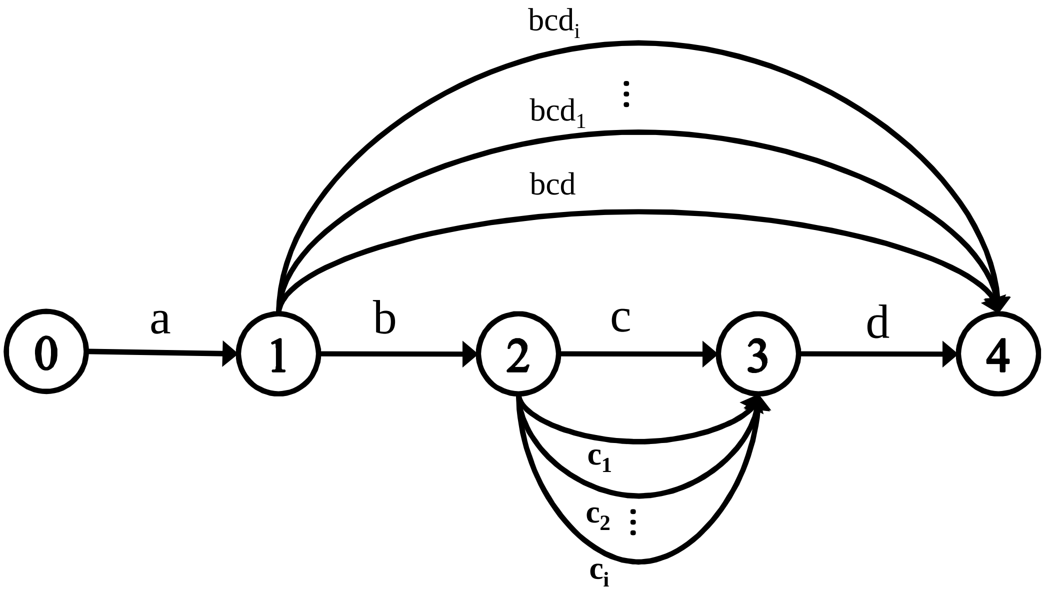


Figure 6-2: Lattice Construction. The horizontal links constitute the base chain. Additional arcs are labeled with alternative translations for the noun phrases bc and bcd.

1. SLM: The language model score. A 3-gram language model was used in the experiments presented in the chapter.

2. SMT_ARC: Feature whose value is 0 if the arc translation is from the RBMT system, and 1 if it is from the SMT system.

3. W_NUM: the number of words on the arc.

4. A_NUM: a constant. This feature contributes to a path-level feature whose value is the number of arcs in the path.

5. IS_NP: A feature with value 1 if the arc corresponds to a noun-phrase span, and 0 otherwise.

6. IN_NON_BASE_NP: Feature with value 1 if the arc corresponds to a non-base

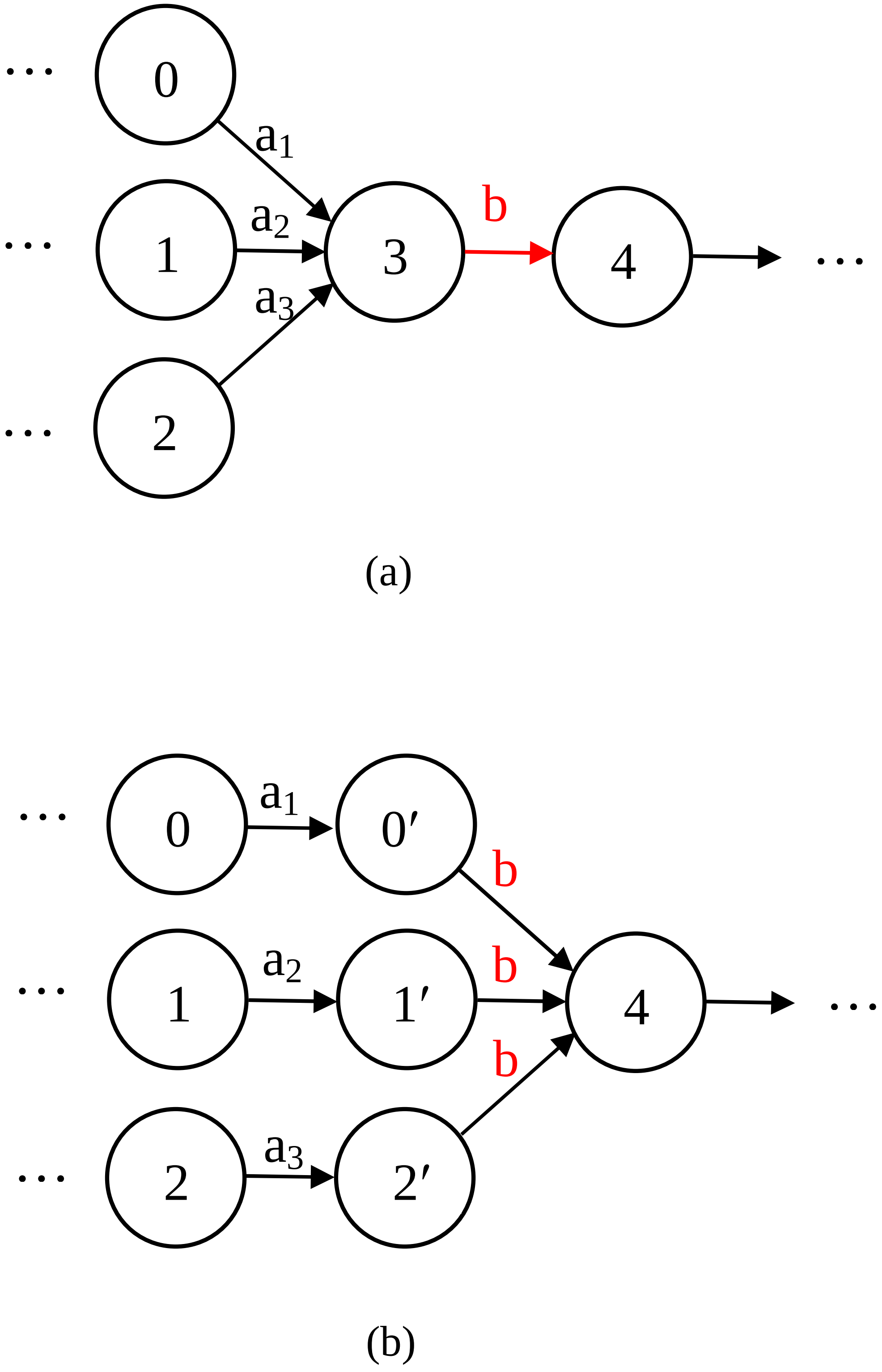
noun phrase, and 0 otherwise.

The use of non-lexical features such as the number of arcs, the number of words or the system that produces the arc translation, each with a tunable weight allows the system to incorporate a preference for the properties that each of these features represents.

All feature values can be calculated and added to the lattice arcs in a straightforward manner when the lattice is being built, with the exception of the language model score (feature SLM). Assigning the language model scores properly requires the additional step of expanding the lattice. The procedure, first proposed in (Odell, 1995) is briefly described again here.

If a language model of order n is used, then the history of $n-1$ previous words is needed to calculate the language model score for a given arc string (that is the target language word or words that the arc is labeled with). But, in general, a lattice arc will have more than one arc that are incoming into its source node (call them preceding arcs). So if an arc has i preceding arcs, the first word of its string will have i word histories. And each of those can have another j preceding arcs, and word histories, and so on. The word history of the arc string, therefore, depends on the preceding arcs in a given path through the lattice.

To simplify the computation of the path scores, the lattice is expanded so that each arc has a unique *n-gram* history. This is done by replicating an arc with i preceding arcs i times. Each arc replica is assigned the same arc string and feature vector as the original arc, but is linked to one of the incoming arcs. Since the arc string can consist of only one word, the expansion step has to be propagated back, possibly as many as to $n-1$ times. This will guarantee that each arc will have a unique *n-gram* word history. The language model score at each new arc can now be computed using that unique word history. Figure 6-3 illustrates lattice expansion through an example. Although the experiments described in this chapter used a 3-



(a)

(b)

Figure 6-3: Lattice Expansion. Node 3 in (a) is expanded into nodes $0'$, $1'$, and $2'$ in (b), to make the three paths disjoint; then the language model score of each can be calculated separately.

gram language model, the illustrated example is shown with a bigram for simplicity. In the example, arc b has three preceding arcs, so it is replicated 3 time, so that each replica has one preceding arc. The three partial paths are thus made disjoint, and the scores for bigrams a_1b, a_2b and a_3b can be computed separately.

6.3.1.2 N-best Reranking

A 3-gram language model is used to expand and score the lattice. Then a list of the top 300 scored translations is extracted and re-ranked using a 5-gram language model in addition to the values of the features above. The process is broken up into 2 steps because 5-gram model is quite expensive, especially in terms of space, because of the longer history of each arc. The re-ranking still takes advantage of the stronger 5-gram model to choose among candidate hypotheses, even though the search results might be sub-optimal.

Using a lower-order n-gram language model during search, then using a stronger, higher order language model to re-rank is a common practice. For instance, the standard Hierdec translation procedure uses a 3-gram language model during decoding to procure an N-best list of candidate translation hypotheses, and those are in turn re-ranked using 5-gram language model scores.

6.3.2 Experimental Setup

The Arabic-to-English Sakhr rule-based translation system is used to translate the test data, providing the skeleton for the lattice. The first two components of the system are a morphological analyzer that generates a list of features for each word in context, including a part-of-speech tag, and a unique sense, and an automatic diacritizer, which uses the output of the morphological analyzer as well as other rules to assign a stem and case ending diacritization to each word. The outputs of these components are used in 4 and 5 respectively. The MT system proper uses the output of the morphological tagger and the diacritizer, together with Arabic grammar rules to produce a rich parse of the source sentence. Transfer rules, and an Arabic-to-English lexicon are then used to transform the Arabic parse tree to English. A generation step is then applied to the output sentence in order to make it more grammatical. This step applies agreement rules among other things. The last step is to make the output more fluent by applying surface transform rules, and a database of English expressions.

The noun phrases are translated using the Hierdec (Shen et al., 2008) string-to-dependency tree hierarchical decoder. 200 million words of Arabic-English parallel data are aligned using GIZA++ (Och and Ney, 2003). The weights of the decoder are tuned using minimum error rate training (Och, 2003). The system is tuned to maximize the BLEU score. 7 billion words of English data are used to train the language model. The decoding uses a 3-gram language model and the N-best output of the decoder is re-ranked using a 5-gram language model trained on the same amount of data. The 3-gram language model is used to rescore the combined lattice.

Experiments were run on two data genres: newswire and web newsgroups. For the newswire experiments, 2040 segments selected from the NIST02 to NIST05 test sets are used for testing. For tuning, 2075 segments from the NIST02 to NIST05 tuning sets are used.

For the web data, two data sets are used: ng_ y1q4_Tune contains 2079 segments and ng_y1_q4 Test contains 2128 segments, all selected from the GALE year 1 quarter 4 LDC web parallel data. The web data has therefore one reference, which partly explains the lower scores in the results table in the next section.

6.3.3 Experimental Results

6.3.3.1 Examples

We first present a few examples, comparing the outputs of the different systems.

(6.1) **Source:** mwskw 32-01 (Afb)- dEA wzyr xArjyp rwsyA Aygwr AyfAnwf

	TER	BLEU	MET
	NIST_MT02_05_Tune		
Hierdec	38.73	55.42	72.02
Sakhr RBMT	51.74	37.13	56.83
	+13.01	-18.29	-15.19
NP Comb 1-best	44.20	45.79	69.32
	+5.47	-9.63	-2.70
NP Comb 50-best	43.45	46.80	68.87
	+4.72	-8.62	-3.15
	NIST_MT02_05_Test		
Hierdec	38.95	55.48	71.44
Sakhr RBMT	51.42	36.86	56.83
	+12.47	-18.62	-14.61
NP Comb 1-best	44.20	46.07	69.25
	+5.25	-9.41	-2.19
NP Comb 50-best	43.48	46.92	68.97
	+4.53	-8.56	-2.47

Table 6.1: Results of noun-phrase based combination for Arabic newswire data.

Alywm AlAvnyn AlY AETA' rwsyA wAlAtHAd AlAwrwby dwrA ADAfyA fy Emlyp AlbHv En Hl lAzmp Al$rq AlAwsT.

RBMT: Moscow 32 - 01 (AFP) - Russia Foreign Minister Igor Ivanov called today Monday for the giving of Russia and The European Union an additional role in the search operation about a solution to the Middle East crisis .

SMT: Moscow 32-01, Russia and the European Union (AFP) - Russia's Foreign Minister Igor Ivanov called on Monday to give additional role in the process of searching for a solution to the Middle East crisis.

NP-Syscomb: Moscow 32 - 01 (AFP) - Russian Foreign Minister Igor Ivanov called today Monday for giving Russia and the European Union an additional role in search about a solution to the Middle East crisis.

Reference: Moscow 10-23 (FP) - Russian Foreign Minister Igor Ivanov called for Russia and the European Union to be given an additional role in the

	TER	BLEU	MET
	ng_y1q4_Tune		
Hierdec	60.41	19.76	47.06
Sakhr RBMT	69.02	15.69	45.11
	+8.61	-4.07	-1.95
NP Comb 1-best	62.93	18.98	48.07
	+2.52	-0.78	+1.01
NP Comb 50-best	62.64	18.84	47.92
	+2.23	-0.92	+0.86
	ng_y1q4_Test		
Hierdec	61.72	17.03	44.15
Sakhr RBMT	67.96	15.65	44.65
	+6.24	-1.38	+0.5
NP Comb 1-best	63.39	17.48	46.10
	+1.67	+0.45	+1.95
NP Comb 50-best	63.37	17.19	45.84
	+1.65	+0.16	+1.69

Table 6.2: Results of noun-phrase based combination for Arabic web data.

process to find a solution for the Middle East crisis.

(6.2) **Source:** AyfAnwf: rwsyA "Trf m$Ark kAml" fy Emlyp AlslAm fy Al$rq AlAwsT

RBMT: Ivanov: Russia "a complete participant end" in the Middle East peace process

SMT: Ivanov: Russia "full participant" in the peace process in the Middle East

NP-Syscomb: Ivanov: Russia "a full participant" in the Middle East peace process

Reference: Ivanov: Russia "fully active partner" in the Middle East peace process

Note that for example 6.1, the RBMT system produces a correct high-level sentence structure, while the SMT system mis-places the phrase “Russia and the European

Union". But the RBMT system translates "process to find a solution" as "search operation about a solution", which is the literal translation of the Arabic expression, but is not an adequate translation. The noun-phrase combination output maintains the correct ordering of the sentence, but gives a more adequate and fluent output. Example 6.2 shows how the noun-phrase combination provides a better translation of the phrase "fully active partner", as "a full participant", compared to the rule-based translation "a complete participant end".

6.3.3.2 Results

This section presents and explains results from the noun-phrase based system combination experiments. For each genre, the results are compared against the Hierdec SMT system using the same data and setup described in section 6.3.2 above.

The first thing to note is that the scores for the rule-based system are considerably lower than those for the SMT system. This is a typical phenomenon since SMT systems trained on large amounts of data are constrained with a target language model, and therefore produce more fluent output that is favored by the automatic evaluation metrics. For newswire (table 6.1), the rule based system scores are around 18 BLEU points less than the SMT baseline, and 12-13 TER points higher. For the web data, the gap is smaller: around 6 TER points and 1.38 BLEU points on the test set.

The web data scores are significantly worse in absolute value for all the systems compared to the newswire data. One reason is that, as mentioned in the previous section, the web data sets are scored with one reference hypothesis only. The second reason is that web data is more difficult to translate than newswire data. It is less well-structured and contains more variation. Web data typically has a larger out-of-vocabulary rate (i.e. percentage of source words in test data that are not found in the training corpus) than newswire data. The out-of-vocabulary rate for the newswire set is 0.31% compared to 0.90% for the web data set used in this section.

The noun-phrase combination systems improve on the scores of the rule-base output significantly. Using the 1-best from the output of the noun-phrase SMT translations, the noun-phrase output for newswire is 7 TER points and 9 BLEU points better for NIST_MT02_05_Test (see table 6.1). Similar improvements are observed for NIST_MT02_05_Tune. But the scores remain significantly lower than those of the SMT baseline. The use of the 50-best hypotheses from the translation of the noun-phrases in the combination provides an additional point gain.

For web data, noun-phrase system combination provides a small gain of 0.45 BLEU points and 1.95 METEOR points compared to the SMT baseline, as table 6.3 shows. The fact that noun-phrase based combination results in a gain for the web data is partly due to the smaller gap between the RBMT and SMT scores for web data. It is also likely that the web data translations benefits more from noun-phrase combination because it contains more phrases and expressions that are not translatable word-for-word, and are therefore harder to translate using a rule-based system. The use of the 50-best hypotheses from the noun-phrase translations provides no additional gain.

6.3.3.3 Word-level System Combination

Word-level system combination (Rosti et al., 2007; Rosti et al., 2008), mentioned in section 6.2 above, is a general method for combining the outputs of multiple translation systems of any kind. A word lattice is built from the outputs of the different systems by incrementally aligning the output words. A language model score and other feature scores are then used to expand and rescore the lattice. For systems that produce multiple translation hypotheses, an N-best list, rather than the top hypothesis is typically added to the lattice. This combination method is most useful when the systems being combined produces output that is varied enough for the lattice to contain hypotheses that are better than those produced by any of the individual systems.

Experiments were run combining the noun-phrase based system with the two

	TER	**BLEU**	**MET**
		ng_y1q4_Tune	
Hierdec	60.41	19.76	47.06
Sakhr RBMT+Hierdec	62.72	20.17	50.53
	+2.31	+0.41	+3.47
Sakhr RBMT+Hierdec+NP-Comb	61.28	21.28	51.26
	+0.87	+1.52	+4.20
Sakhr RBMT+NP-Comb	64.06	19.25	50.56
	+3.65	-0.51	+3.50
Hierdec+NP-Comb	61.14	21.26	51.20
	+0.73	+1.50	+4.14
		ng_y1q4_Test	
Hierdec	61.72	17.03	44.15
Sakhr RBMT+Hierdec	63.40	18.30	48.34
	+1.68	+1.27	+4.19
Sakhr RBMT+Hierdec+NP-Comb	61.87	19.25	49.23
	+0.15	+2.22	+5.08
Sakhr RBMT+NP-Comb	63.96	18.11	49.24
	+2.24	+1.08	+5.09
Hierdec+NP-Comb	61.95	18.86	49.19
	+0.23	+1.83	+5.04

Table 6.3: Effect of noun-phrase system on word-level combination for Arabic web data.

baseline systems (SMT and RBMT) using the word-level combination described in section 6.3.2. The experiments were run on web data, using different combinations of the three systems. The 10-best hypotheses for the two statistical systems (Hierdec baseline and noun-phrase based) were used.

The results of the word-level combination experiments are presented in table 6.3. The improvements in results are all relative to the Hierdec SMT baseline. First, when the SMT and RBMT baselines are combined, a gain of 1.27 BLEU points and 4.19 METEOR points are observed, as shown in the second row of table 6.3. No improvement in the TER score is obtained though. The addition of the noun-phrase based system to the combination increases the gain to 2.22 BLEU points and 5 METEOR points. The combination of the noun-phrase based system with either of the two baseline systems provides a smaller gain than the gain obtained by combining the 3 systems.

The additional gain obtained from adding the noun-phrase based system to the word-level combination of the SMT and RBMT systems indicates that the noun-phrase based system produces an output that is different enough from either of the two other systems to be beneficial in the combination.

6.4 Enhanced Serial System Combination

Word-based system combination and Noun-phrase based system combination can both be characterized as parallel combination methods, since in both methods hypotheses from the different systems are combined in a lattice, from which a new hypothesis is produced. Rule-based MT and SMT systems can also be combined in a serial fashion. The source side of the training data is first translated through the rule-based system, producing output in the target language. But this output typically has different characteristics from the natural target language data. For instance, it will have a different n-gram distribution, and it will be characterized by translation patterns produced by the rule-based system. Call this "new language" target′. A statistical system can be trained using the rule-based translations as a source, and the original target (i.e. from target′ to target). To decode, the test data is first translated through the rule based system, and the output is then translated through the SMT system. Figure 6-4 depicts the flowchart of the serial combination process.

As mentioned at the beginning of this chapter, RBMT systems usually have high-quality handcrafted translation rules which encode a large amount of morphological and syntactic information. They are more likely to produce a correct sentence structure, and to handle long-distance movement correctly. But RBMT systems tend to produce translations that are literal, and lacking in fluency. Serial combination attempts to take advantage of the strengths of both RBMT and SMT systems by:

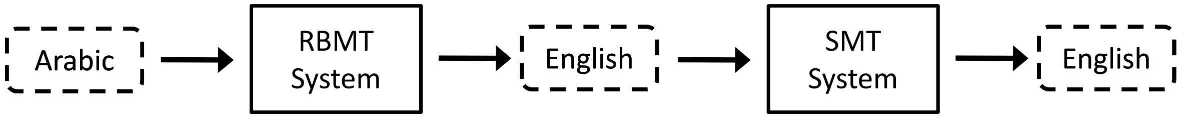


Figure 6-4: Flow Chart of the Serial System Combination Process

- Using the rule-based system to translate the source sentences to the target language, thus producing an overall sentence structure.
- Then using a statistical MT system to transform the output of the RBMT system, while remaining in the same language. The corrections in this case will tend to be local.

The second stage can be seen as a statistical post-editing step, where the SMT component can learn to correct systematic patterns in the output of the RBMT system. Also, the SMT component is constrained by a language model in the target language, which yields more fluent translations in the final output.

This method was first introduced in (Simard et al., 2007a) and (Simard et al., 2007b) for English-to-French and French-to-English translation. The SMT component in that work is trained on the output of the RBMT system and manually post-edited versions of that output.

6.4.1 Arabic-to-English Serial Combination

The serial combination method was also implemented for Arabic-to-English translation, using the Sakhr rule-based MT system, and the Hierdec system for statistical post-processing. Table 6.4 presents results for the serial combination of the Sakhr rule-based system and Hierdec. The system was trained on 30 million words from the Sakhr Arabic-English Parallel Corpus (SSUSAC27). The Hierdec system uses the 7

	TER lc	BLEU lc	BL-Pr lc	MET
	Test1.ara.text.nw			
Hierdec	**37.81**	**55.51**	**55.79**	**70.22**
Serial Comb.	38.17	54.85	55.00	69.64
	+0.36	-0.66	-0.79	-0.58
	Test2.ara.text.nw			
Hierdec	**39.16**	**53.60**	**54.34**	**68.96**
Serial Comb.	39.67	52.32	52.74	68.10
	+0.51	-1.28	-1.60	-0.86
	Tune.ara.text.nw			
Hierdec	**37.19**	**57.16**	**57.19**	**70.64**
Serial Comb.	38.21	55.63	55.67	69.68
	+1.02	-1.53	-1.52	-0.96

Table 6.4: MT scores of serial combination system

billion word English language model mentioned is section 6.3.2. The same tuning and test sets described in section 5.4 are used. The table shows that the scores for the serial combination system improve tremendously over those of the rule-base system, and come to within around 1 point of the Arabic-to-English Hierdec baseline trained on the same corpus.

These results show that serial combination can produce an end-to-end system with comparable quality to a purely statistical system. In the rest of this section, some thoughts on how to enhance this combination method are presented, together with preliminary results that show promise for that direction.

6.4.2 Enhanced Serial Combination

The statistical component of the serial combination system learns translations between the intermediary form of the English (i.e., the output of the rule-based system) and the English references. Once the original Arabic source is translated through the rule-based system, that source is not taken into account anymore. The rule-base translations can contain errors that are not recoverable by the statistical component.

For instance, if the rule-based system deletes content words, because it does not know how to translate them, those deleted words will be impossible to recover, except in cases where the deletions are systematic enough for the statistical system to learn rules that recover them in specific contexts. Word deletion is only one example of irrecoverable RBMT errors. The incorporation of the original source into the serial combination component's statistical model should result in better translations, since both the original source and intermediate translation would be available to the serial combination component. The remainder of this chapter presents some some thoughts on how serial combination can be improved, and some preliminary results in this direction.

Recall that the fundamental equation for the noisy channel model of SMT for Arabic-to-English is:

$$\Pr(e|a) = \frac{\Pr(e).\Pr(a|e)}{\Pr(a)} \tag{6.3}$$

where a is the Arabic source sentence and e is the English target sentence. The translation task is then formulated as a search problem:

$$\hat{e} = \underset{e}{\text{argmax}} \Pr(e|a) = \underset{e}{\text{argmax}} \Pr(e).\Pr(a|e) \tag{6.4}$$

The rule-based translation of the source sentence (denoted here by e_r) can be incorporated into equation 6.4, and therefore into equation 6.6 as follows:

$$\hat{e} = \underset{e}{\text{argmax}} \Pr(e|a) = \underset{e}{\text{argmax}} \Pr(e).\sum_{e_r} \Pr(a, e_r|e) \tag{6.5}$$

The term $\sum_{e_r} \Pr(a, e_r|e)$ can be approximated by $max \Pr(a, e_r|e)$, if we assume that the other values are much smaller:

$$\hat{e} = \underset{e}{\text{argmax}} \Pr(e|a) = \underset{e}{\text{argmax}} \Pr(e).\Pr(a, e_r|e) \tag{6.6}$$

Applying Bayes rule to $\Pr(a, e_r|e)$, we get:

$$\hat{e} = \underset{e}{\text{argmax}}\Pr(e|a) = \underset{e}{\text{argmax}}\Pr(e). \Pr(e_r|e). \Pr(a|e_r, e) \tag{6.7}$$

Equation 6.7 contains the terms $\Pr(e)$, the English language model and $\Pr(e_r|e)$, the serial combination translation model. It also contains the term $\Pr(a|e_r, e)$, which could be interpreted as a two-source translation model. In other words, a generative process corresponding to this model can be defined where the English target language string e generate the intermediate English sentence e_r (this part of the process corresponds to $\Pr(e_r|e)$), then once the sentence e_r is generated, the Arabic string a is generated through another probabilistic process from the two sentences e and e_r. The estimation of the parameters of the model $\Pr(a|e_r, e)$ can be done by generalizing the Expectation Maximization (EM) algorithm usually used to estimate the translation probabilities and alignments of word pairs. With such an alignment model, one can generalize the phrase extraction procedure of phrase-based systems and the corresponding hierarchical rules to extract rules that translate from the two sources to the target language.

The generalization of the alignment procedure in this way would be the preferred approach in terms of modeling generality and flexibility. It would, however, be quite an involved undertaking, especially in terms of implementation. One simplifying assumption that can be made is:

$$\Pr(a|e_r, e) = \Pr(a|e) \tag{6.8}$$

In other words, the assumption is that the knowledge of the intermediate sentence e_r does not affect the probability of generating the source sentence from the target sentence e. This is not an entirely baseless assumption, since a deterministic mapping exists between a and e_r, namely the rule-base translation. Under the independence assumption, equation 6.7 can be rewritten as:

$$\hat{e} = \underset{e}{\text{argmax}} \Pr(e|e_r, a) = \underset{e}{\text{argmax}} \Pr(e). \Pr(e_r|e). \Pr(a|e) \tag{6.9}$$

The tree terms of this equation are the English Language model, the alignment probabilities between the intermediate source and the English target, and the alignment probabilities between the Arabic source and the English target. Each of these two alignment sets can be obtained using a regular word alignment procedure (i.e., GIZA++ (Och and Ney, 2003)). Instead of requiring a new model, and a corresponding new training algorithm, the integration of the two standard models, together with other features, can be done during decoding. Equation 6.9 can be seen as defining the optimization criterion for the decoding of a two-source sentence (a, e_r). The multi-source decoding can be done in various ways, depending on the translation method. For phrase-based SMT, two phrase tables can be extracted from the two alignment sets. A generalization of hierarchical decoding can also be defined in this way. The two phrase tables can be used to extract two rule sets: $\mathbf{R_1} : a \rightarrow e$ and $\mathbf{R_2} : e_r \rightarrow e$. Recall that each of these rule sets is a synchronous CFG, and that the hierarchical decoding procedure consists essentially of a parse of the source sentence, and a simultaneous generation of the target hypotheses with corresponding scores. Usually a chart-style bottom-up parse is used, scanning input spans of increasing length. The parsing procedure can be generalized to use two sources, and the two corresponding rule sets, by first aligning the two input string a and e_r, so that for each span $a(i, j)$, a corresponding span $e_r(i', j')$ can be obtained. Applicable rules from both $\mathbf{R_1}$ and $\mathbf{R_2}$ can then be used in the production of the translation hypotheses. Tunable weights can be used to combine the scores from the two rule sets.

We next present a preliminary experiment which integrates the two models in a more superficial way. This work is presented as a preliminary assessment of the potential benefit that could be obtained from a deeper integration along the lines

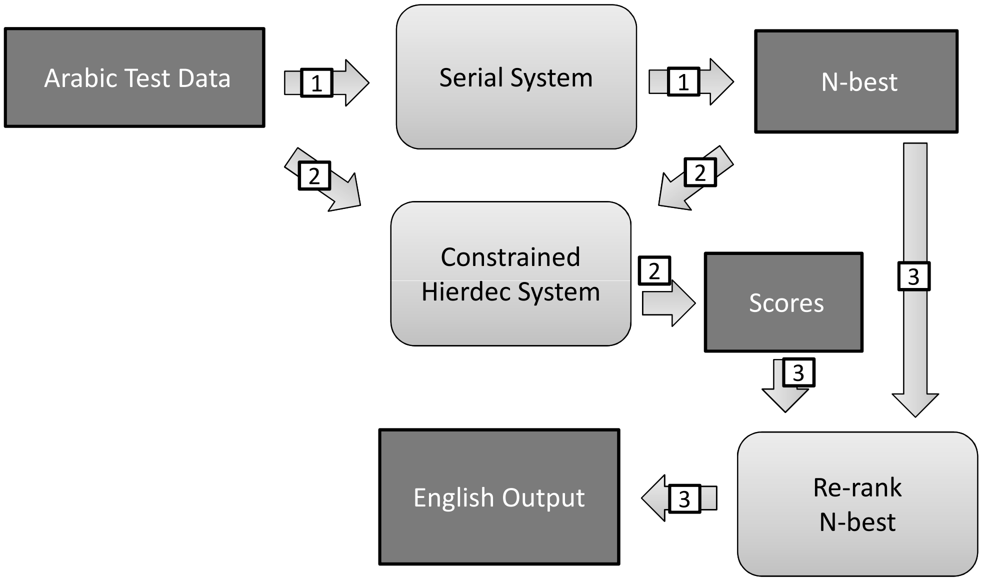


Figure 6-5: Flow Chart of the Enhanced System Combination Process.

described above. In this preliminary experiment, scores from the Hierdec model (i.e. $a \rightarrow e$ translations) are used in the rescoring of the list of N-best hypotheses from the serial combination system ($e_r \rightarrow e$ translations). Figure 6-5 shows this integration procedure. First, the serial combination system is used to obtain a list of N-best hypotheses. Then the Arabic-to-English Hierdec decoder is run in a constrained mode, which attempts to force the decoder to obtain a predefined list of hypotheses. In this case, the constrained output is the N-best list from the serial combination system. A byproduct of the constrained decoding step is to assign scores to the different hypotheses based on the Arabic-to-English Hierdec model. These scores can then be used as additional features in the log-linear model used for rescoring the N-best list. Other rescoring features include a 5-gram language model. It is worth noting that constrained decoding is not always capable of reproducing the required output. Sometimes the system's rule set cannot produce a derivation of that output. In this experiment, only 20% of the serial combination hypotheses were reproduced

	TER **lc**	**BLEU** **lc**	**BL-Pr** **lc**	**MET**
	Test1.ara.text.nw			
Hierdec	**37.81**	**55.51**	**55.79**	**70.22**
Serial Comb.	38.17	54.85	55.00	69.64
	+0.36	-0.66	-0.79	-0.58
Rescored Serial Comb1.	38.12	55.07	55.13	69.88
	+0.31	-0.44	-0.66	-0.34
Rescored Serial Comb2.	37.86	55.38	55.44	70.04
	+0.05	-0.13	-0.35	-0.18
	Test2.ara.text.nw			
Hierdec	**39.16**	**53.60**	**54.34**	**68.96**
Serial Comb.	39.67	52.32	52.74	68.10
	+0.51	-1.28	-1.60	-0.86
Rescored Serial Comb1.	39.53	52.56	52.90	68.28
	+0.37	-1.04	-1.44	-0.68
Rescored Serial Comb2.	39.50	52.77	53.07	68.32
	+0.34	-0.83	-1.27	-0.64
	Tune.ara.text.nw			
Hierdec	**37.19**	**57.16**	**57.19**	**70.64**
Serial Comb.	38.21	55.63	55.67	69.68
	+1.02	-1.53	-1.52	-0.96
Rescored Serial Comb1.	38.05	55.94	55.99	69.93
	+0.86	-1.22	-1.20	-0.71
Rescored Serial Comb2.	37.75	56.09	56.13	70.14
	+0.56	-1.07	-1.06	-0.50

Table 6.5: Results of serial combination rescoring using Hierdec scores.

during constrained decoding.

Table 6.5 shows the results of rescoring the serial system of table 6.4. The row labeled *Rescored Serial Comb1.* refers to experiments that use $\Pr(source|target)$, the backward translation probability from the Hierdec constrained decoding as a rescoring feature. Compared to the results of the serial combination system, rescoring shows a small improvement of 0.2 to 0.3 BLEU points. In the experiments labeled *Rescored Serial Comb2.* additional features are used in rescoring. These consist of $\Pr(target|source)$, the forward translation probability, as well as the lexical smoothing

	TER	**BLEU**	**BL-Pr**	**MET**
	lc	**lc**	**lc**	
	Test1.ara.text.nw			
Hierdec	**37.81**	**55.51**	**55.79**	**70.22**
Serial Comb.	38.17	54.85	55.00	69.64
	+0.36	-0.66	-0.79	-0.58
Rescored Hierdec	37.60	55.81	56.09	70.43
	-0.21	+0.30	+0.30	+0.21
	Test2.ara.text.nw			
Hierdec	**39.16**	**53.60**	**54.34**	**68.96**
Serial Comb.	39.67	52.32	52.74	68.10
	+0.51	-1.28	-1.60	-0.86
Rescored Hierdec	38.77	53.72	54.51	68.94
	-0.39	+0.12	+0.17	-0.02
	Tune.ara.text.nw			
Hierdec	**37.19**	**57.16**	**57.19**	**70.64**
Serial Comb.	38.21	55.63	55.67	69.68
	+1.02	-1.53	-1.52	-0.96
Rescored Hierdec	37.06	57.51	57.59	70.71
	-0.13	+0.35	+0.40	+0.07

Table 6.6: Results of Hierdec rescoring using serial combination.

score from the Hierdec system, and a binary feature on whether the hypothesis has been reproduced by the constrained decoding. The addition of these features shows an additional gain of another 0.2 to 0.3 points.

This particular setup does not require that the rescored system be necessarily the serial combination system, and the constrained decoding system be Hierdec. In fact, a symmetric experiment can be performed where the N-best hypotheses from the Hierdec system are rescored using features from a constrained decoding of the serial combination system. Table 6.6 shows the results of the symmetric experiment, using the same features as in the previous experiment. A small gain of up to 0.3 BLEU points is shown relative to the Hierdec baseline.

The gains obtained from these experiments are rather modest. But these gains were obtained through re-ranking, using translation scores for only 20% of the hypotheses. These preliminary results indicate that further work in the direction of a deeper integration along the lines described earlier in this section have the potential of producing more improvements.

6.5 Summary

This chapter presented some work on combining rule-based and statistical MT systems. The first method presented leveraged the advantages of rule-based and statistical systems respectively, namely a good sentence structure and non-literal translations learned from the training data. The method proposed produced significant gains over the scores of the rule-based baseline, and out-performed the statistical MT baseline for the web data genre. In the second part of the chapter preliminary results were presented on enhancing the serial combination method, together with some thoughts on how this research direction can be further developed.

Chapter 7

Conclusion and Future Work

In this thesis, we have presented several methods that integrate linguistic information into a statistical machine translation framework. We have shown through experimental results that statistical machine translation, characterized by robustness and language-independence, can be further enhanced by applying language specific techniques that make use of the linguistic information. Such techniques can help the statistical model by explicitly addressing aspects of the translation — some of which are a function of the specific language-pair in question — that might be difficult for the statistical models to learn automatically.

The methods presented in the thesis incorporate information at different levels of linguistic abstractions; namely, at the morphological, syntactic, lexical and system levels. The thesis concentrates on translation between Arabic and English, and forms a case study into how specific properties of the languages in question can be leveraged. Some of techniques presented can be applied directly to other language pairs. For others, the details would differ for different languages, but the ideas are still applicable. An example of the first case is the noun-phrase integration method of chapter 6, which can be applied regardless of the language in question if a rule-based system that can produce the appropriate information is available. Syntax reordering (chapter 4) can be in principle be applied to any language pair; but the specific reordering rules will depend on the syntax of the two languages and how they differ.

We have presented experiments on both Arabic-to-English and English-to-Arabic, and shown improvements relative to strong baselines, using relatively large data corpora, and state of the art SMT models such as phrase-based and hierarchical SMT.

The morphological Analysis of Arabic for English-to-Arabic MT in chapter 3, and the syntactic reordering of English in chapter 4 are the first work that concentrates directly on issues specific to English-to-Arabic MT. This direction represents technical challenges that are not present in the opposite direction. The complex morphology of Arabic requires that morphological generation on the Arabic side be handled explicitly, if the problem of sparcity be dealt with. In addition, agreement conditions for gender, number and person in Arabic are stronger than English. This is true for verb conjugation as well as noun-adjective agreement. Producing proper agreement in the Arabic output might also require special modeling, beyond the constraints provided by the language model, since some of the agreement conditions can be long-distance. The importance of MT into Arabic will only grow as the need for translating resources of knowledge from other language for the Arabic world will become more important.

This thesis also presented the first work that used diacritized Arabic for translation from Arabic, by using decision trees to control the increase in the sparsity of the models that result from adding diacritics, while using the information contained in the diacritics that is useful in decreasing the translation probability entropy.

Future Work

The problem of machine translation is far from being solved, and much research and development still needs to be done until machines are able to translate as reliably and fluently as humans do. The demand for faster and cheaper translation between more languages will only increase with the increase in the need to share information between different parts of the globe.

Future work, building on the results in this thesis, can be developed in multiple directions. Clustering as a general approach for dealing with issues of sparsity is promising. The use of decision trees, or other clustering mechanisms, to provide robust estimates for the parameters of various models is likely to result in improvements. In particular, the robust estimation of rule probabilities has the potential of resulting in quite significant gains, as we mentioned in the previous chapter. Oracle experiments suggest that, besides the development of new statistical models, the robust estimation of parameters for models that are currently in use is a fruitful line of research.

The shift into the statistical paradigm that has occurred over the past couple of decades has allowed the field to make dramatic advances, which will likely only increase with the increase in the availability of computational and linguistic resources. But as the improvements from current approaches reach their limit, the focus will likely turn more to utilizing explicit linguistic insight, in combination with statistical methods. So far, most of the work on integrating the statistical and rule-based approaches has been relatively shallow. A tighter integration of the two approaches is likely to produce results that exceed the state-of-the-art in either. One example of such integration is to estimate appropriate probability models over the various linguistic resources available to the rule-based system, such as lexicons, parsing rules, or transfer rules.

The interest in Arabic MT will likely remain high in the foreseeable future, due to political and social conditions in the Arabic world. Many issues that are specific to Arabic translation remain to be addressed. We discuss some of them next with examples.

Translation from Arabic can benefit significantly from better source analysis, especially if the syntactic structure of the Arabic source is taken into consideration.

(Shen et al., 2009) experimented with using syntactic features on the source side, without success. The authors hypothesize that this is likely due to the bad quality of existing Arabic parsers. Reliable parsers will therefore be essential for the success of such efforts.

Specific cases in which Arabic MT might benefit from Arabic syntax analysis include the effect of the Arabic sentence order, combined with subject-verb morphology, on translation. Arabic is a prodrop language, allowing a null subject. So, in principle, a noun phrase after a verb can be either a subject or an object with a null subject in between. Consider In the following example:

(7.1) **Source:** w yHAwl rwbyr mrAwgp AlstTAt bAlwSwl fj>p <lY AlmkAn

MT: He tries to Robert maneuvering the authorities to suddenly reach the place

Reference: Robert tries to elude the authorities by quickly gaining access to the place

An additional subject pronoun has been inserted, likely due to the alignment of the verb "yHAwl" to "he tries" in other occurrences of the two words in the training corpus. The system does not take into consideration the existence of an explicit subject "rwbyr" *'Robert'* in the source.

Analysis of the grammatical role of each of the constituents could be used to inform the translation. Another example of errors is Null Complementizers (that is the equivalent of subordinate conjunction that is not phonetically explicit), a common phenomenon in Arabic. In these errors, these are not supplied in the output translation, as the following example shows:

(7.2) **Source:** mhmp lyst shlp tntZr Eml Alhy}p Altnsyqyp lAEAdp

MT: The task is not easy waiting for the work of the coordinating body

Reference: It is not an easy mission **which** awaits the work of the Coordination Commission

The next example shows an interrogative sentence, where the system fails to produce the fronted auxiliary verb in the English translation. This is another case where source-side syntactic analysis can benefit translation quality.

(7.3) **Source:** lmA*A A$Ad Alkvyrwn bmwqf AlsEwdyp?

MT: Why many praised the position of Saudi Arabia?

Reference: Why did many praise the position of Saudi Arabia?

Arabic names, especially names of persons, are often regular words (mostly adjectives and nouns). Name detection could be used to avoid translating the literal meaning of names, as is the case in the following example.

(7.4) **Source:** wmn byn Al$hdA' **rA}d** fnwnp

MT: Among the martyrs a **major,**

Reference: Among the martyrs **Ra'id** Fannunah

Another area of interest is the translation of Arabic dialects. This area has started gaining increased attention lately, since it presents a number of challenges. The absence of a standardized orthography of the dialects means that users often improvise spellings for words, resulting in a lot of inconsistencies. Another challenge, also partly due to the absence of standardization, is the wide variation within the dialects. The fact that most written resources in Arabic are in Modern Standard Arabic means that the amount of data available in the dialects is quite limited. Corpora on the order of tens of millions of words, such as those used for translating of MT will be very difficult — if at all possible — to collect. This forms an incentive for developing methods that can compensate for the lack in volume by relying more on linguistic knowledge. A relatively large overlap between MSA and the dialects exists, especially in the written form, due to the fact that short vowels are not specified in either case. An interesting research direction would then be the adaptation of the existing MSA models to the dialects.

References

A. Aho and J. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56.

S. Ananthakrishnan, S. Narayanan, and S. Bangalore. 2005. Automatic diacritization of Arabic transcripts for automatic speech recognition. Kanpur, India.

E. Avramidis and P. Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.

I. Badr, R. Zbib, and J. Glass. 2008. Segmentation for English-to-Arabic statistical machine translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 153–156, Columbus, Ohio, June.

I. Badr, R. Zbib, and J. Glass. 2009. Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 86–93.

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *In Proc. of ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, Michigan.

K. Beesley. 2001. Finite-state morphological analysis and generation of Arabic at Xerox research: Status and plans in 2001. In *EACL 2001 Workshop Proceedings on Arabic Language Processing: Status and Prospects*, Toulouse, France.

K. Belnap and N. Haeri. 1997. *Structuralist studies in Arabic linguistics : Charles A. Ferguson's papers, 1954-1994.* Leiden ; New York: E.J. Brill.

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

R. Brent. 1973. *Algorithms for Minimization Without Derivatives.* Prentice-Hall.

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

P. Brown, V. Della Pietra, S. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

J. Brunning, A. de Gispert, and W. Byrne. 2009. Context-dependent alignment models for statistical machine translation. In *NAACL '09: Proceedings of the 2009 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 110–118.

T. Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

C. Cabezas and P. Resnick. 2005. Using WSD techniques for lexical selection in statistical machine translation. In *Technical report, Institute for Advanced Computer Studies (CS-TR-4736, LAMP-TR-124, UMIACS-TR-2005-42)*, College Park, MD.

J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey. 2006. Context-based machine translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas.*

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *EMNLP-CoNLL: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic.

Y. Chan, H. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL).*

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics.*

S. Chen, S. Chen, and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4).

Y. Chen, M. Jellinghaus, A. Eisele, Y. Zhang, S. Hunsicker, S. Theison, C. Federmann, and H. Uszkoreit. 2009. Combining multi-engine translations with Moses. In *Proceedings of the Fourth Workshop on Statistical Machine Translation.*

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL05.*

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).

K. Church and E. Hovy. 1993. Good applications for crummy machine translation. *Machine Translation*, 8(4):239–258.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 531–540.

M. Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proc. 35th Annual Meeting of the Association for Computational Linguistics.*

M. Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania.

B. Cowan, I. Kučerová, and M.l Collins. 2006. A discriminative model for tree-to-tree translation. In *EMNLP*.

A. de Gispert, D. Gupta, M. Popović, P. Lambert, J. B. Mari no, M. Federico, H. Ney, and R. Banchs. 2006. Improving statistical word alignments with morpho-syntactic transformations. In *Proceedings of 5th International Conference on Natural Language Processing, FinTAL'06*, pages 368–379.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Socierty Series B*, 39(1):1–38.

J. DeNero and D. Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL).*

J. Devlin. 2009. Lexical features for statistical machine translation. Master's thesis, University of Maryland, December.

M. Diab, M. Ghoneim, and N. Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *MT Summit XI*, pages 143–149, Copenhagen, Denmark.

B. Dorr, P. Jordan, and J. Benoit. 1999. A survey of current paradigms in machine translation. *Advances in Computers*, 49:2–68.

R. O. Duda, P. E. Hart, and D. G. Stork. 2000. *Pattern Classification.* Wiley-Interscience Publication.

C. Dyer, S. Muresan, , and P. Resnik. 2008. Generalizing word lattice translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL).*

A. Eisele, C. Federmann, H. Saint-Amand, M. Jellinghaus, T. Herrmann, and Y. Chen. 2008. Using Moses to integrate multiple rule-based machine translation engines into a hybrid system. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation.*

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, Sapporo, Japan, July.

C. A. Ferguson. 1959. Diglossia. *Word*, 15.

D. Filimonov and M. Harper. 2009. A joint language model with fine-grain syntactic tags. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing*, Morristown, NJ. Association for Computational Linguistics.

J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*, University of Michigan, USA".

C. S. Fordyce. 2007. Overview of theIWSLT 2007 evaluation campaign. In *International Workshop on Spoken Language Translation*, Trento, Italy.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *ACL*, pages 80–87.

K. Gimpel and N. A. Smith. 2008. Rich source-side context for statistical machine translation. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17, Columbus, Ohio.

S. Goldwater and D. McClosky. 2005. Improving statistical MT through morphological analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 676–683.

N. Habash and O. Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL).*

N. Habash and O. Rambow. 2007. Arabic diacritization through full morphological tagging. In *Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 53–56, Rochester, New York.

N. Habash and F. Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics.*

N. Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of the Machine Translation Summit (MT-Summit).*

N. Haeri. 2000. Form and ideology: Arabic sociolinguistics and beyond. *Annual Review of Anthropology*, 29.

D. Hakkani-Tür, K. Oflazer, and G. Tür. 2000. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of the 18th International Conference on Computational Linguistics.*

P. E. Hart, N. J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4.

L. Huang and D. Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Vancouver, British Columbia. Association for Computational Linguistics.

L. Huang, K. Knight, and A. Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, Boston, MA.

L. Hyafil and R. Rivest. 1976. Constructing optimal binary decision trees is np-complete. *Information Processing Letters.*

F. Jelinek. 1997. *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, USA.

D. Jurafsky and J. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence).* Prentice Hall.

S. M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 400–401.

G. Kiraz, 2001. *Studies in Natural Language Processing*, chapter Computational Non-linear Morphology with Emphasis on Semitic Languages. Cambridge University Press.

D. Klein and C. Manning. 2001. Parsing and hypergraphs. In *In IWPT.*

K. Knight. 1997. Automating knowledge acquisition for machine translation. *AI Mag*, 18(4):81–96.

P. Koehn and H. Hoang. 2007. 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.

P. Koehn and K. Knight. 2003. Feature-rich statistical translation of noun phrases. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July. Association for Computational Linguistics.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

P. Koehn, H. Hoang, and A. Birch. 2007. 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, June.

P. Koehn. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*.

P. Koehn. 2004b. Statistical significance tests for machine translation evaluation. In *EMNLP04*, Barcelona, Spain.

P. Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

K. Koskenniemi. 1983. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki, Department of General Linguistics.

Y. S. Lee, K. Papineni, and S. Roukos. 2003. Language model based Arabic word segmentation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Y. S. Lee. 2004. Morphological analysis for statistical machine translation. In *HLT-NAACL '04: Proceedings of HLT-NAACL 2004*.

D. Lin. 2004. A path-based transfer model for machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*.

A. Lopez and P. Resnik. 2005. Improved hmm alignment models for languages with scarce resources. In *ACL Workshop on Building and Using Parallel Texts*.

A. Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3).

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *In NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.

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

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *EMNLP02*.

D. Marcu, W. Wang, A. Echihabi, and K. Kevin. 2006. Spmt: statistical machine translation with syntactified target language phrases. In *EMNLP06*.

R. Nelken and S. M. Shieber. 2005. Arabic diacritization using weighted finite-state transducers. In *Proceedings of the 2005 ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan.

S. Nießen and H. Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2).

F. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, Hong Kong.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL02*.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28.

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. R. Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*, pages 161–168.

F. Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan.

J. Odell. 1995. *The Use of Context in Large Vocabulary Speech Recognition*. Ph.D. thesis, Cambridge University Engineering Department.

J. Olive. 2005. Global autonomous language exploitation (gale). *DARPA/IPTO Proposer Infomation Pamphlet*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA.

M. Popović and H. Ney. 2004. Towards the use of word stems and suffixes for statistical machine translation.

M. Popović and H. Ney. 2006. Pos-based word reordering for statistical machine translation. In *NAACLE LREC*, Philadelphia, PA.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1nd Conference of Empirical Methods in Natural Language Processing*, pages 133–142, Philadelphia, PA.

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? 88(8).

A. I. Rosti, S. Matsoukas, and R. Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic.

A. I. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz. 2008. Incremental hypothesis alignment for building confusion networks with applicatoin to machine translation system combination. In *Proceedings of the Third Workshop on Statistical Machine Translation*, Columbus, Ohio.

A. I. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz. 2010. BBN system description for WMT10 system combination task. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden.

F. Sadat and N. Habash. 2006. Combination of Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL).*

R. Sarikaya and Y. Deng. 2007. Joint morphological-lexical language modeling for machine translation. In *Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–148.

L. Shen, J. Xu, and R. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio.

L. Shen, J. Xu, B. Zhang, S. Matsoukas, and R. M. Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing.*

M. Simard, C. Goutte, and P. Isabelle. 2007a. Statistical phrase-based post-editing. In *Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, NY.

M. Simard, N. Ueffing, P. Isabelle, and P. Kuhn. 2007b. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

M. Sipser. 2005. *Introduction to the Theory of Computation.* Course Technology.

M. Snover, B. Dorr, R. Schwartz, J. Makhoul, and L. Micciulla. 2006. A study of translation error rate with targeted human annotation. In *Proceedings of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA 2006)*, pages 223–231, Cambridge, MA.

N. Stroppa, A. van den Bosch, and A Way. 2007. Exploiting source similarity for SMT using context-informed features. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pages 231–240.

G. Thurmair. 2005. Hybrid architectures for machine translation systems. *Language Resources and Evaluation*, pages 91–108.

G. Thurmair. 2009. Comparing different architectures of hybrid machine translation systems. In *MT Summit XII: Proceedings of the twelfth Machine Translation Summit*, Ottawa, Ontario, Canada.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*, Edmonton, Canada.

V. Vandeghinste, I. Schuurman, M. Carl, S. Markantonatou, and T. Badia. 2006. METIS-II: Machine translation for low resource languages. In *Proceedings of LERC*, Genoa, Italy.

D. Vergyri and K. Kirchhoff. 2004. Automatic diacritization of Arabic for acoustic modeling in speech recognition. In *Semitic '04: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland.

D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 2005. Word-sense disambiguation for machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouser, BC, Canada.

S. Vogel, F. J. Och, and H. Ney. 2000. The statistical translation module in the verbmobil system. In *Proceedings of KONVENS Ilmenau.*

C. Wang, M. Collins, and P. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 737–745, Prague, Czech Republic, June.

D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *ACL96*, pages 152–158.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

F. Xia and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING '04: The 20th Int. Conf. on Computational Linguistics*, page 508.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL).*

S.J. Young, J.J. Odell, and P.C. Woodland. 1994. Tree-based state tying for high accuracy acoustic modelling. In *HLT'94: Proceedings of the Workshop on Human Language Technology*, pages 307–312.

R. Zbib, S. Matsoukas, R. Schwartz, and J. Makhoul. 2010. Decision trees for lexical smoothing in statistical machine translation. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden.

R. Zens, F. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *KI - 2002: Advances in artificial intelligence*, volume LNAI 2479, pages 18–32. Springer Verlag.

I. Zitouni, J. S. Sorensen, and R. Sarikaya. 2006. Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia.