Robust Machine Translation for Multi-Domain Tasks

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von
Diplom-Informatiker Oliver Bender
aus Heinsberg

Berichter: Prof. Dr.–Ing. Hermann Ney
Prof. Dr. Francisco Casacuberta

Tag der mündlichen Prüfung: 11. März 2010

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

To Gaby

Abstract

In this thesis, we investigate and extend the phrase-based approach to statistical machine translation. Due to improved concepts and algorithms, the quality of the generated translation hypotheses has been significantly improved in recent years. Still, the translation quality leaves a lot to be desired when going beyond traditional translation tasks, such as newswire articles, and when addressing more ambitious translation problems. We extend the state-of-the-art in phrase-based translation which enables us to build a robust translation system for multi-domain input. Robustness is hereby regarded as the ability to produce high quality translations for arbitrary input texts, e.g. automatic transcriptions of recognized speech or other unstructured, potentially noisy input. In this work, we focus on Arabic-English translation tasks.

We study the search problem for phrase-based statistical machine translation in detail. For this, we examine the effect of the different models on the translation quality. Moreover, we make an explicit distinction between reordering (coverage) and lexical hypotheses in the pruning process and stress the importance of the coverage pruning to adjust the balance between hypotheses representing different reorderings (coverage hypotheses) and hypotheses with different lexical representations. We present constraints to solve the reordering problem in machine translation.

To trim our translation system for multi-domain input and to improve the robustness built into the decoder, we apply domain adaptation to the language models and rerank the candidate translations using appropriate rescoring models. We also present our work on adjusting the vocabularies of the speech recognizer and the machine translation system in a preprocessing step and on predicting missing punctuation marks for automatically transcribed speech (in the actual translation process).

Processing morphologically rich languages such as Arabic generally poses high demands on preprocessing. We show that the choice of the appropriate preprocessing strategy depends on the translation domain and on the structure of the input data. Experimental results emphasize how the proper choice of the preprocessing approach helps to increase the translation quality.

In addition, we address the task of improving the translation quality by means of syntactically motivated feature functions within a reranking concept. Then, we investigate different data-driven approaches to the task of transliterating proper names. Often, such names are out-of-vocabulary terms and the intention is to preserve the names by transliteration. Finally, we show how human translators can be assisted by machine translation systems. We compare search strategies for interactive machine translation.

The presented machine translation system achieves state-of-the-art performance and has been successfully applied to the large-scale Arabic-English GALE translation evaluations. Furthermore, the system was ranked among the top submissions for the NIST Open Machine Translation Evaluation 2006 and for the series of IWSLT evaluation campaigns.

Kurzfassung

In dieser Arbeit untersuchen und erweitern wir den phrasenbasierten Ansatz zur maschinellen Übersetzung. Dank verbesserter Konzepte und verfeinerter Algorithmen konnte die Qualität der generierten Übersetzungen in den letzten Jahren deutlich verbessert werden. Die Übersetzungsqualität lässt dennoch zu wünschen übrig, geht man von traditionellen Aufgabenstellungen wie der Übersetzung von Zeitungsartikeln zu anspruchsvolleren Problemen über. Ziel dieser Arbeit ist, den aktuellen Stand der Technik in der phrasenbasierten Übersetzung zu verbessern und ein Übersetzungssystem zu entwickeln, welches robust ist und mehrere Domänen unterstützt. Der Fokus liegt hierbei auf Aufgabenstellungen zur Übersetzung aus dem Arabischen ins Englische. Unter Robustheit verstehen wir die Fähigkeit, treffende Übersetzungen auch für Transkriptionen automatisch erkannter Sprache und andere, potentiell verrauschte, Eingabedaten zu liefern.

Wir beschreiben und analysieren das Suchproblem der phrasenbasierten, statistischen Übersetzung in allen Einzelheiten. Hierzu untersuchen wir den Effekt der einzelnen Modelle auf die Qualität der Übersetzungen. Zudem treffen wir eine explizite Unterscheidung zwischen Umordnungs- (Abdeckungs-) und lexikalischen Hypothesen während des Prunings. Wir heben die Bedeutung des Prunings der Abdeckungshypothesen hervor, um die Anzahl an Hypothesen zu steuern, die unterschiedliche Wortstellungen (Abdeckungshypothesen) und unterschiedliche lexikalische Darstellungen repräsentieren. Wir zeigen Einschränkungen, die das Umordnungsproblem in der maschinellen Übersetzung lösen.

Um unser Übersetzungssystem an mehrfache Domänen anzupassen und um die Robustheit des System zu verbessern, adaptieren wir die Sprachmodelle an die jeweilige Domäne. Mit Hilfe geeigneter Modelle bewerten wir die Hypothesen ein weiteres Mal und aktualisieren die ausgewählten Übersetzungen. Zudem stellen wir unsere Arbeiten vor, die die Vokabularien des Spracherkenners und des Übersetzungssystems angleichen, und Interpunktionszeichen vorhersagen, die in den automatischen Transkriptionen fehlen.

Generell stellt die Verarbeitung morphologisch reicher Sprachen besondere Anforderungen an die Vorverarbeitung der Daten. Wir zeigen, dass die Wahl einer geeigneten Strategie für diese Vorverarbeitung von der Domäne und der Charakteristik der Eingabedaten abhängt. Experimentelle Untersuchungen verdeutlichen, wie die Wahl der richtigen Vorverarbeitungsmethode zur Verbesserung der Übersetzungsqualität beitragen kann.

Ferner befassen wir uns mit der Aufgabenstellung, die Übersetzungsqualität mit Hilfe von syntaktisch motivierten Feature-Funktionen zu verbessern. Ein weiterer Aspekt ist die Untersuchung verschiedener Ansätze zur Transliteration von Eigennamen, da diese dem Übersetzungssystem häufig unbekannt sind. Schließlich befassen wir uns mit dem Bereich der interaktiven Übersetzung und vergleichen Suchstrategien für den Einsatz in interaktiven Systemen.

Das in dieser Arbeit beschriebene System erzielt Ergebnisse, die mit den besten, zur Zeit veröffentlichten Ergebnissen vergleichbar sind. Es wurde im Rahmen der GALE-Evaluationen für die Übersetzungsaufgaben vom Arabischen ins Englische erfolgreich eingesetzt. Des Weiteren gehörte das System zu den besten Systemen bei der "NIST Open Machine Translation Evaluation 2006" sowie für eine Reihe von IWSLT-Evaluationen.

Acknowledgments

At this point, I would like to express my gratitude to all the people who supported and accompanied me during the progress of this work.

First, I would like to thank Hermann Ney for supervising me during the last years and for offering doing research in this interesting and challenging area. In particular, I want to thank for all the opportunities he gave me.

I would also like to thank Francisco Casacuberta from the Universidad Politecnica de Valencia for agreeing to review this thesis and for his interest in this work.

Next, all my colleagues at our lab deserve my gratitude for many fruitful discussions, helpful feedback, and for the very good working atmosphere. Some of you became real friends: Saša, Thomas, Christian, David, Philippe, and Björn. Special thanks go to the SMT group: Richard, Evgeny, Shahram, Arne, Gregor, Saab, Daniel, Yuqi, Maja, Jia, Stefan, and Patrick. Also to some of the "old guys": Nicola, András, Klaus, Wolfgang, Daniel, and Franz. Furthermore, I would like to thank the secretaries for their continuous support.

I am very thankful for the friendly atmosphere and the support I received at SRI International's Speech Technology and Research (STAR) Laboratory, Menlo Park, CA during my stay in 2007. It was a very interesting and valuable experience.

Next, I would like to thank my parents for supporting me and giving me all the chances I had. Thank you to Esther and Kirsten for always encouraging me but also reminding me that there are other things than this work.

Last but not least, this work would not have been possible without Gaby. I have to thank for her love, encouragement, never-ending patience, and so much more. I also want to thank Leo for just being there and for pushing me to write down this thesis.

Contents

1 Introduction **1**
1.1 Statistical Machine Translation . . . 2
1.1.1 Direct Translation Model . . . 3
1.1.2 Phrase-Based Approach . . . 4
1.2 Robust Machine Translation for Multi-Domain Tasks . . . 6
1.3 Related Work . . . 8

2 Scientific Goals **11**

3 Morpho-Syntactic Arabic Preprocessing **15**
3.1 Motivation . . . 15
3.2 Tokenization and Normalization . . . 16
3.3 Word Segmentation . . . 16
3.3.1 Finite State Automaton-Based Approach . . . 17
3.3.2 Word Segmentation Derived From Full Morphological Disambiguation . . . 19
3.3.3 Using POS Tags to Infer Segmentation . . . 21
3.4 Summary . . . 21

4 Search for Phrase-Based Translation **23**
4.1 Motivation . . . 23
4.2 Models Used During Search . . . 24
4.3 Search Algorithms and Pruning Strategies . . . 27
4.3.1 Monotone Search . . . 28
4.3.2 Non-Monotone Search . . . 29
4.3.3 Pruning . . . 29
4.4 Reordering Constraints . . . 33
4.4.1 IBM Constraints . . . 33
4.4.2 ITG Constraints . . . 33
4.5 Generation of Word Graphs and n-Best Lists . . . 34
4.6 Summary . . . 36

5 Reranking Translation Hypotheses Using Structural Properties **37**
5.1 Motivation . . . 37
5.2 Reranking Framework . . . 38
5.2.1 Standard Rescoring Models . . . 38
5.2.2 Supertagging With Lightweight Dependency Analysis (LDA) . . . 39
5.2.3 Link Grammars . . . 40
5.2.4 Maximum Entropy Based Chunking . . . 41
5.3 Summary . . . 42

6 Robustness and Multi-Domains **43**
6.1 Motivation . . . 43
6.2 Domain Adapted Language Models . . . 44
6.2.1 Implementation . . . 44
6.2.2 Results . . . 44
6.3 Adjustment of ASR and SMT Vocabularies . . . 45
6.4 Punctuation Prediction . . . 46
6.5 Summary . . . 47

7 Transliteration of Proper Names **49**
7.1 Motivation . . . 49
7.2 Data-Driven Approaches to Arabic Name Transliteration . . . 50
7.2.1 Phrase-Based Machine Transliteration . . . 50
7.2.2 Transliteration as Grapheme-To-Phoneme Conversion . . . 51
7.2.3 Maximum Entropy Models for Transliteration . . . 53
7.2.4 Conditional Random Fields for Transliteration . . . 55
7.2.5 A Deep Learning Approach to Transliteration . . . 56
7.3 Application Within a System Combination Framework . . . 58
7.4 Summary . . . 59

8 Search Strategies for Interactive Machine Translation **61**
8.1 Motivation . . . 61
8.2 Interactive Machine Translation . . . 62
8.3 Phrase-Based Approach . . . 62
8.3.1 Generation . . . 63
8.4 Interactive Generation . . . 63
8.5 Interactive Generation With Word Graphs . . . 63
8.5.1 Combination of Both Strategies . . . 65
8.6 Summary . . . 65

9 Experimental Results **67**
9.1 Translation Tasks . . . 67
9.1.1 Training . . . 67
9.1.2 Evaluation Criteria . . . 69
9.1.3 Task Descriptions and Corpus Statistics . . . 70
9.1.4 Comparison With Other Research Groups . . . 76
9.1.5 Comparison of Different Preprocessing Approaches . . . 77
9.1.6 Effect of Different Heuristics for Alignment Symmetrization . . . 80
9.1.7 Analysis of the Search . . . 81
9.1.8 Reranking Experiments . . . 93
9.1.9 Speech Translation Experiments . . . 97
9.1.10 Comparison of Search Strategies for Interactive Machine Translation . . . 100
9.2 Transliteration Tasks . . . 102
9.2.1 Evaluation Criteria . . . 103
9.2.2 Task Descriptions and Corpus Statistics . . . 104
9.2.3 Comparison of Different Approaches to Arabic Name Transliteration . . . 104
9.3 Summary . . . 108

10 Scientific Contributions **109**

11 Conclusions **113**
11.1 Outlook 113

A Symbols and Acronyms **115**
A.1 Mathematical Symbols 115
A.1.1 Mathematical Symbols Used for Translation 115
A.1.2 Mathematical Symbols Used for Transliteration 116
A.2 Acronyms 118

List of Figures **121**

List of Tables **123**

Bibliography **125**

Chapter 1

Introduction

> *"Machine translation (MT) is the application of computers to the task of translating texts from one natural language to another. One of the very earliest pursuits in computer science, MT has proved to be an elusive goal, but today a number of systems are available which produce output which, if not perfect, is of sufficient quality to be useful in a number of specific domains."*
>
> *"The Internet has proven to be a huge stimulus for MT, with hundreds of millions of pages of text and an increasingly global – and linguistically diverse – public. What role will MT play in bridging languages barriers in cyberspace? Stay tuned..."*
>
> (The European Association for Machine Translation – EAMT, 2004)[1]

The quoted statements from the European Association for Machine Translation (EAMT) can be read as a concise introduction to this thesis. Machine translation (MT) is defined as the task of automatically translating a text from one natural language to another. Although research in the field of MT goes back to the 1950's, MT is still an open problem today. At the same time, there exist a number of systems which are capable of generating useful translations for specific domains, e.g. translation services in the travel domain as investigated in the series of *International Workshop on Spoken Language Translation (IWSLT)* [Akiba & Federico$^+$ 04, Eck & Hori 05, Paul 06, Fordyce 07], computer-assisted translation (CAT)[2] systems as addressed by the *TransType2* project [SchlumbergerSema S.A. & Intituto Tecnológico de Informática$^+$ 01], and so on. Moreover, the current and steadily growing demand for MT is exemplarily motivated by the success of the World Wide Web. Since independent from your mother tongue, most of the information contained in the web is expressed in a foreign language, MT is a key technology to resolve the language barrier.

Historically, MT systems are distinguished according to the level of linguistic analysis that is performed. Figure 1.1 shows the standard pyramid visualization of the three levels: direct translation approach, transfer approach and interlingua approach.

The *direct translation approach* does not perform any kind of linguistic analysis. Translations from source language to target language are generated word by word. In the *transfer approach*, the translation process is decomposed into the three steps analysis, transfer and generation. First, the source sentence is analyzed syntactically and semantically in order to produce an abstract representation of the input sentence. This representation is then *transferred* into a corresponding representation of the target language. Finally, the generation step produces the target sentence from the intermediate representation. In the *interlingua approach*,

[1] From: `http://www.eamt.org/mt.php`.

[2] In some publications, the term computer-aided translation is used.

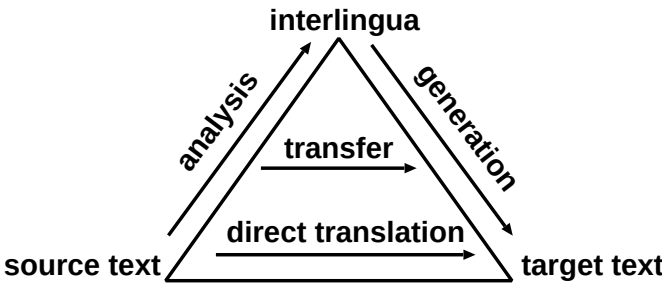

Figure 1.1. Levels of linguistic analysis in an MT system.

an ever deeper analysis produces a completely language independent representation of the input sentence which is used to generate the target language sentence.

Another characteristic is whether the MT system is rule-based or data-driven. In the *rule-based* approach, human experts specify a set of rules which describe the translation process. Obviously, this involves a linguistic analysis of the source text. To capture the phenomena of natural language, a large number of rules is needed which makes the rule-based approach very time consuming and difficult to maintain. Nevertheless, the rule-based approach is predominant in existing textbooks on MT [Hutchins & Somers 92, Dorr 93, Arnold & Balkan$^+$ 94].

Data-driven MT exploits bilingual and monolingual text corpora as main knowledge source and typically follows the direct translation or transfer approach. Here, we often further distinguish between the example-based and the statistical approach to MT. The basic idea of *example-based* MT (EBMT) is to translate by analogy, i.e. a translation is composed of similar translation examples. In the *statistical* approach, MT is treated as a decision problem: given the source language sentence, we have to decide for the target language sentence that is the most probable translation.

In this thesis, we follow the statistical approach to MT. Statistical MT (SMT) is built on statistical decision theory which provides a sound framework for combining several knowledge sources into a single decision criterion with the goal of minimizing the number of errors. The translation of natural languages fits nicely into this framework. Furthermore, SMT systems achieved the best results in recent evaluations, such as the NIST Open MT[3] or IWSLT[4] evaluation campaigns.

1.1 Statistical Machine Translation

The beginnings of statistical MT (SMT) can be traced back to the early 1950's closely related to the work on information theory and cryptography. [Weaver 55] proposed to use an information theoretic approach to MT. Several research projects were set up, however, the problem turned out to be more complicated than expected. As a consequence, funding for MT research was vastly reduced. Increased computing power and the availability of bilingual text corpora, the Canadian Hansards, re-awoke the interest in SMT in the late 1980's. The fundamentals of nowadays SMT systems were published by the IBM Yorktown Heights group [Brown & Cocke$^+$ 88, Brown & Cocke$^+$ 90, Brown & Della Pietra$^+$ 93].

[3]NIST Open Machine Translation (MT) Evaluation: `http://www.nist.gov/speech/tests/mt/`.
[4]IWSLT 2007 International Workshop on Spoken Language Translation: `http://iwslt07.itc.it/index.html`.

1.1.1 Direct Translation Model

In SMT, we are given a source language ("French") sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language ("English") sentence $e_1^I = e_1 \dots e_i \dots e_I$. Among all possible target language sentences, we choose the sentence with the highest probability:[5]

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\}. \tag{1.1}$$

Equation 1.1 is the so-called maximum-a-posteriori (MAP) decision rule, i.e. we select the translation hypothesis which maximizes the posterior probability $Pr(e_1^I | f_1^J)$. The argmax operation denotes the search problem, i.e. how to find the translation with the highest probability among all possible target language sentences.

In the fundamental work on SMT, the posterior probability was decomposed:

$$Pr(e_1^I | f_1^J) = \frac{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)}{Pr(f_1^J)}, \tag{1.2}$$

arriving at:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \tag{1.3}$$

which is referred to as the source-channel approach to SMT [Brown & Cocke$^+$ 90]. Equation 1.3 is also called "the fundamental equation of statistical machine translation" [Brown & Della Pietra$^+$ 93]. Through the decomposition, two knowledge sources are obtained which can be modeled independently. The (target) language model $Pr(e_1^I)$ describes the well-formedness of the target language sentence. The translation model $Pr(f_1^J | e_1^I)$ links the source sentence to the target sentence. In the field of pattern recognition, the source-channel approach has a long history [Duda & Hart 73].

As an alternative approach, many current SMT systems directly model the posterior probability $Pr(e_1^I | f_1^J)$ which was originally proposed by [Papineni & Roukos$^+$ 97, Papineni & Roukos$^+$ 98] for a natural language understanding task. Using a log-linear model for SMT was first proposed by [Och & Ney 02]. Based on the maximum entropy framework [Berger & Della Pietra$^+$ 96], we have a set of M models $h_m(e_1^I, f_1^J)$ and associated model scaling factors λ_m. The direct translation model is then given by:

$$Pr(e_1^I | f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum\limits_{e'^{I'}_1} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, f_1^J)\right)}. \tag{1.4}$$

Thus, we obtain the following MAP decision rule:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \tag{1.5}$$

$$= \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\}. \tag{1.6}$$

[5] The notational convention is as follows. We use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

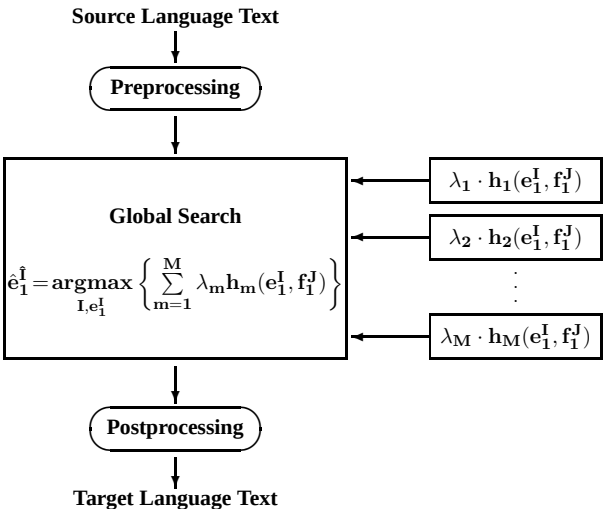

Figure 1.2. Architecture of the direct approach to SMT.

The direct approach has the advantage that additional models $h(\cdot, \cdot)$ can be easily integrated into the overall system. Furthermore, the classical approach given in Equation 1.3 can be interpreted as a special case of this approach. The model scaling factors λ_1^M are optimized according to the maximum class posterior criterion, originally, using the GIS algorithm [Och & Ney 02]. Today, most systems perform minimum error rate training (MERT) [Och 03], i.e. the scaling factors are directly optimized with respect to a certain MT evaluation criterion. Figure 1.2 illustrates the resulting architecture.

1.1.2 Phrase-Based Approach

To capture correspondences between the words in the source and the target language sentences, the IBM group introduced word alignments. An alignment can be viewed as mapping $a : j \rightarrow a_j \in \{0, \ldots, I\}$ assigning a target position a_j to each source position j [Brown & Della Pietra$^+$ 93]. The artificial target position zero is hereby defined to map source words which do not have any equivalence in the target sentence. Formally, the word alignment was incorporated into the model as a hidden variable. An example is shown on the left hand side of Figure 1.3. The German source language sentence "Wenn ich eine Uhrzeit vorschlagen darf?" is written along the x-axis and the English target language translation "If I may suggest a time of day?" along the y-axis. The word alignment is represented by the black squares.

Moreover, in state-of-the-art SMT systems, the word alignments are usually modeled implicitly through bilingual phrases. The basic idea of phrase-based translation (PBT) is to first segment the source language sentence into phrases, then translate each phrase, and finally compose the target sentence of these phrase translations. PBT is motivated by the fact that the context is important in translation. The corresponding phrase segmentation of the above mentioned example is depicted on the right hand side of Figure 1.3. Phrase pairs are represented as boxes.

Phrases are sequences of words in the two languages[6]. Then, a substring pair $(\tilde{f}, \tilde{e})$ of a

[6]Note that these are not necessarily phrases in the linguistic sense.

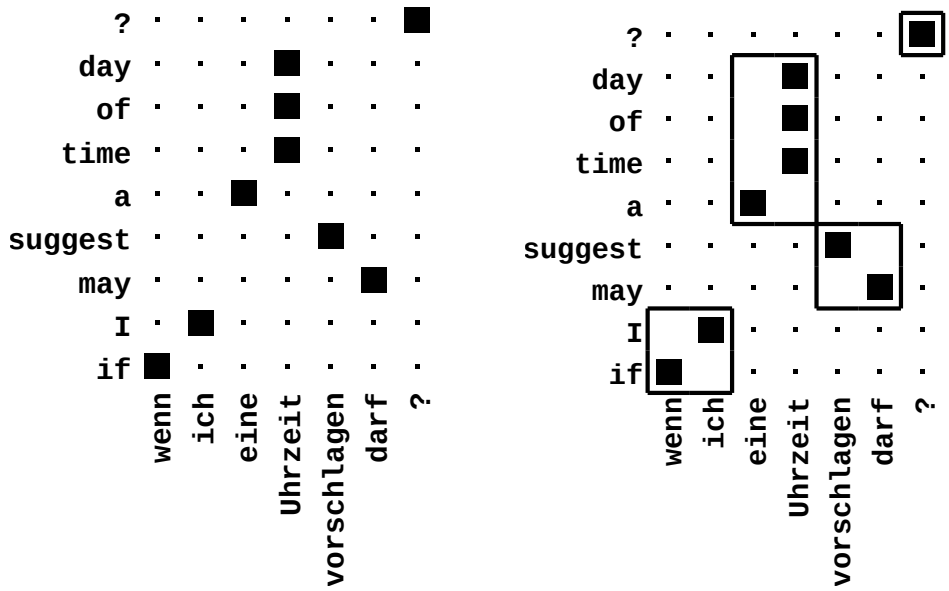

Figure 1.3. Example of a word alignment and the corresponding phrase segmentation.

sentence pair is a bilingual phrase, if:

- $\tilde{f}$ and $\tilde{e}$ are contiguous,
- all words in $\tilde{f}$ are aligned only to words in $\tilde{e}$, and
- all words in $\tilde{e}$ are aligned only to words in $\tilde{f}$.

Formally, we define a segmentation of a given sentence pair (f_1^J, e_1^I) into K phrase pairs:

$$k \quad \rightarrow \quad s_k := (i_k; b_k, j_k), \quad k = 1, \ldots, K \tag{1.7}$$

with i_k denotes the end position of the kth target phrase, and (b_k, j_k) denotes the start and end position of the source phrase which is aligned to the kth target phrase. The segmentations are constrained such that all words in the source and target sentence are covered by exactly one phrase. Hence, there are no gaps and no overlap.

Accordingly, for a given sentence pair (f_1^J, e_1^I) and a given segmentation s_1^K, we define the bilingual phrases $(\tilde{f}_k, \tilde{e}_k)$:

$$\tilde{e}_k \quad := \quad e_{i_{k-1}+1}, \ldots, e_{i_k} \tag{1.8}$$

$$\tilde{f}_k \quad := \quad f_{b_k}, \ldots, f_{j_k} \tag{1.9}$$

including a special symbol to ensure the proper handling of the sentence start and end positions. Figure 1.4 visualizes the phrase segmentation process. The segmentation contains the phrase-level reorderings.

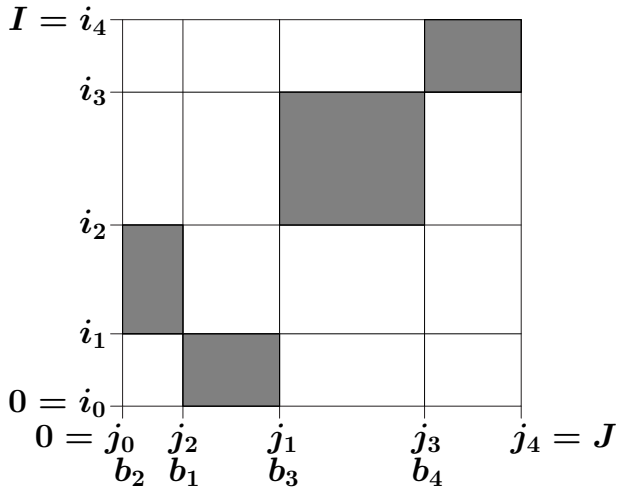

Figure 1.4. Illustration of the phrase segmentation process.

As for the word alignment, the segmentation s_1^K is introduced via a hidden alignment:

$$Pr(e_1^I|f_1^J) = \sum_{s_1^K} Pr(e_1^I, s_1^K|f_1^J) \tag{1.10}$$

$$= \sum_{s_1^K} \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, s_1^K; f_1^J)\right)}{\sum\limits_{e'^{I'}_1, s'^{K'}_1} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, s'^{K'}_1; f_1^J)\right)} \tag{1.11}$$

$$\approx \max_{s_1^K} \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, s_1^K; f_1^J)\right)}{\sum\limits_{e'^{I'}_1, s'^{K'}_1} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, s'^{K'}_1; f_1^J)\right)}. \tag{1.12}$$

In theory, we have to sum over all possible segmentations (Equation 1.11), but in practice, we perform the so-called maximum approximation (Equation 1.12). Thus, in the end we obtain the following MAP decision rule:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmax}_{I, e_1^I} \left\{ \max_{K, s_1^K} \sum_{m=1}^M \lambda_m h_m(e_1^I, s_1^K; f_1^J) \right\}. \tag{1.13}$$

We extend our models to include the additional hidden variable, i.e. we now have models $h(e_1^I, s_1^K; f_1^J)$ which also depend on the segmentation.

1.2 Robust Machine Translation for Multi-Domain Tasks

One of the reasons to follow the statistical approach to MT was that it has been proven competitive or superior to other "traditional" approaches in various comparative evaluations. The translation quality achieved for restricted domains is relatively high. Examples include the domains of news agencies as investigated by the DARPA TIDES and NIST Open MT evaluations,

appointment scheduling which was the scope of the Verbmobil project [Wahlster 00], or tourism which is used in the IWSLT evaluations.

Consequently, more challenging tasks have been tackled in MT research in recent years. The TC-STAR (*Technology and Corpora for Speech to Speech Translation*)[7] project, for example, has dealt with speech translation of the plenary sessions of the European Parliament. The domain and the vocabulary of these speeches are open. Thus, the translation engine has to cope with the effects of spontaneous speech, misrecognitions from the automatic speech recognizer, and unknown words or constructs which are due to the variety of domains. Targeting for a similar direction, the GALE (*Global Autonomous Language Exploitation*)[8] program aims at developing and applying computer software technologies to absorb, translate, analyze, and interpret huge volumes of speech and text in multiple languages. This also involves the processing of noisy and unstructured input data, e.g. data collected from newsgroups or weblogs, or automatic transcriptions of arbitrary broadcast conversations. Within GALE, three research teams participate and evaluate their engines against each other. The SMT system presented in this work is the primary MT engine of the SRI Nightingale team[9] for the Arabic-English tasks.

In comparison to conventional MT tasks, two problems are striking: *robustness* with regard to the translation of automatically recognized speech and unstructured input, and the ability to produce good quality translations for *multi-domain* tasks.

The aim of this work is to build an SMT system which is robust and suited for multi-domain input at the same time. Thereby, we focus on Arabic-English translation tasks. We cover all important steps to set up an Arabic-English SMT system that achieves state-of-the-art performance and present experimental results on different translation tasks. The organization of this thesis is as follows: in the next section, we state important publications and overview the state-of-the-art in SMT. Chapter 2 then briefly summarizes the scientific goals of this work. Processing morphologically rich languages such as Arabic generally leads to high demands on preprocessing. In Chapter 3, we compare different approaches to Arabic preprocessing and investigate methods for word normalization and word segmentation. Chapter 4 deals with the core component of any SMT system, i.e. the search process. The goal of the search is to find the maximizing argument in the MAP decision rule (cf. Equation 1.1). We give a detailed description of the search algorithm, make an explicit distinction between reordering and lexical hypotheses in the pruning process, and analyze different reordering constraints. A straightforward way to incorporate structural knowledge into the translation framework is to generate a repository of possible translation candidates and to rerank these hypotheses using models that reflect the structural properties. The main purpose is to identify the ungrammatical hypotheses from the system's output and, thus, improve its "fluency". This is the focus of Chapter 5. Our extensions to qualify the system for robustness and multi-domains are presented in Chapter 6.

Subsequently, Chapter 7 studies the transliteration of proper names. Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Our work is motivated by the problem of out-of-vocabulary

[7]The TC-STAR project, financed by European Commission within the Sixth Program, is envisaged as a long-term effort to advance research in all core technologies for Speech-to-Speech Translation (SST), `http://www.tc-star.org/`.

[8]GALE is an Information Processing Techniques Office (IPTO) program funded by the Defense Advanced Research Projects Agency (DARPA), `http://www.darpa.mil/ipto/programs/gale/gale.asp`.

[9]For details about the SRI Nightingale team participating in GALE, please see `http://www.speech.sri.com/projects/GALE/`.

(OOV) terms which are a persistent problem in any SMT system. Often, such terms are the names of entities and the intention is to preserve the names by transliteration. Chapter 8 again deals with the search process but concentrates on the application of an SMT system within an computer-assisted translation (CAT) environment. We compare different search strategies for interactive (statistical) MT and analyze how they perform when strict time constraints have to be met. The experimental results for all methods described in this thesis are presented in Chapter 9. Finally, Chapter 10 summarizes the scientific contributions of this work followed by a short conclusion and outlook.

1.3 Related Work

As mentioned before, the beginnings of SMT can be traced back to the early 1950's but the fundamentals of nowadays SMT systems were published by the IBM Yorktown Heights group in the late 1980's and beginning 1990's [Brown & Cocke$^+$ 88, Brown & Cocke$^+$ 90, Brown & Della Pietra$^+$ 93]. They introduced the concept of word alignments to describe the correspondences between source and target words. As only single-word based lexicon models were applied, the translation probabilities depended just on single words. Furthermore, they developed a search algorithm for these models based on stack decoding [Berger & Brown$^+$ 96]. Following the work of the IBM group, the goal of the Johns Hopkins University summer research workshop 1999 was to construct a basic SMT toolkit [Al-Onaizan & Curin$^+$ 99]. The main component was the GIZA tool for the EM training of the word alignment models, which was later extended to GIZA++ [Och & Ney 03]. Even for simple translation models, the search problem in SMT is NP complete [Knight 99]. Consequently, various research groups tried to extend the IBM work to develop more efficient search algorithms, e.g. applying multi-stack decoding [Wang & Waibel 97], greedy techniques [Germann & Jahr$^+$ 01, Germann & Jahr$^+$ 04] or dynamic programming [Tillmann & Vogel$^+$ 97, Tillmann & Ney 00, Tillmann & Ney 03].

A major disadvantage of the baseline IBM models is that they do not take the word context into account. Nowadays, the phrase-based approach to SMT is predominant, i.e. the translation models do not depend on single words but on word groups. Most phrase-based systems, including the one used for this work, are derived from the alignment templates approach [Och & Tillmann$^+$ 99, Och 02, Och & Ney 04]. An alignment template is defined as a triple describing the alignment between a source phrase and a target phrase. They are defined at the level of word classes and are extracted from word-aligned bilingual corpora. In [Och & Tillmann$^+$ 99, Och 02], the alignment templates system was shown to outperform single-word based approaches.

For this thesis, we employ our phrase-based decoder [Zens & Och$^+$ 02, Zens & Ney 04] which is a further development of the alignment templates approach. The phrase extraction uses the same algorithm as for the alignment templates but the phrases are directly defined at the word level, i.e. no word classes are used.

There exist a variety of similar approaches to phrase-based MT: [Tomás & Casacuberta 01], for example, constrained the phrase segmentation to be monotonic and applied the EM algorithm to estimate the phrase translation probabilities. Experimental results were given for a Spanish-to-Catalan translation task. Nevertheless, the monotonicity constraint seems inappropriate for more distinct language pairs. Therefore, [Tomás & Casacuberta 04] presented an extension which, at least, allows for non-monotonic decoding. In [Marcu & Wong 02], a joint probability model is assumed. The phrase extraction does not rely on the word

alignment but directly generates the phrase alignment again using the EM algorithm. Although [Birch & Callison-Burch$^+$ 06] worked on the scalability, this approach remains applicable only for small tasks. Similarly, [Tillmann & Xia 03] described a phrase-based unigram model estimating the joint probability of source and target phrases using relative frequencies. As for the alignment templates, the phrase pairs are extracted from word-aligned bilingual corpora, cf. [Tillmann 03]. Instead of estimating by relative frequencies, [Vogel 03] computed the phrase translation probabilities using the IBM model 1 lexicon scores. Furthermore, they allowed word reorderings within a window of up to three positions and reported an improvement for a Chinese-to-English task in comparison to monotone search. [Koehn & Och$^+$ 03] created a framework to evaluate and compare various aspects of phrase translation methods, such as the phrase extraction algorithm, the word alignment model, and so on. In addition, the publicly available Pharaoh decoder [Koehn 04] advanced research in PBT as it allowed for experiments to be carried out at state-of.the-art level. The decoder was re-written and published as open-source [Koehn & Hoang$^+$ 07] during the Johns Hopkins University summer research workshop 2006.

Another approach which has become popular in recent years is to incorporate syntactical knowledge into the search process. These approaches parse the sentences in one or both of the involved languages and the translations are then performed on tree structures. The tree structures can be monolingual on the target language side as in, for example, [Yamada & Knight 01, Yamada & Knight 02] and more recently [Galley & Hopkins$^+$ 04, Galley & Graehl$^+$ 06], or bilingual, i.e. synchronous tree structures as e.g. in [Wu 96, Wu 97, Lin 04, Melamed 04, Chiang 05, Ding & Palmer 05, Zollmann & Venugopal 06, Chiang 07]. Some of the approaches are based on the phrase-based approach, e.g. [Chiang 05, Chiang 07] constructed hierarchical transducers for translation. The model is a syntax-free grammar which is learnt from bilingual corpora without any syntactic information. It consists of phrases which may contain sub-phrases, such that a hierarchical structure is induced. Also, [DeNeefe & Knight$^+$ 07] utilized methods from the phrase-based approach. Even though all of these approaches are formally syntax-based, not all of them deploy linguistic constituents.

It was also investigated to model the translation process as a finite state transducer. The use of finite state techniques for MT was already proposed in [Alshawi 96, Vidal 97, Knight & Al-Onaizan 98, Bangalore & Riccardi 00, Casacuberta & Llorens$^+$ 01]. This approach solves the translation problem by estimating a language model on a "bilanguage" defined over the source and target language. Then, the translation transducer is basically an acceptor for the bilanguage. [Kumar & Byrne 03, Kumar & Byrne 05] presented a weighted finite state transducer implementation for the alignment templates and for the phrase-based approach. The approach is appealing as there exists publicly available tools for manipulating finite state automata, for instance [Mohri & Pereira$^+$ 02, Kanthak & Ney 04].

Often, these transducer approaches were applied to spoken language translation (SLT) tasks [Vidal 97, Bangalore & Riccardi 00, Casacuberta & Llorens$^+$ 01] as the finite state techniques allow for a straightforward coupling of automatic speech recognition (ASR) and machine translation (MT). A theoretical basis for combining the recognition model scores and the translation model scores was given by [Ney 99]. Whereas simple speech translation systems translate single-best recognizer output, recent improvements have been reported by using multiple recognition hypotheses. [Bozarov & Sagisaka$^+$ 05, Bertoldi 05], for instance, reported moderate improvements by translating n-best ASR hypotheses. [Schultz & Jou$^+$ 04, Matusov & Kanthak$^+$ 05, Matusov & Kanthak$^+$ 06] processed speech recognition lattices. The interface between ASR and MT poses a field of research on its own and is clearly out of

the scope of this thesis. However, most state-of-the-art ASR systems were developed without considering the recognized word sequence as input to MT. The recognized word sequences are divided into utterances based on speech/non-speech detection such that the MT system has to cope with possibly very long (containing several sentences) or very short (just one to two word fragments) utterances. Here, we focus on our work on adjustment of ASR and MT vocabularies, automatic sentence segmentation and punctuation prediction [Bender & Matusov$^+$ 07, Matusov & Hillard$^+$ 07].

SMT systems have not only been used as pure translation engines but also as companion tools to assist skilled human translators. The concept of machines assisting human translators was first proposed by [Kay 80] and later refined to interactive machine translation by [Foster & Isabelle$^+$ 96]. In such an environment, human translators interact with a translation system that acts as an assistance tool and dynamically provides a list of translations that best complete the part of the source sentence already translated. Refinements were presented in [Foster & Isabelle$^+$ 97, Langlais & Foster$^+$ 00a, Langlais & Foster$^+$ 00b, Foster 02, Foster & Langlais$^+$ 02]. A first implementation was carried out in the TransType project [Langlais & Foster$^+$ 00b] and further enhanced in the course of the TransType2 project [SchlumbergerSema S.A. & Intituto Tecnológico de Informática$^+$ 01]. As part of TransType2, the systems were proofed useful in several field trials with professional translators [Macklovitch & Nguyen$^+$ 05, Macklovitch 06]. Here, we deal with search strategies for interactive (statistical) machine translation systems.

Also, the task of transliterating names has received a significant amount of research in the last decade. There exist two different approaches to this task: first, the actual transliteration of proper names, i.e. the mapping of the source language names onto new, plausible target language spellings, and second, the (phonetic) matching of the source language names against a large list of candidate transliterations. The state-of-the-art in name transliteration generally involves some form of generative component, e.g. [Knight & Graehl 97] proposed a generative model based on finite state techniques for Japanese-English back transliteration which was extended to Arabic-English transliteration [Stalls & Knight 98] and eventually incorporated web-counts to rescore the transliteration candidates [Al-Onaizan & Knight 02b]. Further details about their transliteration algorithm can be found in [Knight & Graehl 98, Al-Onaizan & Knight 02a]. Other generative approaches were proposed in [Meng & Lo$^+$ 01, Moore 03, Huang & Vogel$^+$ 03, Huang & Vogel$^+$ 04, Haizhou & Min$^+$ 04, Sherif & Kondrak 07, Kashani & Popowich$^+$ 07]. Such a generative model is also a core component of any system that follows the second approach. The concept of matching names against a list of transliteration candidates emerged from the attempts to better handle named entities in MT, e.g. [Lee & Chang 03]. In addition to the generative model, a large list of names in the target language and a similarity metric for the words in the two languages are crucial. Obviously, research work and experimental findings for the second approach, e.g. [Huang & Vogel$^+$ 04, Freitag & Khadivi 07, Hermjakob & Knight$^+$ 08], are always closely related to the first approach. Moreover, [Hermjakob & Knight$^+$ 08] focused on the question when to transliterate names and when to let the MT system generate the translations of proper names. To promote the work on name handling, an ACE entity translation pilot evaluation (ET07) was recently set up [Day 07]. Here, we focus on the first approach. We compare different statistical methods for name transliteration and test how they perform when combined within a system combination framework.

Chapter 2

Scientific Goals

The aim of this work is to build a robust SMT system for multi-domain translation tasks. As discussed in the introductory chapter, the successful application of SMT systems for conventional MT tasks such as the translation of newswire articles has set up more challenging translation tasks. We study robustness regarding unstructured and automatically recognized input and the special requirements of multi-domain tasks. In particular, the following scientific goals are pursued:

- **Detailed analysis of the search problem:**
  Search is the task of finding the target language sentence that is the most probable translation of a given source language sentence. [Zens 08] formulated the search problem for phrase-based SMT, analyzed the search space, and presented implementations of different search algorithms in detail. He resorted to a beam search strategy to address the search problem and arrived at the conclusion that it is important to make an explicit distinction between reordering and lexical hypotheses in the pruning process. Thereby, he concentrated mainly on the Chinese-English language pair and on traditional NIST data, i.e. input out of the newswire domain. In this thesis, we take up the work of [Zens 08] and analyze the search problem in phrase-based MT for the Arabic-English language pair. We investigate the contribution of the different models applied during search and study the reordering problem linked to Arabic-English translation tasks. Furthermore, we pay attention to the targeted goals, i.e. robustness and multi-domain input.

- **Robustness and multi-domains:**
  Open domains and noisy input, e.g. weblog documents or automatic transcriptions of arbitrary broadcast conversations, pose a special challenge for any SMT system. Parts of the unstructured and ambiguous text components are normalized within the preprocessing step. Any further approach aims at adapting the MT system to the specific input domain. The most straightforward way to perform domain adaptation for the overall translation system is by use of genre-specific language models (LMs). In the literature, one distinguishes approaches performing a combination of several sub-LMs, most often an interpolation of a general and a domain-specific model as proposed by [Seymore & Rosenfeld 97], and approaches additionally aiming at retrieval of documents from large monolingual corpora, e.g. [Iyer & Ostendorf 99]. More recently, [Zhao & Eck$^+$ 04] reported improvements in translation quality by building specific language models for each sentence to be translated. Apparently, these improvements are connected to very high computational costs. [Bulyko & Matsoukas$^+$ 07] performed discriminative adaptation and optimized the LM interpolation weights w.r.t. an MT evaluation criterion. Then, they used the adapted LMs for n-best rescoring of speech translation hypotheses and reported slight improvements of 0.3-0.4% Bleu compared to an unadapted LM. Here, we combine different corpora

representing various genres and tune the interpolation weights such that the LM perplexity is minimized per genre. Thus, we arrive at domain-specific LMs for each genre which are applied already during search.

For SLT, i.e. the translation of automatically recognized speech, most publications aim for integrated approaches by coupling of automatic speech recognition (ASR) and machine translation (MT). However, the integrated approach was not implementable for the presented system as our system was the primary MT engine of the SRI GALE team and thus had to translate transcriptions of different speech recognizers. Moreover, these ASR systems were developed without considering the recognized word sequence as input to MT. Hence, we adapt the ASR output to a more conventional SMT task. We use the algorithm of ICSI/UW [Matusov & Hillard$^+$ 07] for sentence unit segmentation and focus on our work on adjustment of ASR and MT vocabularies and punctuation prediction [Bender & Matusov$^+$ 07]. Predicting punctuation marks in the ASR output worked well for Spanish as the source language [Matusov & Mauser$^+$ 06]. However, the Arabic punctuation rules are quite different from the English ones. Therefore, the prediction of punctuation marks in the target language with the joint power of the phrase-based translation models and the language model is more reliable.

- **Morpho-syntactic Arabic preprocessing:**
  Arabic is a highly inflected language compared to languages like English which have very little morphology. A usual phenomenon in Arabic is the attachment of a group of words which are semantically dependent on each other. Hence, an Arabic word can be decomposed into "prefixes, stem, and suffixes". Furthermore, diacritics are generally omitted in written Arabic which leads to high ambiguity of words. [Larkey & Ballesteros$^+$ 02] already showed that Arabic word segmentation improves the accuracy of information retrieval systems, and a comparable work was done by [Diab & Hacioglu$^+$ 04] for part of speech (POS) tagging. In this work, we present our finite state automaton (FSA) based approach to Arabic word segmentation and compare it with two different approaches [Habash & Rambow 05, Mansour & Sima'an$^+$ 07] which involve some form of morphological analysis. In [Habash & Rambow 05], a morphological analyzer was used for the word segmentation and POS tagging. [Mansour & Sima'an$^+$ 07] presented a (morphological) POS tagger. For both tools, the text input to the SMT system is obtained by further preprocessing of the analyzer output. Unlike these approaches, our FSA method is unsupervised, i.e. we do not need any manually segmented training data, is easily adaptable to different target languages, and is computationally very efficient. We analyze varying segmentation schemes and normalization strategies, and examine the impact of the applied preprocessing on the translation quality.

- **Incorporation of structural properties:**
  Statistically driven MT systems make a lot of errors that seem, at least from a human point of view, illogical. We investigate a means of identifying ungrammatical hypotheses from the output of an SMT system by using grammatical knowledge that expresses syntactic dependencies of words or word groups. We introduce several methods that try to establish this kind of linkage between the words of a translation hypothesis and, thus, determine its well-formedness, or "fluency". Thereto, we rerank the n-best translation candidates using appropriate rescoring models which capture these dependencies. In a similar approach, [Och & Gildea$^+$ 04] investigated a large amount of different feature

functions. The field of application varied from simple syntactic features, such as IBM model 1 score, over shallow parsing techniques to more complex methods using grammars and intricate parsing procedures. The results were rather disappointing. Only one of the simplest models, i.e. the implicit syntactic feature derived from IBM model 1 score, yielded consistent and significant improvements. All other methods had only a very small effect on the overall performance.

- **Transliteration of proper names:**
  Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Our work is motivated by the problem of out-of-vocabulary (OOV) terms which are a persistent problem in any SMT system. Often, such terms are the names of entities and the intention is to preserve the names by transliteration. As mentioned in the previous section, most nowadays name transliteration systems combine a generative component with additional knowledge sources, e.g. by matching names against a list of transliteration candidates or by using (cross-document) coreference for the two languages. In this thesis, we focus on the generative component and analyze different statistical methods which have been successfully applied to other NLP tasks in the context of name transliteration. These methods are purely data-driven and do not require any additional knowledge but a set of training name pairs. The system of [Freitag & Khadivi 07] was one of the transliteration engines of the NYU-Fair Isaac-RWTH entity translation system [Ji & Blume$^+$ 07] used to participate in the ACE entity translation pilot evaluation (ET07). [Freitag & Khadivi 07] thereby proved to achieve state-of-the-art transliteration accuracy but also carried out experiments on the same Arabic-English transliteration task investigated in this work. Thus, we use their results as reference values. Finally, we test how the individual approaches perform when combined within a ROVER-based system combination framework.

- **Search strategies for interactive machine translation:**
  Here, we deal with search strategies for interactive (statistical) MT systems. Clearly, the best approach would be to start a new search for every given prefix. However, in these kind of systems, response time is a crucial factor for a human translator as delays higher than a fraction of a second are not acceptable. With today's algorithms and available computing power these time restrictions can not be met when doing a full search for each prefix, so the performance achieved with this strategy will be an upper bound of the performance we get in the real system. Consequently, [Och & Zens$^+$ 03] developed an efficient algorithm for interactive translation using word graphs as representation of the search space. We extended the system of [Och & Zens$^+$ 03] and carried out further experiments [Barrachina & Bender$^+$ 09] in the course of the TransType2 project. In this work, we study efficient search strategies and compare their capabilities with the system using the full search strategy.

Chapter 3

Morpho-Syntactic Arabic Preprocessing

In this chapter, we focus on preprocessing Arabic text input to SMT systems. Arabic is a highly inflected language in contrast to English which has very little morphology. This morphological richness makes SMT from Arabic to English a challenging task. As for other languages, the text input must be first tokenized. Furthermore, we apply normalization rules and remove all diacritics. We then address the morphological richness of the Arabic language by splitting the words in prefixes, stem and suffixes, thus simplifying the inflected Arabic text input. We compare our finite state automaton (FSA) based approach [Isbihani & Khadivi$^+$ 06] to two different approaches [Habash & Rambow 05, Mansour & Sima'an$^+$ 07] which have been investigated in the course of this work. We analyze varying segmentation schemes and normalization strategies, and examine the impact of the applied preprocessing on the translation quality.

3.1 Motivation

A usual phenomenon in Arabic is the attachment of a group of words which are semantically dependent on each other. For instance, prepositions like "and" and "then" are usually attached to the next word. This applies to the definite article "the" as well. In addition, personal pronouns are attached to the end of verbs, whereas possessive pronouns are attached to the end of the previous word which constitutes the possessed object. According to this, an Arabic word can be decomposed into *prefixes*, *stem* and *suffixes*.

With respect to SMT, the Arabic morphology directly impacts the vocabulary size and the number of singletons the SMT systems have to deal with. The inflected Arabic words do not occur often enough in the training data to be captured by machine learning algorithms. This can lead to an inefficient word/phrase alignment and to a huge number of unknown words when translating new text.

In order to tackle this problem and to increase the translation quality of SMT systems, each Arabic word is decomposed into its parts, i.e. prefixes, stem and suffixes. [Larkey & Ballesteros$^+$ 02] already showed that Arabic word segmentation improves the accuracy of information retrieval systems. In [Lee & Papineni$^+$ 03], a statistical approach for Arabic word segmentation was presented. It decomposes each word into a sequence of morphemes (prefixes - stem - suffixes) where all possible prefixes and suffixes are split from the original word. A comparable work was done by [Diab & Hacioglu$^+$ 04] including the discussion of a part of speech (POS) tagging method for Arabic. In [Habash & Rambow 05], a morphological analyzer was used for the word segmentation and the POS tagging.

3.2 Tokenization and Normalization

As for any language which is to be translated via SMT, the Arabic input must be first tokenized. Here, words and punctuation marks (except for abbreviations) are separated. Although abbreviations are not widely spread in Arabic, we encounter more and more abbreviations, especially for person names and some titles like doctor or professor. Since most abbreviations in Arabic consist of one single character, this problem can be easily solved. Another criterion is that Arabic has some characters which appear only at the end of a word. We use this criterion to split words that are wrongly attached to each other.

Moreover, the Arabic written language does not contain vowels. Instead, diacritics are used to define the pronunciation of a word (diacritics are written under or above the pronounced character(s) in the word). Usually, these diacritics are omitted in written text, thus increasing the ambiguity of a word. Resolving the ambiguity of a word is then dependent on the context. However, the authors sometimes write a diacritic on a word to help the reader and give him a hint which word is really meant. As a result, a single word with the same meaning can be written in different ways. For example, شعب (*$Eb*) can be read as *sha'ab* (Eng. "nation") or *sho'ab* (Eng. "options")[1]. If the author wants to give the reader a hint that the second word meaning is intended, he can write شُعب (*$uEb*) or شُعَب (*$uEab*). To avoid this problem, we normalize the text by removing all diacritics.

After tokenizing the text, the length of the sentences increases considerably, especially the number of occurrences of the stripped article ال (*Al*, Eng. "the") is very high. Not every article in an Arabic sentence matches to an article in the target language. One of the reasons is that the adjective in Arabic is annotated with an article if the word it describes is definite. So, if a word has the prefix *Al*, then its adjective will also have *Al* as a prefix. In order to reduce the sentence length, we remove all these articles which are supposed to be attached to an adjective. A different method for determiner deletion was described by [Lee 04].

3.3 Word Segmentation

As mentioned in the beginning of this chapter, one way to simplify Arabic text input for SMT systems is to split the words in prefixes, stem and suffixes. The most straightforward method for Arabic word segmentation is to simply decide where to split the inflected words based on the frequency of the resulting stems in comparison to the frequency of the inflected words. If the inflected word has a higher frequency than all possible stems, it will not be split. This method is very similar to the method used for splitting German compound words [Koehn & Knight 03] and may advance the SMT models by harmonizing the Arabic data sets (training data as well as test sets). We would just need a set of all prefixes and suffixes and their possible combinations. However, a simple frequency-based approach does not account for any type of linguistic knowledge. It is possible to split a word to parts without linguistic meaning. Another disadvantage is its unawareness of the relationship between the split parts and the translation of the inflected word. This method may split an inflected word even if it has a single word translation in English.

In the next subsections, we describe the three approaches to Arabic word segmentation investigated in this work.

[1] There are other possible pronunciations for the word شعب (*$Eb*) than the two mentioned.

Table 3.1. Arabic prefixes handled in this work and their English meanings.

Prefix	و	ف	ك	ل	ب	أَل
Transliteration	*w*	*f*	*k*	*l*	*b*	*Al*
Meaning	and	and then	as, like	in order to	with, in	the

Table 3.2. Arabic suffixes handled in this work and their English meanings.

Suffix	ي	ني	ك	كمَا، كم ، كن
Transliteration	*y*	*ny*	*k*	*kmA, km, kn*
Meaning	my	me	you, your (sing.)	you, your (pl.)

Suffix	نَا	ه	هَا	همَا، هم، هن
Transliteration	*nA*	*h*	*hA*	*hmA, hm, hn*
Meaning	us, our	his, him	her	them, their

3.3.1 Finite State Automaton-Based Approach

For this approach, we restrict the set of prefixes and suffixes to those shown in Table 3.1 and 3.2. Each of the prefixes and suffixes has at least one meaning which can be represented by a single word in the target language. The tables show the corresponding transliterations and English meanings as well.

Some of the prefixes can be combined. For example, the word وبَالقلم (*wbAlqlm*) which means "and with the pen" has a prefix which is a combination of three prefixes, namely و (*w*), ب (*b*) and ال (*Al*). In contrast, the suffixes we handle in this work can not be combined with each other. Thus, the compound word pattern handled here is: *prefixes - stem - suffix*. All possible prefix combinations that do not contain ال (*Al*) allow the stem to have a suffix. Note that there are other suffixes not handled here, such as ات (*At*), ان (*An*) and ون (*wn*) which form the plural of a word. We omit them because they do not have their own meaning.

To segment the Arabic words into prefixes, stem and one suffix, we implemented two finite state automata. The first finite state automaton (FSA) strips off the prefixes. To this prefix FSA, we then append the second FSA which takes care of the suffixes.

Figure 3.1 shows the FSA for stripping all possible prefix combinations. The prefix س (*s*) changes the verb tense to the future and should thus be added to the set of prefixes which must be stripped (cf. Table 3.1). The *s* prefix can only be combined with *w* and *f*. Our motivation is that the future tense in English is built by adding the separate word "will".

In detail, the automaton shown in Figure 3.1 consists of the following states:

`S`:	the starting point of the automaton,
`E`:	the end state which can only be reached if the resulting stem already exists in the text,
`WF`:	the state is passed through if the word begins with *w* or *f*, and
`C`, `K`, `L`, `B` and `AL`:	the states are passed through if the word begins with *s, k, l, b* or *Al*, respectively.

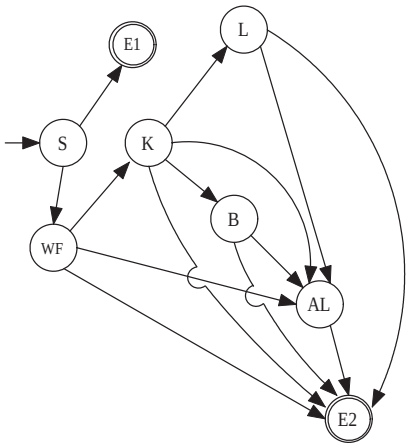

Figure 3.1. Finite state automaton (FSA) for stripping prefixes off Arabic words.

To reduce the number of erroneous segmentations, we allow only transitions such that the resulting stem occurs at least one time in the training corpus. We run the segmenter iteratively and add the produced words to the vocabulary after each iteration. Doing so, we ensure that most of the inflected Arabic words are recognized and segmented. Also, we will be able to recognize unseen inflected words in the next iteration(s). Experiments showed that running the segmenter twice is sufficient and that in higher iterations most of the added segmentations are wrong.

However, we obtained additional stems in the segmented text which do not make sense in Arabic. Although restricting the finite state segmenter in such a way that words are segmented only if the yielded stem already exists in the corpus, we still produce falsely segmented words. Moreover, because the stripping of prefixes and suffixes is done consecutively, not all possible word segmentations are considered. Another problem is that the finite state segmenter does not care about ambiguities but splits everything it recognizes. For example the word فرد (*frd*): in one case, the word means "person" and the character *f* is separable from the word. Therefore *f* can not be segmented. In the other case, the word can be segmented to *f rd* which means "and then he answers" or "and then an answer". If the words الفرد, فرد and رد (*Alfrd*, *frd* and *rd*) occur in the corpus, then the finite state segmenter will transform *Alfrd* which means "the person" to *Al f rd* which can be translated as "the and then he answers". Hence, the meaning of the original word is distorted.

To solve these problems, we improve our approach in a way that prefixes and suffixes are recognized simultaneously. The segmentation of ambiguous words should be avoided. In this manner, we intend to postpone resolving such ambiguities to the SMT system. The question now is how the segmentation of ambiguous words can be avoided. For this, it is sufficient to find a word that contains the prefix as a non-separable character. E.g. in the last example, the word *Alfrd* contains the prefix *f* as a non-separable character and therefore only *Al* can be stripped off the word. The next question is whether a character belongs to the word or is a prefix. We can extract this information using the invalid prefix combinations. For example, *Al* is always the last prefix which can occur. Therefore, all characters that occur in a word after *Al* are non-separable characters. This method can be applied for all invalid combinations to extract new rules deciding whether a character in a word is a non-separable one or not.

On the other side, all suffixes we handle in this approach (see Table 3.2) are pronouns. Therefore, it is not possible to combine them as a suffix. We use this fact to make a decision whether the last characters in a word are non-separable or can be stripped. For example, the word تركهم (*trkhm*) means "he lets them". If we suppose that *hm* is a suffix that must be stripped off, then we can conclude that *k* is a non-separable character and not a suffix. In this way, we are able to extract the decisions whether and how to segment a word from the corpus itself.

In order to implement these changes, the original automaton is modified. Instead of splitting a word, we keep track of the states traversed until the end state is reached and label the word accordingly. We use the techniques described above to generate "negative" labels which avoid the corresponding splitting. If a word is labeled to be split and "negative" at the same time, only the negation is considered. At the end, each word is split according to its label.

Unlike the two following approaches, this method is unsupervised, i.e. we do not need any manually segmented training data, and is therefore easily adjustable in the sense that the segmented Arabic text is best adapted to different target languages.

3.3.2 Word Segmentation Derived From Full Morphological Disambiguation

[Habash & Rambow 05] presented an approach to using a morphological analyzer for tokenizing and morphologically tagging (including part-of-speech tagging) Arabic words in one process. The analyzed output can be used as preprocessed text input for MT, either just the segmented words or including the full morphological information. Following their approach, [Habash & Sadat 06] then studied the effect of different strategies for word segmentation on the translation quality. The tool is called *MADA* (Morphological Analysis and Disambiguation for Arabic) and is publicly available from the Columbia University group[2].

MADA is motivated by the fact that Arabic has a large set of morphological features such as gender, number, person or voice, which have to be distinguished from a set of attachable clitics (e.g. the definite article "the" ال (*Al+*) , or the class of particle proclitics "to/for" لِ (*l+*), "by/with" بِ (*b+*), "as/such" كَ (*k+*), and so on). The question whether to split off a clitic or feature or whether to simply abstract it is dependent on the context. MADA was designed to resolve this ambiguity in Arabic words [Habash & Rambow 05].

The toolkit is trained on the Penn Arabic Treebank [Maamouri & Bies$^+$ 04] and uses the Buckwalter Arabic Morphological Analyzer (BAMA) [Buckwalter 02] to limit the set of possible word analyses. MADA then uses the following morphological features to select among the BAMA analyses:

- POS:
  Basic part-of-speech tag. The POS tag set is a subset of the tag set that was introduced for the English Penn Treebank: V (Verb), N (Noun), PN (Proper Noun), AJ (Adjective); AV (Adverb), PRO (Nominal Pronoun), P (Preposition/Particle); D (Determiner); C (Conjunction), NEG (Negative Particle), NUM (Number), AB (Abbreviation), IJ (Interjection), PX (Punctuation) and X (Unknown).

[2]MADA is a full morphological tagger for Modern Standard Arabic developed by Nizar Habash and Owen Rambow, Columbia University, and is freely available at `http://www1.cs.columbia.edu/~rambow/software-downloads/MADA_Distribution.html`.

Table 3.3. Possible MADA analyses for the word وَالي (*wAly*), (from [Habash & Rambow 05]).

lexeme	gloss	POS	Conj	Part	Pron	Det	Gen	Num	Per	Voice	Asp
wAliy	ruler	N	NO	NO	NO	No	masc	sg	3	NA	NA
<ilaY	and to me	P	YES	NO	YES	NA	NA	NA	NA	NA	NA
waliy	and I follow	V	YES	NO	NO	NA	neut	sg	1	act	imp
|l	and my clan	N	YES	NO	YES	NO	masc	sg	3	NA	NA
|liy~	and automatic	AJ	YES	NO	NO	NO	masc	sg	3	NA	NA

- Conj:
  Binary feature checking if there is a cliticized conjunction in the word.
- Part:
  Binary feature checking if there is a cliticized particle in the word.
- Pron:
  Binary feature checking if there is a pronominal clitic in the word.
- Det:
  Binary feature checking if there is a cliticized definite determiner آل (*Al+*).
- Gen:
  Gender: masculine, feminine, neuter.
- Num:
  Number: singular, dual, plural.
- Per:
  First, second or third person.
- Voice:
  Active or passive voice.
- Asp:
  Aspect: imperfective, perfective, imperative.

Figure 3.3 shows the MADA feature values for different BAMA analyses of the word وَالي (*wAly*). For each of these features, a classifier is trained using the *YamCha* toolkit (Yet Another Multipurpose CHunk Annotator)[3] from [Kudo & Matsumoto 03], and the final word analysis is chosen by majority agreement, i.e. the number of classifiers agreeing with the analysis.

In order to obtain suitable text input for SMT systems, the MADA output needs to be further preprocessed as SMT systems such as the one presented in this thesis are trained and run on plain text only. [Habash & Sadat 06] presented different preprocessing schemes for this task, i.e. strategies for which features resp. clitics to split off and which to just abstract off. Out of the presented schemes, we examine the *D2* and *ATB* scheme in this work:

- D2 splits off the class of conjunction clitics (*w+* and *f+*) and the class of particles (*l+*, *k+*, *b+* and *s+*).

[3] YamCha is a generic, customizable, and open source text chunker oriented toward a lot of NLP tasks, and is freely available at `http://chasen.org/~taku/software/yamcha/`.

- ATB performs a tokenization that is fully compatible with the guidelines of the Penn Arabic Treebank. It decliticizes similar to D2 but splits off all pronominal clitics as well.

Additionally, we test normalizing Yaa ({ى, ي} → ى) and Alef ({ ا, أً, إٍ, أٌ, آ} → ا), and investigate the effect of the orthographical normalization module for word stems which can be either enabled or disabled via MADA's config.

3.3.3 Using POS Tags to Infer Segmentation

[Diab & Hacioglu$^+$ 04] proposed solutions to word segmentation and POS tagging of Arabic text. In the first step, the text must be converted to the Buckwalter encoding which is a one-to-one mapping of the Arabic UTF-8 characters to ASCII correspondents. In the second step, the text is being segmented and tokenized. Next, a partial lemmatization is done, and finally the POS tagging is performed. The approach of [Diab & Hacioglu$^+$ 04] is *data-driven*, i.e. except for annotated training data it does not require any additional lexical information. For the purpose of training, the Penn Arabic Treebank was used.

Similar to [Habash & Rambow 05], [Mansour & Sima'an$^+$ 07] reported improvements in POS tagging accuracy by incorporating a lexical analyzer. However, they started from an existing (morphological) POS tagger for Modern Hebrew (*MorphTagger*) [Bar-Haim & Winter 05] and showed that it is possible to port it to Arabic using a morphological analyzer (Buckwalter's analyzer) and a tagged corpus (the Arabic Tree Bank). They reported state-of-the-art tagging accuracy but found at the same time that further improvements are hindered by the limited coverage of the morphological analyzer. This can be due to "unknown" words (OOVs) or "known" words for which a morphological analysis is missing. [Mansour & Sima'an$^+$ 07] reported that missing analyses are more severe than OOVs and therefore proposed a method to smooth their HMM-based MorphTagger using [Diab & Hacioglu$^+$ 04]'s data-driven SVM model. If the tagging probability of the MorphTagger is below an empirically calculated threshold, the model adds the analyses of the SVM-based model. Figure 3.2 depicts the enhanced tagging algorithm of MorphTagger. More details about the combination of the character-based and analyzer-based models can be found in [Mansour 08].

As for MADA, the text input to the SMT system is again obtained by further preprocessing of the MorphTagger output. Here, we split off the class of conjunction clitics (*w+* and *f+*), the class of particles (*l+*, *k+*, *b+* and *s+*) and all pronominal clitics, i.e. we employ MADA's ATB scheme. Otherwise, no normalization is used.

3.4 Summary

As for any language which is to be translated via SMT, the Arabic text input must be tokenized first, i.e. words and punctuation marks (except for abbreviations) are separated. We then normalize the text by removing the entire diacritics and all articles which are supposed to be attached to an adjective. Furthermore, SMT from Arabic to English becomes a challenging task because Arabic has a large set of morphological features such as gender, number, person or voice, which have to be distinguished from the set of attachable clitics, e.g. the definite article "the" ال (*Al+*) , or the class of particle proclitics "to/for" لِ (*l+*), "by/with" بِ (*b+*), "as/such" كَ (*k+*), etc. The question whether to split off a clitic or feature or whether to simply abstract it off is dependent on the context. We investigate three different methods for resolving

```
  INPUT: input sentence s
0 produce analyses for each word in s using the morphological analyzer
  combined with the corpus analyses
1 calculate lexical and contextual probabilities using available
  annotated corpora (ML estimation)
2 run Viterbi's algorithm for HMM disambiguation and calculate
  a rank which is composed from the probability given by the model
  and the length of the sentence
3 IF rank > threshold THEN
4     OUTPUT: tagging
5 ELSE
6     run the character-based model over the sentence and add the new
      analyses generated
7 combine the analyses generated by the morphological analyzer and the
  character-based model, update the lexicon probabilities and rerun the
  model
```

Figure 3.2. Enhanced tagging algorithm of *MorphTagger* (from [Mansour & Sima'an$^+$ 07]).

this ambiguity in Arabic word segmentation. In Chapter 9, we present experimental findings in terms of corpus statistics as well as translation performance for two Arabic-English translation tasks (including multi-domains). More precisely, we compare:

- *IFSA*:
  the improved finite state automaton-based approach as described in Subsection 3.3.1,
- *MADA*:
  the method applying the disambiguation tool, and here we further analyze the :
  - *D2* scheme,
  - *ATB* scheme,
  - normalization of *Yaa* and *Alef*,
  - effect of the orthographical normalization module for word stems,
- *MorphTagger*:
  the word segmentation inferred from POS tags.

Chapter 4

Search for Phrase-Based Translation

In this chapter, we address the search problem in machine translation. According to [Ney 01], modeling, training, and search compose the three problems one has to deal with in SMT. The challenge in modeling is to capture the dependencies of the source and target language sentences. Once set up, the free model parameters are estimated from bilingual data during training. The actual translation problem, i.e. the search, finally deals with finding the best target language translation among all possible translation hypotheses. Initially, we briefly review the models of our decoder as they have a direct impact on the search problem. We give a detailed description of the search algorithm, make an explicit distinction between reordering and lexical hypothesis in the pruning process, and analyze different reordering constraints. Finally, we describe the generation of word graphs and n-best lists which are the input for the second pass reranking (cf. Chapter 5). As already mentioned in Section 1.3, this work bases on our phrase-based decoder which was described in detail by [Zens 08]. Here, we address the specific requirements of robust MT and multi-domain MT, and thereby focus on Arabic-English translation tasks. For a description of the entire decoder, the reader is referred to [Zens 08].

4.1 Motivation

Search (or decoding or generation) is the task of finding the target language sentence e_1^I that maximizes the posterior probability given the source sentence f_1^J. Here, we consider the MAP decision rule for the direct phrase-based approach (cf. Equation 1.13):

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \max_{K, s_1^K} \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K; f_1^J) \right\}. \tag{4.1}$$

We have to carry out a maximization over all possible target sentences e_1^I and over all possible segmentations s_1^K. Simply enumerating all target sentences is practically infeasible, thus we have to decide on:

- the number of phrases K,
- the phrase segmentation s_1^K of the source sentence, and
- the phrase translations $\tilde{e}$ for each source phrase $\tilde{f}$.

Moreover, we can exploit the structural properties of the models. We can interpret the search as a sequence of decisions $(\tilde{e}_k, b_k, j_k)$ for $k = 1, \ldots, K$. At each step, we choose a source phrase $\tilde{f}_k$ identified by its start and end positions $s_k = (b_k, j_k)$ and the corresponding translation $\tilde{e}_k$. To ensure that there are no gaps and no overlap, we keep track of the set of source positions

that are already translated ('covered'). We call this the *coverage* set $C \subseteq \{1, \ldots, J\}$ and refer to the number of covered source positions of a hypothesis as its cardinality c. Furthermore, the search space can be considered a graph where the edges are labeled with the decisions $(\tilde{e}, j, j')$ and the nodes are labeled with the coverage sets C. The initial state is labeled with the empty coverage set, i.e. no source word is translated yet. The goal state corresponds to the full coverage $C = \{1, \ldots, J\}$. Each path through this graph corresponds to a possible translation of the source sentence, simply by concatenation of the target phrases $\tilde{e}$ along the path. Hence, the search problem can also be interpreted as finding the optimal path through this graph. We use a beam search strategy [Jelinek 98] and apply dynamic programming [Bellman 57] to tackle the search problem.

4.2 Models Used During Search

As described in Section 1.1.2, we use a log-linear combination of several models in search. In this section, we list the models (also called feature functions) used during search, i.e. in the first pass.

- Phrase-based model:
  During decoding, the hypotheses are generated by concatenating target language phrases. The pairs of source and target phrases that are consistent with the word alignment are extracted from the bilingual training corpus [Zens & Och$^+$ 02]. Then, we use relative frequencies to estimate the phrase translation probabilities:

  $$p(\tilde{f}|\tilde{e}) = \frac{N(\tilde{f}, \tilde{e})}{N(\tilde{e})} \,. \tag{4.2}$$

  $N(\tilde{f}, \tilde{e})$ denotes the number of co-occurrences of a phrase pair $(\tilde{f}, \tilde{e})$ that are consistent with the word alignment. The resulting feature function is:

  $$h_{\mathrm{Phr}}(e_1^I, s_1^K; f_1^J) \;=\; \sum_{k=1}^{K} \log p(\tilde{f}_k|\tilde{e}_k) \,. \tag{4.3}$$

  To obtain a more symmetric model, the phrase-based model is used for both translation directions. The inverse feature function is analogously:

  $$h_{\mathrm{iPhr}}(e_1^I, s_1^K; f_1^J) \;=\; \sum_{k=1}^{K} \log p(\tilde{e}_k|\tilde{f}_k) \,. \tag{4.4}$$

- Phrase count features:
  Rare phrases tend to be overestimated and thus errors can originate from erroneous translations in the training data and misaligned words. We include features based on the counts of phrase pairs. As feature, we use an indicator whether the joint count of a phrase pair $N(\tilde{f}, \tilde{e})$ is below a threshold τ:

  $$h_{\mathrm{C},\tau}(e_1^I, s_1^K; f_1^J) \;=\; \sum_{k=1}^{K} [N(\tilde{f}_k, \tilde{e}_k) \leq \tau] \,. \tag{4.5}$$

  $[\cdot]$ denotes a true or false statement [Graham & Knuth$^+$ 94]. In the experiments, we use three phrase count features with manually chosen thresholds 1.0, 2.0 and 3.0 (each with its own model scaling factor).

- Word-based lexicon model:
  Long phrases are rare and therefore tend to be overestimated. We use a word-based lexicon model to smooth the phrase translation probabilities. The computation is similar to IBM model 1 but takes into account only the words within the phrase pair:

$$h_{\text{Lex}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \log \left(p(f_j|e_0) + \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \right) . \tag{4.6}$$

  Here, e_0 denotes the empty word and we use the IBM model 4 lexicon trained with GIZA++ as word translation probabilities $p(f|e)$. Analog to the phrase-based model, this model is also used in the inverted translation direction:

$$h_{\text{iLex}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{i=i_{k-1}+1}^{i_k} \log \left(p(e_i|f_0) + \sum_{j=b_k}^{j_k} p(e_i|f_j) \right) . \tag{4.7}$$

  Alternatively, the word-based lexicon can be modeled as a *noisy-OR* gate [Pearl 88], i.e. a disjunctive interaction having multiple independent causes. Here, the motivation is that different target words e can generate the same source word f. The resulting feature functions for both directions are:

$$h_{\text{Nor}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \log \left(1 - \prod_{i=i_{k-1}+1}^{i_k} (1 - p(f_j|e_i)) \right) , \tag{4.8}$$

$$h_{\text{iNor}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{i=i_{k-1}+1}^{i_k} \log \left(1 - \prod_{j=b_k}^{j_k} (1 - p(e_i|f_j)) \right) . \tag{4.9}$$

- Deletion model:
  For each source word, we use a feature indicating whether a target word with a probability higher than a given threshold τ exists. Otherwise, the word is considered a deletion. The model simply counts the number of deletions:

$$h_{\text{Del}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \prod_{i=i_{k-1}+1}^{i_k} [\, p(f_j|e_i) < \tau \,] . \tag{4.10}$$

  Analog to the word-based lexicon model, we use the GIZA++ IBM model 4 lexicon as word translation probabilities $p(f|e)$ and apply the deletion model for the inverse direction as well:

$$h_{\text{iDel}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \sum_{i=i_{k-1}+1}^{i_k} \prod_{j=b_k}^{j_k} [\, p(e_i|f_j) < \tau \,] . \tag{4.11}$$

  [Och & Gildea$^+$ 03] presented a deletion model in a rescoring/reranking framework in order to penalize hypotheses that miss the translation of a word. Here, we use a within-phrase variant already during search.

- Word and phrase penalty model:
  These two models are simple heuristics influencing the average sentence and phrase lengths and can thus be used to enable the decoder to generate longer or shorter translation candidates:

$$h_{\mathrm{WP}}(e_1^I, s_1^K; f_1^J) = I\,, \tag{4.12}$$
$$h_{\mathrm{PP}}(e_1^I, s_1^K; f_1^J) = K\,. \tag{4.13}$$

  The penalties result in a constant cost per produced word and phrase. The word penalty was already proposed in [Brown & Della Pietra+ 93] to counteract the general preference for shorter translations.

- Target language model:
  We apply a standard n-gram language model which is built using the SRI Language Modeling toolkit [Stolcke 02] (smoothing technique is modified Kneser-Ney discounting with interpolation [Kneser & Ney 95, Chen & Goodman 98]):

$$h_{\mathrm{LM}}(e_1^I, s_1^K; f_1^J) = \sum_{i=1}^{I+1} \log p(e_i | e_{i-n+1}^{i-1})\,. \tag{4.14}$$

  Note the $(I+1)^{\mathrm{th}}$ term in the sum: the language model includes a sentence end probability for which we have defined e_{I+1} as sentence boundary marker.

- Distortion penalty model:
  This reordering model assigns costs simply based on the number of word positions skipped from the end position of a phrase to the start position of the next phrase. The distance is also called distortion, and the model has also been used in, e.g. [Bender & Zens+ 04]:

$$h_{\mathrm{Dist}}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K+1} q_{\mathrm{Dist}}(b_k, j_{k-1}) \tag{4.15}$$

  with

$$q_{\mathrm{Dist}}(j, j') = |j - j' + 1|\,. \tag{4.16}$$

  The distortion penalty model assigns costs of zero to a translation which is monotone at the phrase level. The more phrases are reordered, the higher the distortion penalty. Often it is combined with a limit D on the jump width:

$$q_{\mathrm{Dist}}(j, j') = \begin{cases} |j - j' - 1| & \text{if} |j - j' - 1| < D \\ \infty & \text{else} \end{cases}\,. \tag{4.17}$$

Except for the language model and distortion penalty model, all the models above are defined on the level of single phrases, i.e. they have no dependencies across phrase boundaries and thus can be computed for each phrase pair without context information. We pack these models together and call them *phrase* models. With respect to the sequence of decisions $(\tilde{e}_k, b_k, j_k)$, the phrase models do not depend on the decisions taken so far. According to [Zens 08], $q_{\mathrm{TM}}(\tilde{e}_k, b_k, j_k)$ denotes the weighted sum of all phrase model scores for this decision (also called *state*). The language model on the other hand depends on the last $(n-1)$ words of the target sentence,

and the distortion penalty model depends on the end position of the previous phrase. We want to compute the scores of individual states and therefore introduce state copies and distinguish them according to their history. We use $\tilde{e}' \oplus \tilde{e}$ to denote the language model history after expanding the given history $\tilde{e}'$ with the phrase $\tilde{e}$. The score of this language model expansion $q_{\mathrm{LM}}(\tilde{e}|\tilde{e}')$ is computed as follows:

$$q_{\mathrm{LM}}(\tilde{e}|\tilde{e}') = \lambda_{\mathrm{LM}} \cdot \sum_{i=1}^{|\tilde{e}|} \log p(\tilde{e}^i|\tilde{e}^{i-1}, \ldots, \tilde{e}^1, \tilde{e}') \,. \tag{4.18}$$

Here, $\tilde{e}^i$ denotes the ith word of the phrase $\tilde{e}$. Accordingly, we use $q_{\mathrm{DM}}(j, j')$ to denote the weighted score of a jump from source position j to source position j':

$$q_{\mathrm{DM}}(j, j') = \lambda_{\mathrm{Dist}} \cdot |j - j' + 1| \,. \tag{4.19}$$

Obviously, $(\tilde{e}, j, j')$ can occur multiple times during the search. To avoid repeated computations, we determine the set of possible target phrases for all source phrases before the actual search and store the target phrases along with their scores in a table $E(j, j')$.

Summarizing, the states in the search space can be identified by a triple $(C, \tilde{e}, j)$, where C denotes the coverage set, $\tilde{e}$ denotes the language model history, and j denotes the end position of the last source phrase. During search, a translation hypothesis is expanded by computing the successor states for the current state $(C, \tilde{e}, j)$. The expansion with a phrase pair $(\tilde{e}', j'', j')$ yields the successor state $(C \cup \{j'', \ldots, j'\}, \tilde{e} \oplus \tilde{e}', j')$. The corresponding score is calculated as:

$$q_{\mathrm{TM}}(\tilde{e}', j'', j') + q_{\mathrm{LM}}(\tilde{e}'|\tilde{e}) + q_{\mathrm{DM}}(j, j'') \,. \tag{4.20}$$

We have to ensure that there is not overlap, i.e. $C \cap \{j'', \ldots, j'\} = \emptyset$.

4.3 Search Algorithms and Pruning Strategies

Given the definitions of the previous section and treating the search space as a graph, the search problem can be considered as finding the optimal path through the graph. As the size of the search graph is exponential in the source sentence length and as it has been shown by [Knight 99] that the search problem is NP-hard, we have to use approximations to find a solution efficiently. Using dynamic programming (DP), i.e. first computing small subproblems and then assembling the solution for the whole problem from these subproblems, we reduce the number of paths that we have to explore in the search graph. The idea of beam search is to keep the promising candidates and to discard hypotheses that are unlikely to yield the optimal solution. Beam search may generate suboptimal solutions. In the following, we make an explicit distinction between *reordering* and *lexical* hypotheses:

- **Coverage hypothesis** C. We use the term coverage hypothesis to refer to the set of all lexical hypotheses with the same coverage C.
- **Lexical hypothesis** $(C, \tilde{e}, j)$. A lexical hypothesis is identified by a coverage C, a language model history $\tilde{e}$ and a source sentence position j.

The number of coverage or more intuitive reordering hypotheses indicates how many alternative reorderings per cardinality are investigated during the search. The number of lexical hypotheses per reordering hypothesis indicates the lexical alternatives that are taken into account.

```
    INPUT: source sentence f_1^J, translation options E(j, j') for 1 ≤ j ≤ j' ≤ J,
           models q_TM(·) and q_LM(·)
 0  Q(0, $) = 0 ; all other Q(·, ·) entries are initialized to −∞
 1  FOR j = 1 TO J DO
 2      FOR j' = max{0, j − L_s} TO j − 1 DO
 3          FOR ALL LM histories ẽ' DO
 4              FOR ALL target phrases ẽ'' DO
 5                  score = Q(j', ẽ') + q_TM(ẽ'', j + 1, j') + q_LM(ẽ''|ẽ')
 6                  ẽ = ẽ' ⊕ ẽ''
 7                  IF score > Q(j, ẽ)
 8                  THEN Q(j, ẽ) = score
 9                          B(j, ẽ) = (j', ẽ')
10                          A(j, ẽ) = ẽ''
```

Figure 4.1. Monotone search algorithm for phrase-based translation (from [Zens 08]).

4.3.1 Monotone Search

If we prohibit phrase rearrangements during translation, both source and target sentence are processed in monotone order. There is no reordering of the phrases and the distortion penalty model becomes unnecessary. As a consequence, the *m*onotone search problem can be solved efficiently using DP. Accordingly for the MAP optimization problem (cf. Equation 4.1), we define the quantity $Q(j, \tilde{e})$ as the maximum score of a phrase sequence that ends with the language model history $\tilde{e}$ and covers positions 1 to j of the source sentence, and obtain the following DP recursion:

$$Q(0, \$) = 0 \tag{4.21}$$

$$Q(j, \tilde{e}) = \max_{\substack{j' : j - L_s \leq j' < j \\ \tilde{e}'', \tilde{e}' : \tilde{e}' \oplus \tilde{e}'' = \tilde{e}}} \left\{ Q(j', \tilde{e}') + q_{\mathrm{TM}}(\tilde{e}'', j' + 1, j) + q_{\mathrm{LM}}(\tilde{e}''|\tilde{e}') \right\} \tag{4.22}$$

$$\hat{Q} = \max_{\tilde{e}} \left\{ Q(J, \tilde{e}) + q_{\mathrm{LM}}(\$; \tilde{e}) \right\}. \tag{4.23}$$

Here, the \$ symbol denotes the sentence boundary marker and L_s denotes the maximum phrase length in the source language. During the search, we store back-pointers to the previous best decision $B(\cdot, \cdot)$ and to the maximizing arguments $A(\cdot, \cdot)$. After the search, the back-pointers are used to trace back the best decisions and generate the translation. The resulting complexity is linear in the length of the source sentence and thus allows for a very efficient implementation. The pseudo code for the monotone search algorithm for phrase-based translation is depicted in Figure 4.1. $E(j', j)$ denotes the set of possible phrase translations of the source phrase $\tilde{f} = f_{j'}, ..., f_j$.

The drawback is that reordering is only possible within the phrases (solely the sequence of phrases is enforced to be monotone). However, the monotone search algorithm is suitable for language pairs which have a similar word order, such as French-English, Spanish-English or also Arabic-English. In Chapter 9, we experimentally show this.

4.3.2 Non-Monotone Search

If we tackle the translation problem for language pairs with different word order, e.g. Chinese-English or Japanese-English, the monotone search is inappropriate. Instead, we need to explicitly permit reordering. [Tillmann & Ney 03] described a non-monotone search algorithm for single-word based translation. The idea is that the search proceeds synchronously with the number of already translated source positions. Therefore, they call their algorithm source cardinality synchronous search. Here, we use a phrase-based version of their method. We generate the translation phrase by phrase, i.e. the search is monotone in the target language. To permit reordering, we allow to jump forth and back within the source sentence. As one constraint of the phrase-based approach is that each source position is translated by exactly one target phrase, we have to keep track of the source positions already translated. Recapitulate that this corresponds to the coverage $C \subseteq \{1, ..., J\}$.

In contrast to the monotone search, the auxiliary quantity for the DP recursion now also depends on the coverage set. $Q(C, \tilde{e}, j)$ denotes the maximum score of a path leading from the initial state to the state $(C, \tilde{e}, j)$:

$$Q(\emptyset, \$, 0) = 0 \tag{4.24}$$

$$Q(C, \tilde{e}, j) = \max_{\substack{j'', j': j' \leq j < j' + L_s \wedge \{j', ..., j\} \subseteq C \\ \tilde{e}', \tilde{e}'': \tilde{e}' \oplus \tilde{e}'' = \tilde{e}}} \Big\{ Q(C \setminus \{j', ..., j\}, \tilde{e}', j'') + q_{\mathrm{TM}}(\tilde{e}'', j', j) + q_{\mathrm{LM}}(\tilde{e}''|\tilde{e}') + q_{\mathrm{DM}}(j'', j') \Big\} \tag{4.25}$$

$$\hat{Q} = \max_{\tilde{e}, j} \Big\{ Q(\{1, ..., J\}, \tilde{e}, j) + q_{\mathrm{LM}}(\$|\tilde{e}) + q_{\mathrm{DM}}(j, J+1) \Big\} . \tag{4.26}$$

As for the monotone search, we store back pointers $B(\cdot)$ to the previous best decision as well as the maximizing arguments $A(\cdot)$. For each cardinality c, we have to iterate over all possible source phrase lengths l. Then, we iterate over the possible predecessor coverages C' with cardinality $c - l$. Next, we have to select a source phrase $\tilde{f} = f_j, ..., f_{j+l}$ by choosing the start position j. Eventually, we consider all existing predecessor states $\tilde{e}', j'$ and all translation options $\tilde{e}'' \in E(j, j+l)$ to compute the score of the expansion. In Figure 4.2, we illustrate the search. For each cardinality, we have a list of coverage hypotheses, here represented as boxes. For each coverage hypothesis, we have a list of lexical hypotheses, here represented as circles. We generate a specific lexical hypothesis (the black circle) with cardinality c by expanding shorter hypotheses. The hypotheses with cardinality $c - 1$ are expanded with one-word phrases, the hypotheses with cardinality $c - 2$ are expanded with two-word phrases etc. Consequently, all generated hypotheses have the same cardinality which allows for very efficient recombination and pruning. In contrast, e.g. [Koehn & Hoang$^+$ 07] or [Tillmann 06] expand hypotheses with cardinality c into higher cardinalities.

So, in principle we have to loop over all possible coverage sets which yields an exponential complexity of the non-monotone search algorithm, taking into account that $\sum_{c=0}^{J} \binom{J}{c} = 2^J$. To make the translation process manageable, we resort to a beam search strategy and apply pruning at several levels.

4.3.3 Pruning

Our decoder implements two variants of pruning: threshold pruning (or beam pruning) and histogram pruning [Steinbiss & Tran$^+$ 94]. Threshold pruning secures that only hypotheses are

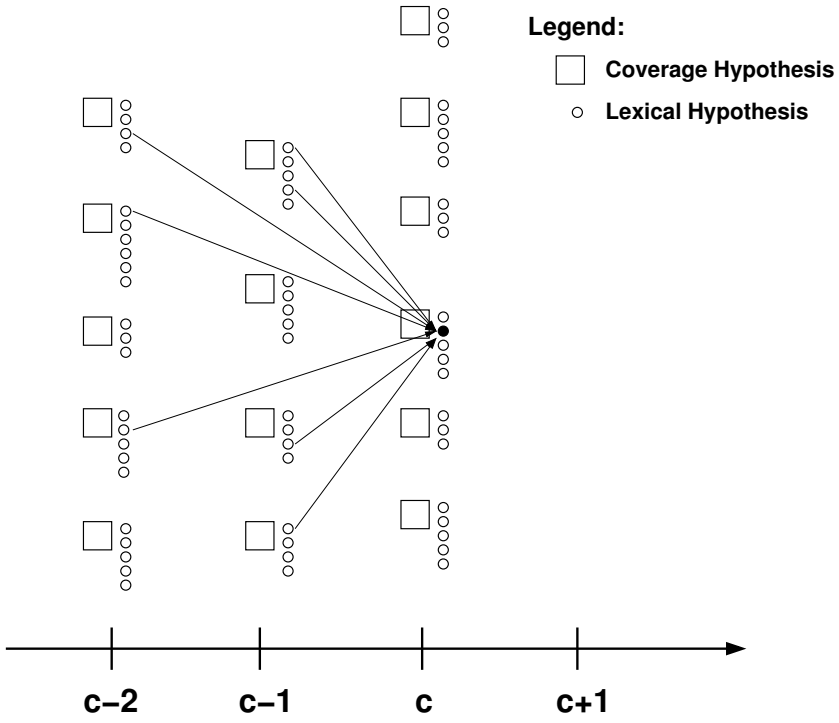

Figure 4.2. Illustration of the (non-monotone) search process. For each cardinality, we have a list of coverage hypotheses (boxes). For each coverage hypothesis, we have a list of lexical hypotheses (circles). A hypothesis with cardinality c can be generated by expanding a hypothesis of cardinality $c-1$ with a one-word phrase, by expanding a hypothesis of cardinality $c-2$ with a two-word phrase, and so on.

kept whose scores are close to the best one. A drawback is that threshold pruning affects the beam size only indirectly and that there is no upper limit on the number of hypotheses in the beam. Histogram pruning on the other hand means that only the best N hypotheses are kept and is thus a very simple way of limiting the beam size. In this work, we mainly use histogram pruning. Threshold pruning is only applied in some cases, e.g. to tune the system in terms of translation speed for the interactive MT setup.

In the search process, we make an explicit distinction between *reordering* and *lexical* pruning [Bender & Zens$^+$ 09]:

- **Reordering pruning:**
  The number of reordering hypotheses per cardinality c is limited. If we prune a reordering hypothesis, we remove all associated lexical hypotheses.

- **Lexical pruning:**
  The number of lexical hypotheses per reordering hypothesis is limited.

More precisely, we use the following pruning strategies:

- **Observation pruning:**
  Here, we limit the number of phrase translation candidates per source phrase. This is done before search. We apply observation histogram pruning with parameter N_O. Thus,

if there are more than N_O target phrases for a particular source phrase, then we keep only the top N_O candidates. In practice, $N_O = 50$ already achieves good results while keeping the phrase tables handy, especially for large-scale tasks as GALE.

- **Coverage pruning per cardinality:**
  Here, we consider all coverage hypotheses with a given cardinality c. $Q(C)$ is defined as the maximum score of any hypothesis with coverage C (cf. Equation 4.28), and is used here as score of the coverage hypothesis C. During the pruning, we compare hypotheses which cover different parts of the source sentence. Hence, it is important to use a rest score estimate for completing these hypotheses. Without such a rest score estimate, the search would first focus on the easy-to-translate part of the source sentence. In this work, we deploy a rest score estimate $R(C, j)$ including translation, language and distortion model computed on sequences of source positions. A detailed description of the rest score estimate is given in [Zens 08]. The rest cost estimate is of special importance for smaller beam sizes. Let τ_c denote the pruning threshold, then we keep a coverage hypothesis with score $Q(C)$ if
  $$Q(C) + \tau_C \geq \max_{\substack{C:|C|=c,\\ \tilde{e},j}} \left\{ Q(C, \tilde{e}, j) + R(C, j) \right\} . \tag{4.27}$$
  Generally, we apply histogram pruning with parameter N_C. Thus, if there are more than N_C coverage hypotheses for a particular cardinality c, we keep only the top N_C candidates. Recapitulate that if we prune a coverage hypothesis C, we remove all lexical hypotheses with coverage C. In practice, N_C is in the range of 50 to 500.

- **Lexical pruning per coverage:**
  Here, we consider all lexical hypotheses that share the same coverage C, differing for instance in their language model history $\tilde{e}$ or the end position of the last phrase j. We therefore include the rest cost estimate $R(C, j)$ mainly for the distortion model. Let τ_L denote the pruning threshold and let $Q(C)$ denote the maximum score of any hypothesis with coverage C:
  $$Q(C) = \max_{\tilde{e},j} \left\{ Q(C, \tilde{e}, j) + R(C, j) \right\} . \tag{4.28}$$
  Then, we keep a hypothesis with score $Q(C, \tilde{e}, j)$ if:
  $$Q(C, \tilde{e}, j) + R(C, j) + \tau_L \geq Q(C) . \tag{4.29}$$
  Generally, we apply histogram pruning with parameter N_L. Thus, if there are more than N_L hypotheses for a particular coverage C, then we keep only the top N_L candidates. In practice, N_L is in the range of 5 to 50.

Due to histogram pruning, the computational complexity of the non-monotone search is now linear in the sentence length (although encountering a rather large constant factor). On the other hand, the search is now no longer guaranteed to find the (globally) optimal translation candidate. However, the experimental results show that the pruning strategies cause almost no loss in translation quality. Chapter 9 also stresses the importance of the coverage pruning to adjust the balance between hypotheses representing different reorderings (coverage hypotheses) and hypotheses with different lexical representations.

Putting everything together, the resulting pseudo code for the non-monotone search algorithm including pruning is given in Figure 4.3. The basic concept of the algorithm is the same as for

```
    INPUT: source sentence f_1^J, translation options E(j, j') for 1 ≤ j ≤ j' ≤ J,
           models q_TM(·), q_LM(·) and q_DM(·)
 0  Q(∅, $, 0) = 0 ; all other Q(·, ·, ·) entries are initialized to −∞
 1  FOR cardinality c = 1 TO J DO
 2      IF c > L_s THEN purgeCardinality c − L_s − 1
 3      FOR source phrase length l = 1 TO min{L_s, c} DO
 4          FOR ALL coverages C' ⊂ {1, ..., J} : |C'| = c − l DO
 5              FOR ALL start positions j ∈ {1, ..., J} : C' ∩ {j, ..., j + l} = ∅ DO
 6                  coverage C = C' ∪ {j, ..., j + l}
 7                  FOR ALL states ẽ', j' ∈ Q(C', ·, ·) DO
 8                      partial score q = Q(C', ẽ', j') + q_DM(j', j)
 9                      IF q + R(C, j + l) + q_TM(j, j + l) isTooBadForCoverage C
10                      THEN CONTINUE
11                      FOR ALL phrase translations ẽ'' ∈ E(j, j + l) DO
12                          partial score q' = q + R(C, j + l) + q_TM(ẽ'', j, j + l)
13                          IF q' isTooBadForCoverage C THEN BREAK
14                          score = q + q_TM(ẽ'', j, j + l) + q_LM(ẽ''|ẽ')
15                          IF score+R(C, j + l) isTooBadForCoverage C THEN CONTINUE
16                          language model state ẽ = ẽ' ⊕ ẽ''
17                          IF score > Q(C, ẽ, j + l)
18                          THEN Q(C, ẽ, j + l) = score
19                                   B(C, ẽ, j + l) = (C', ẽ', j')
20                                   A(C, ẽ, j + l) = ẽ
21     pruneCardinality c
```

Figure 4.3. Detailed non-monotone search algorithm for phrase-based translation (from [Zens 08]).

the monotone search. A key challenge is to perform as much computations as possible outside the inner loops and to apply pruning wherever applicable. The function 'pruneCardinality c' applies coverage and cardinality pruning after all hypotheses with the current cardinality c have been generated (line 21). In the function 'purgeCardinality c', we free the memory (except for the trace back information) of all hypotheses with cardinality c. For example, the coverage sets and the LM histories are not needed anymore and thus the memory can be reused. Furthermore, we can stop the expansion whenever it is clear that the resulting hypotheses would be pruned anyway. This is done via the 'x isTooBadForCoverage C' function. As the translation options $E(\cdot,\cdot)$ are sorted once before the search, we can process the hypotheses according to their score and check:

- if the partial score plus rest cost estimate plus a defined estimation for the translation model score is too bad, we can skip all of the possible phrase translations (line 9),
- if the partial score without LM is already too bad, we can omit the LM score computation (line 13),

- if the score of the expansion is too bad, we can ignore to check for recombination (line 15).

4.4 Reordering Constraints

In the previous section, we covered search algorithms for phrase-based translation and presented pruning strategies to approach the exponential complexity of the non-monotone search. Another way to reduce the search space is to constrain the number of word reorderings and thereby reduce the number of translation candidates that have to be expanded during search. Contrary to the theoretical expectation, constrained reorderings are even superior to unconstrained reorderings in practice due to less search errors (the search problem is simplified) and unreliable probability estimates for the unconstrained case. In this section, we describe the two types of reordering constraints implemented in our decoder.

4.4.1 IBM Constraints

The so-called *IBM constraints* were first formulated for single-word based translation models by [Berger & Brown$^+$ 96] and are based on permutations with restricted displacement [Lehmer 70]. At the beginning, each position in the source sentence is marked uncovered. We then process the source positions from left to right and mark all covered ones. According to the IBM constraints, we are allowed to skip a position and come back to it at a later step but the next position must be one of the first k uncovered positions. Doing so, there are no more than $k-1$ skipped positions at any step. Let $C \subseteq \{1, ..., J\}$ denote the coverage, then it violates the IBM constraints, if

$$|C| + k \leq \max C \,. \tag{4.30}$$

Typically, k is set to 4. [Tillmann & Ney 00, Tillmann & Ney 03] presented an efficient implementation to apply the IBM constraints for single-word based models and gave the illustration shown in Figure 4.4. The source sentence positions are plotted along the x-axis. Yet uncovered positions are marked with unfilled circles, already covered positions with filled circles, and the uncovered positions that are candidates for extension are marked with unfilled squares.

In our decoder, we use an extension of the method of [Tillmann & Ney 00, Tillmann & Ney 03]. Instead of allowing to skip $k-1$ word positions, we allow to skip up to $k-1$ word blocks. This enables the IBM constraints to be used on a phrase basis. The gaps are then filled with phrases (multiple phrases per gap are permitted). For a coverage C, we check if the number of gaps is less than k:

$$\big|\{j > 1 | j \in C \wedge j-1 \notin C\}\big| < k \,. \tag{4.31}$$

The test whether the coverage C violates the IBM constraints can be easily integrated into the beam search algorithm, e.g. in line 6 of our search algorithm in Figure 4.3. This reduces the number of translation hypotheses that must be expanded and thus speeds up the search.

4.4.2 ITG Constraints

The *ITG constraints* were introduced by [Wu 95]. Some of the originally intended applications were, for instance, Chinese word segmentation and sentence splitting into sub-sentential chunks.

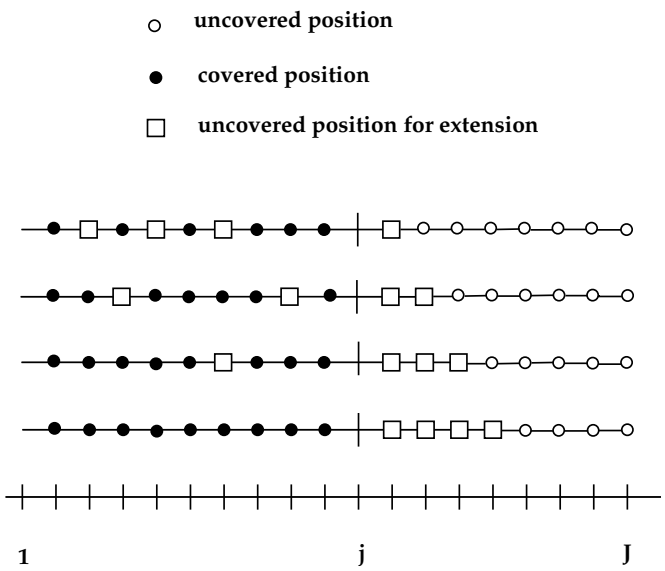

Figure 4.4. Illustration of the IBM constraints (from [Tillmann & Ney 00]).

ITG stands for *inversion transduction grammars*. Later, the ITG constraints were also used for SMT [Wu 96, Wu & Wong 98] and bilingual parsing [Wu 97].

[Wu 95] interpreted the input sentence as a sequence of blocks. At the beginning, each position is a block of its own. Then, we select two consecutive blocks and merge them to a single block by choosing between two options: either keep the target blocks in monotone order or invert the order. The concept is illustrated in Figure 4.5. The dark boxes represent the two blocks to be merged. Once two blocks are merged, they are treated as a single block and only can be merged further as a whole. It is not allowed to merge one of the sub blocks again.

[Wu 97] showed a polynomial-time search algorithm for the ITG constraints which is based on dynamic programming and which is similar to the CYK-parsing algorithm [Kasami 65, Younger 67]. However, the CYK-style algorithm affects the language model recombination adversely as the LM history has to be considered at the beginning and at the end of each interval. Hence, a phrase by phrase generation of the target sentence is not possible. Instead, we kept up the phrase-based approach and modified our beam search decoder such that it produces only translation hypotheses that are confirm with the ITG constraints. We re-organized the coverage in form of a stack of spans (a contiguous sequence of source positions) where each of them is ITG parsable. Furthermore, we relaxed the ITG constraints to the so-called Baxter permutations [Baxter 64] which are a superset of the ITG permutations. This relaxation avoids computational overhead when constructing the stack of spans. For a detailed description of the algorithm to test if a permutation is ITG parsable and of the actual integration into the beam search algorithm, the reader is referred to [Zens 08].

4.5 Generation of Word Graphs and n-Best Lists

So far, the algorithms and models presented in this section were only designed for the first pass in MT, i.e. the generation of the single best translation hypotheses. For some applications, we are also interested in multiple translation candidates. Examples include the optimization of the

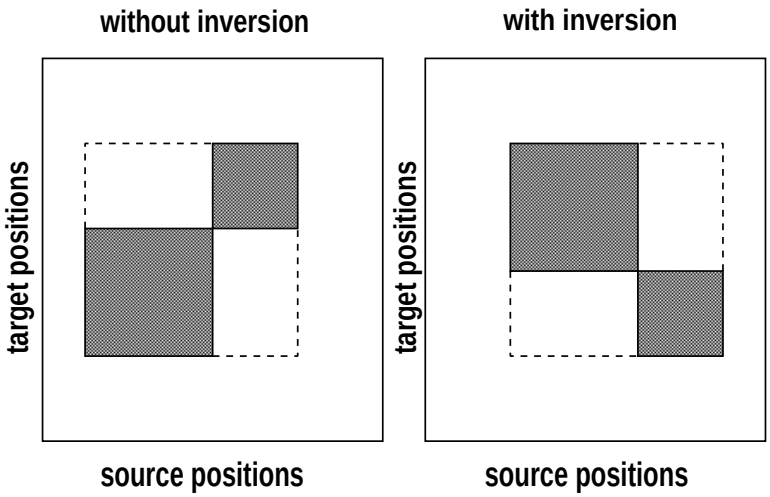

Figure 4.5. Illustration of the ITG reordering constraints: monotone and inverted concatenation of two consecutive blocks.

model scaling factors using the MERT framework of [Och 03], estimation of confidence measures [Ueffing 05], or MBR decoding [Kumar & Byrne 04]. The most prevalent use of multiple translation candidates is within the reranking framework in the second pass in MT. Here, the motivation is to further increase the quality of the MT output by using additional models which are not applicable during the first pass. Chapter 5 addresses the reranking framework in MT with the focus on the incorporation of structural knowledge into the translation framework.

Multiple translation candidates are usually represented as word graphs (also called lattices) or n-best lists. Their generation for single-word based translation models was described by [Ueffing & Och$^+$ 02] and for phrase-based models in [Koehn 03, Zens & Ney 05, Hasan & Zens$^+$ 07]. The actual search is very similar to the single-best method: instead of deleting the back pointers to the recombined hypotheses we store them, resulting in possibly multiple incoming edges to the nodes in the search graph. In contrast, each node has exactly one incoming edge for the single-best search. Thus, we do not generate a tree but a graph of translation hypotheses. Given this graph, multiple ways there exist to extract the n best translation candidates:

- apply the methods of standard finite state toolkits, e.g. [Kanthak & Ney 04][1],
- use the extension of the A^* algorithm as described in [Ueffing & Och$^+$ 02], or
- use an implementation of the k shortest path algorithm, e.g. [Eppstein 01].

In this thesis, we follow [Ueffing & Och$^+$ 02] and use the A^* algorithm.

[1]The RWTH FSA toolkit is an efficient and flexible toolkit to create, manipulate and optimize finite-state automata, and is publicly available: `http://www-i6.informatik.rwth-aachen.de/$\sim$kanthak/fsa.html`.

4.6 Summary

In SMT, modeling, training and search compose the three problems one has to deal with according to [Ney 01]. The focus of this chapter was on the search problem, yet we briefly reviewed the models which are directly applied in the search. These models consist of phrase models which have no dependencies across phrase boundaries. Therefore, they can be computed for each phrase pair without considering the context information, the language model and the distortion penalty model. The latter two have dependencies across phrase boundaries. We showed how to organize the search space in order to allow for an efficient phrase by phrase generation of the translation hypotheses. We then gave a detailed description of the DP search algorithms implemented in our decoder. As the complexity of the (non-monotone) search is exponential in the source sentence length (it has been shown by [Knight 99] that the search problem is NP-hard), we have to use approximations to find a solution efficiently. On the one hand, we presented pruning strategies to approach the exponential complexity, on the other hand, we constrain the number of word reorderings. Here, our decoder provides the IBM constraints and the ITG constraints to reduce the number of translation candidates that must be expanded during search.

In the pruning process, we make an explicit distinction between reordering (coverage) and lexical hypotheses. Chapter 9 stresses the importance of the coverage pruning to adjust the balance between hypotheses representing different reorderings (coverage hypotheses) and hypotheses with different lexical representations. To be specific, we use the following (histogram) pruning strategies:

- observation pruning,
- lexical pruning per coverage, and
- coverage pruning per cardinality.

Finally, we sketched the generation of multiple translation candidates for further processing, e.g. within the reranking framework. We follow [Ueffing & Och$^+$ 02] to generate a graph of translation hypotheses and use their extension of the A^* algorithm to extract the n best translation candidates from this graph.

Chapter 5

Reranking Translation Hypotheses Using Structural Properties

At the end of the previous section, we briefly sketched how to generate multiple translation candidates during search, i.e. the first pass in MT. In this section, the focus is on the the second pass in MT, i.e. the reranking framework to further increase the MT quality by exploiting these hypotheses. We pay particular attention to the incorporation of structural knowledge into the translation framework. In the following, we introduce the main methodologies used for deriving syntactic dependencies on words or word groups, namely supertagging/lightweight dependency analysis (LDA), link grammars and maximum entropy based chunking. These contributions were published in [Hasan & Bender$^+$ 06].

5.1 Motivation

Though much better than traditional rule-based approaches, statistically driven MT systems still make a lot of errors that seem, at least from a human point of view, illogical. The main purpose of this chapter is to investigate a means of identifying ungrammatical hypotheses from the MT output by using grammatical knowledge that expresses syntactic dependencies of words or word groups. We introduce several methods that try to establish this kind of linkage between the words of a hypothesis and, thus, determine its well-formedness, or "fluency". We perform rescoring experiments that rerank n-best lists according to the presented framework.

As methodologies deriving well-formedness of a sentence we use supertagging [Bangalore & Joshi 99] with lightweight dependency analysis (LDA)[1] [Bangalore 00], link grammars [Sleator & Temperley 93] and a maximum entropy (ME) based chunk parser [Bender & Macherey$^+$ 03]. The former two approaches explicitly model the syntactic dependencies between words. Each hypothesis that contains irregularities, such as broken linkages or non-satisfied dependencies, should be penalized or rejected accordingly. For the ME chunker, the idea is to train n-gram models on the chunk or POS sequences and directly use the log-probability as feature score. In general, these concepts and the underlying programs should be robust and fast in order to be able to cope with large amounts of data (as it is the case for n-best lists).

In [Och & Gildea$^+$ 04], the effects of integrating syntactic structure into a state-of-the-art SMT system were investigated. The approach is similar to the approach presented here: first, a word graph is generated using the baseline SMT system and n-best lists are extracted accordingly, then additional feature functions representing syntactic knowledge are added and the corresponding scaling factors are trained discriminatively on a development n-best list.

[1] In the context of this work, the term LDA is not to be confused with *linear discriminant analysis*.

[Och & Gildea$^+$ 04] investigated a large amount of different feature functions. The field of application varies from simple syntactic features, such as IBM model 1 score, over shallow parsing techniques to more complex methods using grammars and intricate parsing procedures. The results were rather disappointing. Only one of the simplest models, i.e. the implicit syntactic feature derived from IBM model 1 score, yielded consistent and significant improvements. All other methods had only a very small effect on the overall performance.

5.2 Reranking Framework

Using the techniques described in Section 4.5, we first run our SMT decoder to generate the graph of multiple translation candidates and to extract the n-best list accordingly. Within the reranking framework, we then apply additional models to rescore the translation hypotheses. Those additional models are typically hard to apply during the search, either because of the high computational demands or because they require that the translation hypothesis is fully generated. For instance, the IBM model 1 $p(f_1^J|e_1^I)$ involves a sum over the target positions, which is not applicable to partial hypotheses. Subsequently, we add the model scores to the n-best list and train the corresponding scaling factors discriminatively on a development n-best list. In the next subsections, we present standard models for rescoring of MT hypotheses followed by the description of the three models designed to derive syntactic dependencies on words or word groups, namely supertagging/LDA, link grammars and ME based chunking.

5.2.1 Standard Rescoring Models

In the second pass, we rerank the generated n-best translation candidates applying the following rescoring models:

- IBM model 1:
  This rescoring model measures the quality of the translations by using the IBM model 1 lexicon probabilities estimated during the word alignment training on a sentence level. Although very simple, this model yields good improvements according to [Och & Gildea$^+$ 03].

- deletion model:
  During IBM model 1 rescoring, we count all source words whose lexical probability given each target word is below a specified threshold, in the experiments the threshold was chosen between 10^{-1} and 10^{-4}.

- sentence length model:
  As described in [Zens & Ney 06], we explicitly model the target sentence length I by summing up the posterior probabilities of those target candidates that have length I.

- count language models:
  We apply on-the-fly language model estimation from n-gram counts using deleted interpolation. In the experiments, the Google n-gram counts, counts collected on the GigaWord corpus and the BBN web counts are used. We typically use 5-grams for this rescoring model.

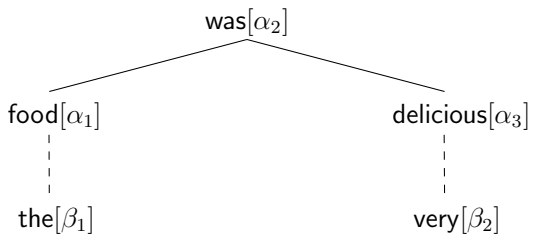

Figure 5.1. LDA: example of a derivation tree, β nodes are the result of the adjunction operation on auxiliary trees, α nodes of substitution on initial trees.

5.2.2 Supertagging With Lightweight Dependency Analysis (LDA)

Supertagging [Bangalore & Joshi 99] uses the Lexicalized Tree Adjoining Grammar formalism (LTAG) [XTAG Research Group 01]. Tree Adjoining Grammars incorporate a tree-rewriting formalism processing elementary trees that can be combined by two operations: more complex tree structures of the sentence considered are derived either by substitution or adjunction. Lexicalization allows us to associate each elementary tree with a lexical item called the *anchor*. In LTAGs, every elementary tree has such a lexical anchor (also called head word). It is possible that there is more than one elementary structure associated with a lexical item, as e.g. for the case of verbs with different sub-categorization frames.

The elementary structures are called initial and auxiliary trees. They hold all dependent elements within the same structure, thus imposing constraints on the lexical anchors in a local context. Basically, supertagging is very similar to part-of-speech (POS) tagging. Instead of basic POS tags, the elementary structures of LTAGs are annotated to the words of a sentence as richer descriptions. They are called *supertags* in order to distinguish them from ordinary POS tags. The result is an "almost parse" because of the dependencies coded within the supertags. Usually, a lexical item can have many supertags, depending on the various contexts it appears in. Therefore, the local ambiguity is larger than for the case of POS tags. An LTAG parser for this scenario can be very slow, i.e. its computational complexity is in $O(n^6)$ because of the large number of supertags, i.e. elementary trees, that have to be examined during a parse. In order to speed up the parsing process, we apply n-gram models on a supertag basis to filter out incompatible descriptions and thus improve the performance of the parser. The simple supertagging approach based on n-grams helps to reduce the possible number of supertags for each word of a sentence and hence facilitates the task of the parser. In [Bangalore & Joshi 99], a trigram supertagger with smoothing and back-off was reported that achieves an accuracy of 92.2% when trained on one million running words.

There is another benefit of the dependencies coded in the elementary structures. We can use them to actually derive a shallow parse of the sentence in linear time. The procedure was presented by [Bangalore 00] and is called *lightweight dependency analysis*. The concept is comparable to *chunking*. The lightweight dependency analyzer (LDA) finds the arguments for the encoded dependency requirements. There exist two types of *slots* that can be filled. On the one hand, nodes marked for substitution (in α-trees) have to be filled by the complements of the lexical anchor. On the other hand, the foot nodes (i.e. nodes marked for adjunction in β-trees) take words that are being modified by the supertag. Figure 5.1 shows a tree derived by LDA on the sentence *"the food was very delicious"*. The supertagging and LDA tools are

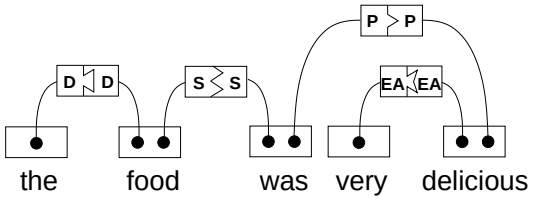

Figure 5.2. Link grammar: example of a valid linkage satisfying all constraints.

available from the XTAG research group website.[2]

As features considered for the reranking experiments we choose:

- supertagger output: directly use the log-likelihoods as feature score. This did not improve performance significantly, so the model was eventually discarded.
- LDA output:
  - dependency coverage: determine the number of covered elements, i.e. where the dependency slots are filled and use the number of non-filled slots as penalty
  - separate features for the number of modifiers and complements determined by the LDA

5.2.3 Link Grammars

Similar to the ideas presented in the previous subsection, link grammars also explicitly code dependencies between words [Sleator & Temperley 93]. These dependencies are called *links* which reflect the local requirements of each word. Several constraints have to be satisfied within the link grammar formalism to derive correct linkages, i.e. sets of links, of a sequence of words:

1. Planarity: links are not allowed to cross each other
2. Connectivity: links suffice to connect all words of a sentence
3. Satisfaction: linking requirements of each word are satisfied

An example of a valid linkage is shown in Figure 5.2. The link grammar parser that we use is freely available from the authors' website.[3] Again as for the LTAG or for supertagging, the link grammar formalism is lexicalized which allows for enhancing the methods with probabilistic n-gram models. In [Lafferty & Sleator$^+$ 92], the link grammar was used to derive a new class of language models that incorporate capabilities for expressing long-range dependencies between words. This is only rarely possible using traditional n-gram LMs. The link grammar dictionary that specifies the words and their corresponding valid links currently holds approximately 60 000 entries and handles a wide variety of phenomena in English. It is derived from newspaper texts.

[2]XTAG is an on-going project to develop a wide-coverage grammar for English using a lexicalized Tree Adjoining Grammar (TAG) formalism, `http://www.cis.upenn.edu/~xtag/`.

[3]The Link Grammar Parser is a syntactic parser of English, based on link grammar, an original theory of English syntax, `http://www.link.cs.cmu.edu/link/`.

[NP the food] [VP was] [ADJP very delicious]

the/DT food/NN was/VBD very/RB delicious/JJ

Figure 5.3. Chunking and POS tagging: example of a chunk parse, a tag next to the opening bracket denotes the type of chunk, whereas the corresponding POS tag is given after the word.

Within our reranking framework, we use link grammar features that express a possible well-formedness of the translation hypothesis. The simplest feature is a binary one stating whether the link grammar parser could derive a complete linkage or not. This should be a strong indicator of a syntactically correct sentence. Additionally, we added a normalized cost of the matching process which turned out not to be very helpful for rescoring, so it was discarded.

5.2.4 Maximum Entropy Based Chunking

Like the methods described in the two preceding subsections, text chunking consists of dividing a text into syntactically correlated non-overlapping groups of words. Figure 5.3 shows again our example sentence *"the food was very delicious"* illustrating this task. Chunks are represented as groups of words between square brackets. We employ the 11 chunk types as defined for the CoNLL-2000 shared task [Tjong Kim Sang & Buchholz 00].

For the experiments, we apply a maximum entropy (ME) based tagger which has been successfully evaluated on natural language understanding [Bender & Macherey$^+$ 03] and named entity recognition [Bender & Och$^+$ 03]. Within this tool, we directly factorize the posterior probability and determine the corresponding chunk tag for each word of an input sequence. We assume that the decisions depend only on a limited window $e_{i-2}^{i+2} = e_{i-2}...e_{i+2}$ around the current word e_i and on the two predecessor chunk tags c_{i-2}^{i-1}. In addition, part-of-speech (POS) tags g_1^I are assigned and incorporated into the model (cf. Figure 5.3). Thus, we obtain the following second-order model:

$$
\begin{aligned}
& Pr(c_1^I | e_1^I, g_1^I) = \\
& = \prod_{i=1}^{I} Pr(c_i | c_1^{i-1}, e_1^I, g_1^I) && (5.1) \\
& = \prod_{i=1}^{I} p(c_i | c_{i-2}^{i-1}, e_{i-2}^{i+2}, g_{i-2}^{i+2}), && (5.2)
\end{aligned}
$$

where the step from Equation 5.1 to Equation 5.2 reflects our model assumptions.

Furthermore, we implemented a set of binary valued feature functions for our system, including lexical, word and transition features, prior features, and compound features, cf. [Bender & Macherey$^+$ 03]. We run simple count-based feature reduction and train the model parameters using the Generalized Iterative Scaling (GIS) algorithm [Darroch & Ratcliff 72]. In practice, the training procedure tends to result in an overfitted model. To avoid this, a smoothing method is applied where a Gaussian prior on the parameters is assumed [Chen & Rosenfeld 99].

Within our reranking framework, we first use the ME based tagger to produce the POS and chunk sequences for the different n-best list hypotheses. Given several n-gram models trained on the Wall Street Journal (WSJ) corpus for both POS and chunk models, we then rescore

the n-best hypotheses and simply use the log-probabilities as additional features. In order to adapt our system to the characteristics of the MT training data used, we build POS and chunk n-gram models on the training corpus part and make use of them as for the WSJ models.

Note that the ME chunking approach does not model explicit syntactic linkages of words. Instead, it incorporates a statistical framework to exploit valid and syntactically coherent groups of words by additionally looking at the word classes.

5.3 Summary

Given the fact that statistically driven MT systems still make a lot of errors, we showed how to set up a reranking framework to enhance the MT quality by exploiting multiple translation candidates using additional rescoring models. Those additional models are typically hard to apply during the search, either because of the high computational demands or because they require that the translation hypothesis is fully generated. We started using standard models that have shown to be particularly useful for rescoring of MT hypotheses in the past and then added syntactically motivated features. The goal was to analyze whether shallow parsing techniques help in identifying ungrammatical hypotheses. The experimental findings in Chapter 9 show that the use of syntactically motivated feature functions within a reranking concept helps to slightly reduce the number of translation errors of the overall translation system. Although the improvement is only moderate, the results are nevertheless comparable or better to the ones from [Och & Gildea$^+$ 04].

Chapter 6

Robustness and Multi-Domains

As stated in the scientific goals, the aim of this work is to extend the state-of-the-art in phrase-based SMT in order to build a robust SMT system for multi-domain translation tasks. The SMT system presented in this work is the primary MT engine of the SRI Nightingale team for the Arabic-English tasks within the GALE project. Consequently, many of the contributions of this thesis were developed in the scope of the GALE project. In the previous sections, we showed how to address the morphological richness when preprocessing Arabic input to SMT, how to organize the search to efficiently generate translation candidates, and how to set up a reranking framework to further enhance the MT quality. In this section, we present our extensions to qualify the system for robustness and multi-domains. This involves the processing of noisy and unstructured input data, possibly automatically generated speech transcriptions. Thus, one has to cope with effects of spontaneous speech, misrecognitions from the automatic speech recognizer (ASR), and unknown words or constructs which are due to the variety of domains. We cover our work on the domain adaptation for the applied language models, the adjustment of the ASR and SMT vocabularies, and the prediction of punctuation marks that are missing from the ASR output. In [Bender & Matusov$^+$ 07], we applied these extensions to the GALE speech translation tasks.

6.1 Motivation

When going from (clean) newswire text translation to the translation of noisy and unstructured input data, e.g. data collected from newsgroups or weblogs, or automatic transcriptions of arbitrary broadcast conversations, one has to focus on a couple of problems. First of all, there will be a lot of out-of-vocabulary words (OOVs) due to the variety of domains but also a lot of unknown word constructs as the documents differ completely in their characteristics, e.g. transcriptions of broadcast news which should contain clean text except for recognition errors vs. completely unstructured weblogs. In general, if automatic speech recognition (ASR) is involved in the translation process, it has to be ensured that both the ASR system and the SMT system use matching vocabularies. Furthermore, ASR output lacks the existence of any type of punctuation marks or sentence segmentation. Case information is not present, numbers and abbreviations are written out as words, recognition errors occur, and one has to deal with effects of natural speech, like hesitations and filler words. In comparison to conventional MT tasks, two problems are striking: *robustness* with regard to the translation of automatically recognized speech and unstructured input, and the ability to produce good quality translations for *multi-domain* tasks.

The SMT system presented in this thesis has proven to attain these goals, as it has been successfully applied to the series of GALE translation evaluations. Within the GALE evaluations, the systems have to translate

- text input out of two different domains: newswire (NW) texts as was used for most previous MT evaluations and web texts (WT) derived from newsgroups and weblogs, as well as
- recorded speech out of two different domains: broadcast news (BN) and broadcast conversations (BC) which are focused more on discussions and call-ins that have a conversational style of speech.

In the evaluation setup, we perform the two-pass approach: in the first pass our statistical phrase-based decoder generates n-best translation candidates which are reranked applying additional models in the second pass. We apply domain adapted language models as this is the most straightforward way to address varying input domains and to tailor an SMT system to a specific domain. .

6.2 Domain Adapted Language Models

In this section, we describe the training of language models resulting in genre-specific domain adaptation for the overall SMT system. For the LM training, we combine different corpora representing various genres, e.g. broadcast news or conversations. The interpolation weights for these corpora are determined with the Downhill-Simplex algorithm which is a standard approach for training the parameters of the log-linear model combination. The optimization criterion is the perplexity of the interpolated LM on a development set.

6.2.1 Implementation

We use the SRI Language Modeling toolkit [Stolcke 02] and incorporate the Downhill-Simplex method from the Numerical Recipes [Press & Teukolsky$^+$ 02]. The genre-specific training corpora are separately loaded as dynamic language models where the interpolated probabilities of n-grams are calculated on-the-fly. This results in very efficient training of the interpolation weights, i.e. only the probabilities accessed during perplexity calculations are merged for the different LMs. Depending on the number of models, the training converges after 30–40 iterations.

In a final step, a static mixture of the LMs is created and written to disk. Thus, it is possible to train several tuned baseline models (e.g. additional ones using more data) and again interpolate them using this approach. Interestingly, when applied to the specific genres such as WT, BN or BC, the perplexity reductions on the development set carry over nicely to the test set which makes this method appealing.

6.2.2 Results

In Table 6.1, the perplexity reductions on the GALE test sets are shown for the different domains. We use the GALE 2007 MT development set (Dev) for optimization of the interpolation weights and then use the GALE 2006 MT evaluation set (Eval) as a blind test set. The baseline denotes perplexities obtained with a standard modified Kneser-Ney discounted language model without any genre-specific tuning (base) and trained on the whole target language corpora of the available bilingual data (GALE allowed corpora). After tuning the weights for each of the six main sub-parts of the LM and for each genre via Downhill-Simplex, we obtain significant

Table 6.1. Perplexities on the GALE test sets (GALE 2007 MT development set (DEV) and GALE 2006 MT evaluation set (EVAL)) for various settings: BASE – baseline 4-gram w/ KN discounting; DMIX-GS – genre-specific adaptation using DS; DMIX-GS* – genre-specific adaptation using additional data.

	4-gram w/ KN discounting			total
	BASE	DMIX-GS	DMIX-GS*	red. [%]
DEV-NW	93.8	93.4	83.2	-11.3
DEV-WT	168.6	141.6	124.4	-26.2
DEV-BN	81.9	76.4	72.6	-11.4
DEV-BC	99.1	80.6	79.4	-19.9
EVAL-NW	99.9	104.2	90.5	-9.4
EVAL-WT	212.8	169.9	144.2	-32.3
EVAL-BN	127.8	108.5	103.4	-19.1
EVAL-BC	116.3	95.9	90.5	-22.2

reductions in terms of perplexity (DMIX-GS). As there is additional monolingual data available (e.g. GigaWord v2, TDT, BBN data, ...), this procedure is repeated, resulting in column DMIX-GS*. We also include the GALE data releases for distillation (LDC2007E02) and web data (LDC2007E04 and LDC2007E44). DMIX-GS* is tuned using DMIX-GS plus five additional (genre-specific) LMs.

On the test set, overall reductions reach from 9–32%. The smallest effect can be seen on news data, which is reasonable due to most of the data coming from the newswire domain and already reflecting the domain sufficiently well. The biggest effect can be seen on broadcast conversations (22%) and web text (32%), the genres which are "most diverse" from traditional newswire text. The effect on the translation error measures can be seen in Chapter 9.

6.3 Adjustment of ASR and SMT Vocabularies

As in any data-driven approach, our SMT system requires the proper preprocessing of training and testing data (cf. Chapter 3). Otherwise, the system will encounter many words that have not been observed in training and that are thus missing from the phrase tables and word lexica. When translating automatically recognized speech, preprocessing becomes even more important as the ASR and SMT systems are usually trained on different training data using different preprocessing tools. To overcome these difficulties and to adjust the ASR and SMT vocabularies, we perform the following steps. Note that both systems process UTF-8 encoded data.

1. First, we apply some rule-based normalization of Arabic words as described in [Isbihani & Khadivi$^+$ 06], e.g. always mapping the hamza at the beginning of a word onto the same form, or removing the tanween character at the end of a word.

2. The next step is to split pre- and suffixes. For morphologically rich languages like Arabic, this step is important to reduce the size of the vocabulary and to obtain a computationally manageable system. In former experiments [Bender & Isbihani$^+$ 06], we used the MADA tool [Habash & Rambow 05] for morphological disambiguation and applied

the D2-scheme of [Habash & Sadat 06] for word segmentation. These tools require the input to be Buckwalter encoded. Buckwalter maps the Arabic (UTF-8) characters onto an ASCII alphabet and is thus error prone since there may still be English words and non-Arabic characters in the original input which can not be represented in the Buckwalter encoding. We therefore decided to extract all splittings of pre- and suffixes on a Buckwalter encoded and MADA preprocessed version of the training data, to recode the splittings in UTF-8, and to apply them as mappings.

3. In a third step, the spoken numbers are converted to digits and regular expressions are used to categorize numbers, URLs and e-mail addresses.

6.4 Punctuation Prediction

The translations of the ASR output are expected to have proper sentence boundaries and punctuation. However, this annotation can not be transferred from the automatic transcripts, since the raw ASR output is just a sequence of words for a given audio document. We perform the sentence segmentation using the algorithm of ICSI/UW [Matusov & Hillard$^+$ 07] which applies multiple acoustic and language model features to compute posterior probabilities of a segment boundary after each word. If the segmentation posterior probability is higher than a given threshold, a segment boundary is inserted. The threshold is optimized on a development set. Alternatively, we can use the dynamic programming algorithm of [Matusov & Mauser$^+$ 06] with the posterior probability as the main feature. This algorithm has the advantage that the minimum and maximum sentence lengths can be explicitly specified, and an explicit language-specific sentence length model can be used. A limit on the maximum sentence length – 60 words – is necessary to reduce the computational complexity of the MT search. We set the minimum sentence length to three words since one- and two-word segments are difficult to translate because of the missing context.

The ICSI/UW sentence segmentation system is able to predict the sentence type, i.e. if a sentence is a statement or a question. This information is used to generate the sentence-final punctuation – a period or a question mark. We insert these punctuation marks into the ASR output and translate them as usual.

In order to obtain sentence-internal punctuation in the English translations, we let the MT system predict the commas as described by [Matusov & Mauser$^+$ 06]. To this end, we train the word alignment as usual with punctuation marks present in the source and target part of the bilingual training corpus. Then, we remove all sentence-internal punctuation marks from the source part of the corpus only, adjusting the word alignment indices. Thus, many bilingual phrases extracted from the modified alignment contain target language commas (and other sentence-internal punctuation like semicolon) as insertions. During decoding, the decision on whether or not to insert a comma is made jointly by the translation model and the language model. The scaling factors of the MT models are re-optimized on the ASR output for the development set.

An alternative to this approach would have been predicting punctuation marks in the Arabic ASR output and translating them. This worked well for Spanish as the source language [Matusov & Mauser$^+$ 06]. However, the Arabic punctuation rules are quite different from the English ones. Moreover, in the available training data the punctuation is not consistent; there are many long sentences which have no sentence-internal punctuation at all. Therefore,

the prediction of punctuation marks in the target language with the joint power of the phrase-based translation models and the language model is more reliable. In [Al-Onaizan & Mangu 07], experiments on Arabic-English translation tasks also showed the advantage of this approach.

6.5 Summary

In this chapter, we presented the extensions to enable our decoder to produce translations of good quality even when processing noisy and unstructured input data, possibly automatically generated text transcriptions and a smorgasbord of different domains. In comparison to conventional newswire MT tasks, two problems are striking: robustness with regard to the translation of automatically recognized speech and unstructured input, and the ability to produce good quality translations for multi-domain tasks. Chapter 9 shows significant improvements compared to a conventional "newswire" setup of our decoder or compared to the first GALE 2006 system achieved by introducing new feature functions, applying domain adapted genre-specific language models, adding additional data and reranking the candidate translations. We also described our work on adjusting the ASR and SMT vocabularies in a preprocessing step to MT and on predicting punctuation marks that are missing from automatically transcribed speech in the translation process.

Chapter 7

Transliteration of Proper Names

Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Our work is motivated by the problem of out-of-vocabulary (OOV) terms which are a persistent problem in any SMT system. Often, such terms are the names of entities and the intention is to preserve the names by transliteration. This chapter studies the transliteration of Arabic proper names. We carry out experiments on two different corpora and investigate different statistical approaches to the transliteration task. These methods are purely data-driven and do not require any additional knowledge but a set of training name pairs. Finally, we analyze the benefit of the individual approaches when applied within a system combination framework. In Section 9.2, we report experimental results for the individual systems as well as for the light-weighted system combination obtained on two different corpora.

7.1 Motivation

Transliteration has been in use in machine translation systems, e.g. Russian-English, since the existence of the field of machine translation. However, to our knowledge it was first studied as a machine learning problem by [Knight & Graehl 97, Knight & Graehl 98] using probabilistic finite state transducers (FSTs). Subsequently, the performance of their system was greatly improved by combining different spelling and phonetic models [Al-Onaizan & Knight 02a]. In the literature, there exist two different approaches to the transliteration task: first, the actual transliteration of proper names as stated above, i.e. the mapping of the source language names onto new, plausible target language spellings, and second, the (phonetic) matching of the source language names against a large list of candidate transliterations. The concept of matching names against a list of transliteration candidates emerged from the attempts to better handle named entities in MT, e.g. [Lee & Chang 03]. In addition to a generative model, a large list of names in the target language and a similarity metric for the words in the two languages are crucial. In this way, [Huang & Vogel$^+$ 04] constructed a probabilistic Chinese-English edit model as part of a larger alignment solution using a heuristic bootstrapped procedure. [Freitag & Khadivi 07] proposed a technique which combines conventional MT methods with a single layer perceptron. Moreover, [Hermjakob & Knight$^+$ 08] focused on the question when to transliterate names and when to let the MT system generate the translations of proper names.

In this work, we focus on the first approach and compare different statistical methods for Arabic name transliteration. In contrast to the second approach and to many recent publications, these methods are purely data-driven. Section 7.2 describes the individual methods being investigated. Finally, we analyze the benefit of the individual methods when applied within

فةومپ	→	Fattumah
تلمت	→	Talammit
گسَانَى	→	Ghassani
مەمد الموسَى	→	Muhammad Al Musa
سبخپ ەسيَان المنبةه	→	Sabkhat Hasyan Al Munbatih

Figure 7.1. Examples of Arabic names with their corresponding English transcriptions.

a system combination framework. We proceed as is customary in speech recognition, i.e. we follow the Recognizer Output Voting Error Reduction (ROVER) approach [Fiscus 97].

[Freitag & Khadivi 07] analyzed that it is generally impossible to achieve perfect performance for the transliteration of Arabic names since in Arabic, many sounds, such as short vowels, diphthong markers, and doubled consonants, are usually not written. They calculated that approximately 25% of the characters in the English transcriptions must be inferred, thus posing a baseline character error rate of 25% to be achieved through basic transliteration approaches. Some Arabic names with their corresponding English transcriptions are shown in Figure 7.1.

7.2 Data-Driven Approaches to Arabic Name Transliteration

7.2.1 Phrase-Based Machine Transliteration

The transliteration system of [Al-Onaizan & Knight 02a] can be regarded as an SMT system which translates source language characters to target language characters. Also, [Freitag & Khadivi 07] used an SMT system for comparison with state-of-the-art and provided evidence that the combination of their sequence alignment model with the SMT model promised further improvement. Accordingly, the idea was to use our state-of-the-art phrase-based SMT system [Zens & Och$^+$ 02, Zens & Ney 04] for Arabic name transliteration, i.e. the translation of characters instead of words. Formally, we are given a sequence of source language characters $s_1^M = s_1, ..., s_m, ..., s_M$ representing an Arabic name which is to be translated into a sequence of target language characters $t_1^N = t_1, ..., t_n, ..., t_N$. As customary for SMT, we apply the so-called maximum-a-posteriori (MAP) decision rule and choose the (English) character sequence with the highest probability among all possible target language character sequences:

$$\hat{t}_1^{\hat{N}} = \underset{N, t_1^N}{\operatorname{argmax}} \left\{ Pr(t_1^N | s_1^M) \right\} . \tag{7.1}$$

As for the conventional text translation task, we deploy a log-linear model:

$$Pr(t_1^N | s_1^M) = \frac{\exp\left(\sum_{k=1}^K \lambda_k h_k(t_1^N, s_1^M)\right)}{\sum\limits_{N', t'^{N'}_1} \exp\left(\sum_{k=1}^K \lambda_k h_k(t'^{N'}_1, s_1^M)\right)} . \tag{7.2}$$

In general, we proceed in almost the same manner as for the translation of texts:

- First of all, we extract the character phrases from the generalized character alignment.

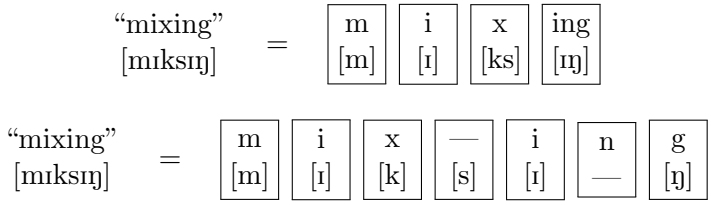

Figure 7.2. G2P co-segmentations for the pronunciation of the example word "mixing": a sequence of four general graphones vs. a segmentation into seven singular graphones (from [Bisani & Ney 08]).

- The character-level phrase translation and lexicon models are used for both directions.
- We use a 3-gram character sequence model, a character penalty and a phrase penalty.
- However, we do not consider any reordering model due to the monotonicity of the transliteration task,
- The model scaling factors λ_1^K are tuned for maximum transliteration accuracy.

7.2.2 Transliteration as Grapheme-To-Phoneme Conversion

Grapheme-to-phoneme conversion (G2P), or phonetic transcription, is the task of finding the pronunciation of a word given its written form. In text-to-speech and speech recognition systems, for instance, the G2P component is crucial for predicting pronunciations that are missing from the lexicon. Here, we resort to the Sequitur G2P toolkit[1] of [Bisani & Ney 02, Bisani & Ney 03] which implements joint-sequence models. Formally, an orthographic form is given as a sequence of letters, also referred to as characters or graphemes. We denote the set of graphemes as G. A pronunciation is represented in terms of a phonemic transcription, i.e. a sequence of phoneme symbols. The set of phonemes is denoted as Φ. Then, the task of G2P is defined as:

$$\boldsymbol{\varphi}(\boldsymbol{g}) = \underset{\boldsymbol{\varphi}' \in \Phi^*}{\operatorname{argmax}} \left\{ p(\boldsymbol{\varphi}', \boldsymbol{g}) \right\} , \tag{7.3}$$

i.e. for a given orthographic form (sequence of letters) $\boldsymbol{g} \in G^*$ we seek the most likely pronunciation (phoneme sequence) $\boldsymbol{\varphi} \in \Phi^*$.

The fundamental idea of joint-sequence models is that the relation of input and output sequences can be generated from a common sequence of joint units which carry both input and output symbols [Bisani & Ney 08]. We follow their naming convention and refer to joints units as *graphones*. A grapheme-phoneme joint multi-gram, or *graphone* for short, is a pair $q = (\boldsymbol{g}, \boldsymbol{\varphi}) \in Q \subseteq G^* \times \Phi^*$ of a letter sequence and a phoneme sequence of possibly different length. In the joint multi-gram model it is assumed that for each word its orthographic form and its pronunciation are generated by a common sequence of graphones. Each graphone can carry multiple input and output symbols by definition, but in the simplest case, each graphone

[1] The Sequitur G2P software was developed at RWTH Aachen University - Department of Computer Science by Maximilian Bisani, and is freely available: `http://www-i6.informatik.rwth-aachen.de/web/Software/index.html`.

carries zero or one letter and zero or one phoneme. We call a such graphone *singular*. In the next step, the letter and the phoneme sequences are grouped into an equal number of segments. Such a grouping is called a joint segmentation, *co-segmentation*, or more general alignment. Obviously, the segmentation into graphones is not unique. Figure 7.2 shows the pronunciation of the example word "mixing" as a sequence of four general graphones vs. a segmentation into just singular graphones. The latter corresponds to the conventional definition of finite state transducers (FST). While the general segmentation provides additional freedom in how to group the input and output symbols, the simple FST-style or "01-to-01" segmentation proved to achieve the best performance for the transliteration task.

[Bisani & Ney 08] resolved the segmentation ambiguity by summing over all possible graphone sequences:

$$p(\boldsymbol{g}, \boldsymbol{\varphi}) = \sum_{\boldsymbol{q} \in S(\boldsymbol{g}, \boldsymbol{\varphi})} p(\boldsymbol{q}) \,. \tag{7.4}$$

$\boldsymbol{q} \in Q^*$ hereby denotes the sequence of graphones and $S(\boldsymbol{g}, \boldsymbol{\varphi})$ is the set of all co-segmentations of $\boldsymbol{g}$ and $\boldsymbol{\varphi}$:

$$S(\boldsymbol{g}, \boldsymbol{\varphi}) = \left\{ \boldsymbol{q} \in Q^* \;\middle|\; \begin{array}{lcl} \boldsymbol{g}_{q_1} \smile \cdots \smile \boldsymbol{g}_{q_K} & = & \boldsymbol{g} \\ \boldsymbol{\varphi}_{q_1} \smile \cdots \smile \boldsymbol{\varphi}_{q_K} & = & \boldsymbol{\varphi} \end{array} \right\} . \tag{7.5}$$

Here, $\smile$ is the sequence concatenation operation and $K = |\boldsymbol{q}|$ identifies the length of the graphone sequence $\boldsymbol{q}$. In this way, the joint probability distribution $p(\boldsymbol{g}, \boldsymbol{\varphi})$ is reformulated as a probability distribution $p(\boldsymbol{q})$ over graphone sequences $\boldsymbol{q} = q_1, \ldots, q_K$, and can now be modeled using a standard M-gram approximation:

$$p(q_1^K) = \prod_{j=1}^{K+1} p(q_j | q_{j-1}, \ldots, q_{j-M+1}) \,. \tag{7.6}$$

A detailed description of the model estimation is beyond the scope of this thesis, instead the reader is referred to [Bisani & Ney 08]. Briefly sketched, the model training is done in two steps:

- First, maximum likelihood and expectation maximization (EM) training is used to infer the graphones.
- Second, the input is segmented into a stream of graphones and absolute discounting with leaving-one-out is applied to estimate the actual M-gram model.

Interpreting the characters of the English target names as phonemes, we are able to use the Sequitur G2P toolkit to transliterate the Arabic names. Under this assumption, the joint multi-grams correspond to an N-gram language model on Arabic letter rendering tuples, e.g. for the name pair (فةومپ, Fattumah):

$$p \left(\begin{array}{c|cc} \text{و} & \text{ف} & \text{ة} \\ \downarrow & \downarrow & \downarrow \\ u & fat & t \end{array} \right) .$$

7.2.3 Maximum Entropy Models for Transliteration

In [Bender 02], we developed a maximum entropy (ME) based tagger which is suitable for any kind of natural language processing (NLP) tagging task. The tool was successfully applied for named entity recognition (NER) [Bender & Och$^+$ 03], natural language understanding (NLU) [Bender & Macherey$^+$ 03], and for part-of-speech (POS) tagging/shallow parsing [Bender 02]. In the scope of this thesis, the POS and chunk models have already been used within the reranking framework as presented in Chapter 5.

All of these applications represent a conventional (NLP) tagging task, i.e. for each of the applications, we assign the appropriate tag/label for *each* word. Consequently, the tasks can be considered a one-to-one mapping and an alignment becomes redundant. In contrast, the generation of the alignment resp. segmentation is compulsory for the transliteration task. In order to still be able to use the ME tagger for transliteration, we utilize the output of the G2P toolkit. We take the segmentation as determined by the joint multi-grams, e.g. using the same name pair from above (فةومپ, Fattumah):

ف	ة	و	م	پ
↓	↓	↓	↓	↓
fat	t	u	ma	h

and insert "null words" ϵ, such that the transliteration task can be interpreted as a one-to-one mapping:

ف	ϵ	ϵ	ة	و	م	ϵ	پ
↓	↓	↓	↓	↓	↓	↓	↓
f	a	t	t	u	m	a	h

In the experiments, we also investigated the use of named ϵ, i.e. "null words" which are dependent on their context, but we were not able to exceed the baseline performance applying just a single global ϵ.

Using this one-to-one interpretation of the transliteration task, we can now proceed as described in, e.g. [Bender & Och$^+$ 03], i.e. we directly factorize the posterior probability. We assume that the decisions only depend on a limited window of $s_{n-d}^{n+d} = s_{n-d}, \ldots, s_{n+d}$ around the current source character s_n and on the immediate predecessor target character. Thus, we obtain the following first-order model:

$$Pr(t_1^N|s_1^N) = \prod_{n=1}^{N} Pr(t_n|t_1^{n-1}, s_1^N) \tag{7.7}$$

$$\cong \prod_{n=1}^{N} p_{\lambda_1^M}(t_n|t_{n-1}, s_{n-d}^{n+d}) \,. \tag{7.8}$$

Although the ME tagger supports higher order models, in practice it turned out that there is no benefit in going beyond first-order dependencies. Furthermore, a window size of $d = 4$ source characters turned out to be the optimal choice between transliteration performance and model size. A well-founded framework for directly modeling the posterior probability $p(t_n|t_{n-1}, s_{n-4}^{n+4})$ is maximum entropy [Berger & Della Pietra$^+$ 96]. In this framework, we have a set of M feature

functions $h_m(t_n|t_{n-1}, s_{n-4}^{n+4}), m = 1, \ldots, M$. For each feature function h_m, there exists a model parameter λ_m. Using the following notation:

$$H(t_{n-1}, t_n, s_{n-4}^{n+4}) = \exp\left(\sum_{m=1}^{M} \lambda_m \cdot h_m(t_{n-1}, t_n, s_{n-4}^{n+4})\right), \tag{7.9}$$

we obtain the ME model for transliteration:

$$p_{me}(t_1^N|s_1^N) = \frac{\prod_{n=1}^{N} H(t_{n-1}, t_n, s_{n-4}^{n+4})}{\prod_{n=1}^{N} \sum_{\tilde{t}} H(t_{n-1}, \tilde{t}, s_{n-4}^{n+4})}, \tag{7.10}$$

We deploy different types of binary valued feature functions:

- Lexical features:
  The characters s_{n-4}^{n+4} are compared to a vocabulary and possibly mapped onto an 'unknown character'. Formally, the feature

  $$h_{s,d,t}(t_{n-1}, t_n, s_{n-4}^{n+4}) = \delta(s_{n+d}, s) \cdot \delta(t_n, t), \quad d \in \{-4, ..., 4\}, \tag{7.11}$$

  will fire if the source character s_{n+d} matches the vocabulary entry s and if the prediction for the current source character equals t. $\delta(\cdot, \cdot)$ denotes the Kronecker-function.

- Transition features:
  Transition features model the first-order dependencies:

  $$h_{t',d,t}(t_{n-1}, t_n, s_{n-4}^{n+4}) = \delta(t_{n-1}, t') \cdot \delta(t_n, t). \tag{7.12}$$

- Prior features:
  The single target character priors are incorporated by prior features. They just fire for the currently observed character:

  $$h_t(t_{n-1}, t_n, s_{n-4}^{n+4}) = \delta(t_n, t). \tag{7.13}$$

- Compound features:
  Using the feature functions defined so far, we can only specify features that refer to a single character. To enable also character phrases and arbitrary source/target character combinations, we introduce the following compound features:

  $$\begin{aligned} h_{\{z_1,d_1\},...,\{z_K,d_K\},t}(t_{n-1}, t_n, s_{n-4}^{n+4}) &= \prod_{k=1}^{K} h_{z_k,d_k,t}(t_{n-1}, t_n, s_{n-4}^{n+4}), \\ & \quad z_k \in \{s, t'\}, \quad d_k \in \{-4, ..., 4\}. \end{aligned} \tag{7.14}$$

In theory, the principle of maximum entropy does not directly concern itself with the issue of feature selection [Berger & Della Pietra$^+$ 96]. In practice, however, feature selection is crucial to the performance of ME-based approaches. Moreover, it is important to reduce the number of active features to speed up the training process for complex tasks. In our system, we use simple

count-based feature reduction. Given a threshold K, we only include those features that have been observed on the training data at least K times. Although this method does not guarantee to obtain a minimal set of features, it performed well in practice. Experiments were carried out with different thresholds, and it turned out that for the transliteration task, a threshold of 5 or 6 is the best choice for all features.

For training, we use the maximum class posterior probability criterion:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}} \left\{ \sum_{n=1}^{N} \log p_{\lambda_1^M}(t_n | t_{n-1}, s_{n-4}^{n+4}) \right\}. \tag{7.15}$$

This corresponds to maximizing the likelihood of the ME model. Since the optimization criterion is convex, there is only a single optimum and no convergence problems occur. To train the model parameters λ_1^M we use the Generalized Iterative Scaling (GIS) algorithm [Darroch & Ratcliff 72]. In practice, the training procedure tends to result in an over-fitted model. To avoid over-fitting, [Chen & Rosenfeld 99] suggested a smoothing method where a Gaussian prior on the parameters is assumed. Instead of maximizing the probability of the training data, we now maximize the probability of the training data times the prior probability of the model parameters:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}} \left\{ p(\lambda_1^M) \cdot \sum_{n=1}^{N} p_{\lambda_1^M}(t_n | t_{n-1}, s_{n-4}^{n+4}) \right\}, \tag{7.16}$$

where

$$p(\lambda_1^M) = \prod_m \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\lambda_m^2}{2\sigma^2} \right]. \tag{7.17}$$

This method tries to avoid very large values for λ_m and avoids that features that occur only once for a specific class get value infinity. Note that there is only one parameter σ for all model parameters λ_1^M.

During transliteration, the search is performed using the so-called maximum approximation, i.e. the most likely sequence of target characters $\hat{t}_1^N$ is chosen among all possible sequences t_1^N:

$$\hat{t}_1^N = \underset{t_1^N}{\operatorname{argmax}} \left\{ Pr(t_1^N | s_1^N) \right\} \tag{7.18}$$

$$= \underset{t_1^N}{\operatorname{argmax}} \left\{ \prod_{n=1}^{N} p_{\lambda_1^M}(t_n | t_{n-1}, s_{n-4}^{n+4}) \right\}. \tag{7.19}$$

Therefore, the time-consuming re-normalization in Equation 7.10 is not needed during search. We run a Viterbi search to find the highest probability sequence [Borthwick & Sterling$^+$ 98].

7.2.4 Conditional Random Fields for Transliteration

Conditional random fields (CRFs) are a framework for building probabilistic models to segment and label sequence data [Lafferty & McCallum$^+$ 01]. The underlying idea is to define a conditional probability distribution over label sequences given a particular observation sequence, rather than defining a joint distribution over both label and observation sequences. Due to

their conditional nature, CRFs offer the ability to relax the strong independence assumptions required by HMMs. Furthermore, they also avoid the label bias problem of ME models. In the literature, CRFs have been reported to outperform both ME models and HMMs on a variety of tasks in different fields, such as natural language processing, speech recognition and bioinformatics.

From the modeling point of view, ME models and CRFs are very similar but only differ in the normalization term. While ME models are normalized on a position level, CRFs are normalized on a sentence or more general sequence level. Using the same notation (7.9) as for the ME models, we obtain:

$$p_{crf}(t_1^N|s_1^N) = \frac{\prod_{n=1}^{N} H(t_{n-1}, t_n, s_{n-4}^{n+4})}{\sum_{\tilde{t}_1^N} \prod_{n=1}^{N} H(\tilde{t}_{n-1}, \tilde{t}, s_{n-4}^{n+4})} . \tag{7.20}$$

Apart from the normalization, the CRF approach is identical to the ME method, i.e. we base on the same (G2P) segmentation, use the same feature functions and apply the same training and decision criteria. Hence, the model parameters are estimated to maximize the class posterior probability and a Viterbi search is run to determine the most likely sequence of target characters $\hat{t}_1^N$. For our experiments, we use the open-source CRF++ toolkit[2] which is also used for example by [Kudo & Yamamoto$^+$ 04] for Japanese morphological analysis.

7.2.5 A Deep Learning Approach to Transliteration

Finally, we investigate another transliteration technique which is based on deep belief networks (DBNs). DBNs were shown to work well for other machine learning problems. Our work was also motivated by the fact that DBNs have certain properties which are very interesting for transliteration and that the diverse nature of the DBN transliterations makes them compelling to be combined with conventional techniques as presented in the previous subsections. We published our initial experiments in [Deselaers & Hasan$^+$ 09].

Deep architectures in machine learning and artificial intelligence are becoming more and more popular after an efficient training algorithm has been proposed [Hinton & Osindero$^+$ 06], although the idea is known for some years [Ackley & Hinton$^+$ 85]. Deep belief networks consist of multiple layers of restricted Boltzmann machines (RBMs) and are built from RBMs by first training an RBM on the input data. A second RBM is built on the output of the first one and so on until a sufficiently deep architecture is created. RBMs are stochastic generative artificial neural networks with restricted connectivity. From a theoretical viewpoint, RBMs are interesting because they are able to discover complex regularities and find notable features in data [Ackley & Hinton$^+$ 85].

Here, we learn encoders for the source and target names respectively and then connect these two through a joint layer to map between the two languages. This joint layer is trained in the same way as the top-level neurons in the deep belief classifier from [Hinton & Osindero$^+$ 06]. In Figure 7.3, a schematic view of our DBN for transliteration is shown. On the left and on the right are encoders for the source and target names respectively. To transliterate a source name, it is passed through the layers of the network. First, it traverses through the source encoder on

[2] CRF++ is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data, and is freely available: `http://crfpp.sourceforge.net/`.

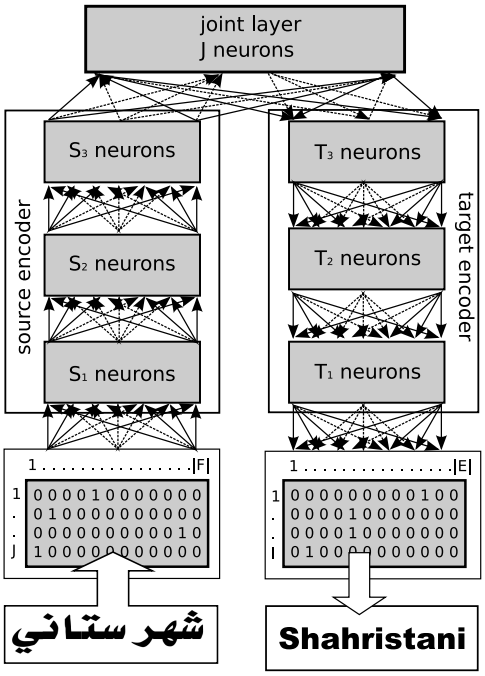

Figure 7.3. A schematic representation of our DBN for transliteration.

the left, then it passes into the joint layer, finally traversing down through the target encoder. Each layer consists of a set of neurons receiving the output of the preceding layer as input. The first layers in the source and target encoders consist of S_1 and T_1 neurons, respectively; the second layers have S_2 and T_2 nodes, and the third layers have S_3 and T_3 nodes, respectively. A joint layer with J nodes connects the source and the target encoders.

In this DBN, the number of nodes in the individual layers are the most important parameters. The more nodes a layer has, the more information can be conveyed through it, but the harder the training: the amount of data needed for training and thus the computation time required is exponential in the size of the network [Ackley & Hinton$^+$ 85].

To transliterate a source name it is first encoded as a D_F-dimensional binary vector S_F and then fed into the first layer of the source encoder. The S_1-dimensional output vector O_{S1} of the first layer is computed as

$$O_{S1} \leftarrow 1/\exp\left(1 + w_{S1}S_F + b_{S1}\right), \tag{7.21}$$

where w_{S1} is a $S_1 \times D_F$-dimensional weight matrix and b_{S1} is an S_1-dimensional bias vector. S_F is the D_F-dimensional representation of the input, O_{S1} is a S_1 dimensional vector. The outcome of $w_{S1}S_F$ in Equation 7.21 is an S_1-dimensional vector and the other operations are element-wise operations on vectors.

The output of each layer is used as input to the next layer as follows:

$$O_{S2} \leftarrow 1/\exp\left(1 + w_{S2}O_{S1} + b_{S2}\right), \tag{7.22}$$

$$O_{S3} \leftarrow 1/\exp\left(1 + w_{S3}O_{S2} + b_{S3}\right). \tag{7.23}$$

After the source encoder has been traversed, the joint layer is reached which processes the data twice: once using the input from the source encoder to get a state of the hidden neurons O_{SJ}

and then to infer an output state O_{JT} as input to the topmost level of the output encoder

$$O_{SJ} \leftarrow 1/\exp\left(1 + w_{SJ}O_{S3} + b_{SJ}\right), \tag{7.24}$$
$$O_{JT} \leftarrow 1/\exp\left(1 + w_{JT}O_{SJ} + b_{JT}\right). \tag{7.25}$$

This output vector is decoded by traversing downwards through the output encoder:

$$O_{T3} \leftarrow 1/\exp\left(1 + w_{T3}O_{JT} + b_{T3}\right), \tag{7.26}$$
$$O_{T2} \leftarrow 1/\exp\left(1 + w_{T2}O_{T3} + b_{T2}\right), \tag{7.27}$$
$$O_{T1} \leftarrow w_{T1}O_{T2} + b_{T1}, \tag{7.28}$$

where O_{T1} is a vector encoding a name in the target language. Note that this model is intrinsically bidirectional since the individual RBMs are bidirectional models and thus it is possible to transliterate from source to target and vice versa.

A problem with DBNs and transliteration is the data representation. The input and output data are sequences of commonly varying length but a DBN expects input data of constant length. To represent a source or target language name, it is converted into a sparse binary vector of dimensionality $D_F = |F| \cdot J$ or $D_E = |E| \cdot I$, respectively, where $|F|$ and $|E|$ are the sizes of the alphabets and I and J are the lengths of the longest names. If a name is shorter than this, a *padding letter* w_0 is used to fill the spaces. This encoding is depicted in the bottom part of Figure 7.3. Since the output vector of the DBN is not binary, we infer the maximum a-posterior hypothesis by selecting the letter with the highest output value for each position.

For the training, we follow the method proposed in [Hinton & Osindero$^+$ 06]. Each of the RBMs is trained individually in order to find a good starting point for back-propagation on the whole network. We use the average squared error over the output vectors between reference and inferred names as training criterion. Thus, the whole training procedure consists of 4 phases. First, an auto-encoder for the source names is learnt. Second, an auto-encoder for the target names is learnt. Third, these auto-encoders are connected by a top connecting layer, and finally back-propagation is performed over the whole network for fine-tuning of the weights. For a more detailed description of the training process, the reader is referred to [Deselaers & Hasan$^+$ 09].

7.3 Application Within a System Combination Framework

Motivated by the differences in transliteration performance and although, e.g. the DBN approach being clearly outperformed by the other systems, we perform light-weighted system combination using Recognizer Output Voting Error Reduction (ROVER), which is known to work well in speech recognition [Fiscus 97]. We investigate the potential benefit of the diverse nature of the individual systems' transliterations when fed into a system combination framework. Since, we currently only consider the single-best output of each system, ROVER is just a simple majority voting on character level after a Levenshtein alignment of all system outputs has been performed. Figure 7.4 shows the lattice representation for the ROVER combination (full lattice vs. combined edges) for the Arabic example name "تير". Although the English reference transliteration is "*Tir*", the investigated systems agree in "*Tayr*" being the most likely transliteration hypothesis.

We analyze the contribution of the individual systems and examine different combination setups taking into account the potential "similarity" of the approaches. To get an idea of the theoretically achievable improvement, we calculate the oracle error rates. This means

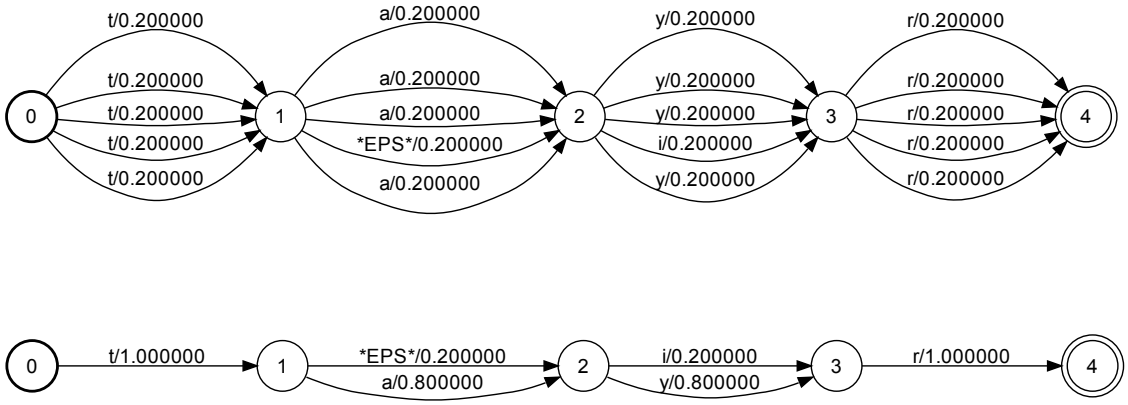

Figure 7.4. Lattice representation for the ROVER combination (full lattice vs. combined edges) for the Arabic example name "تير". English reference transliteration is "*Tir*" In contrast, the investigated systems agree in "*Tayr*" being the most likely transliteration.

that, for each slot in the ROVER lattice, we choose the character that matches the reference transliteration without taking into account which system generated this character. Obviously, this results in a far too optimistic upper bound.

We also try to optimize the system weights using Powell's method which is a prototype of conjugate direction methods. The idea for these methods is based on the fact that the optimum of a function that is separable can be found by optimizing separately each component. Clearly, Powell's method can be used to minimize the number of falsely chosen characters and thus to compute the optimal system weights w.r.t. the transliteration accuracy. However, the experimental results in Section 9.2 show that the improvements over equal systems weights are negligible.

7.4 Summary

In this chapter, we studied the transliteration of Arabic proper names. Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Our work was motivated by the problem of out-of-vocabulary (OOV) terms which are a persistent problem in any SMT system. Often, such terms are the names of entities and the intention was to preserve the names by transliteration.

In contrast to the (phonetic) matching of source language names against a large list of candidate transliterations and to many recent publications, we focused on purely data-driven approaches that do not require any additional knowledge but just a set of training name pairs. We showed how the following approaches, that have been successfully applied to NLP or general machine learning tasks, can be reformulated in order to be qualified for transliteration:

- (monotone) phrase-based MT on character level,

- grapheme-to-phoneme conversion,
- maximum entropy models,
- conditional random fields, and
- deep belief networks.

Finally, we analyzed the benefit of the individual methods when applied within a system combination framework. We proceeded as is customary in speech recognition, i.e. we followed the Recognizer Output Voting Error Reduction (ROVER) approach [Fiscus 97].

In Chapter 9, we carry out experiments on two different corpora and report results for the individual systems as well as for the light-weighted system combination. We compare the performance of the different approaches, analyze their contribution within the ROVER framework and examine different combination setups taking into account the potential "similarity" of the approaches.

Chapter 8

Search Strategies for Interactive Machine Translation

This chapter deals with search[1] strategies for interactive (statistical) machine translation systems. We first describe the concept of interactive machine translation from a statistical point of view. Next, we discuss the two search strategies investigated in this work and show how they can be successfully combined. In chapter 9, experimental results comparing the search strategies are presented and conclusions are drawn in Chapter 11. This work was published in [Bender & Hasan$^+$ 05].

8.1 Motivation

Although great progress has been made in the field of automatic translation in the last years, the produced translations are far from being perfect. Apart for very limited domains, no current state-of-the-art system can be directly used for real life applications. The produced sentences often contain grammatical errors and even the preservation of the meaning is not always achieved. Therefore a manual post-processing of the texts has to be done.

The concept of *interactive machine translation* already has a long history, and the first systems appeared in the end of the 1960's. However in most of these systems the user doesn't have a direct control over the translation process, and most of the user interaction is reduced to performing source language disambiguation on demand. The approach we center on in this work was first suggested by [Foster & Isabelle$^+$ 96] and an implementation was carried out in the TransType project [Langlais & Foster$^+$ 00b]. In such an environment, human translators interact with a translation system that acts as an assistance tool and dynamically provides a list of translations that best complete the part of the source sentence already translated. Further refinements were presented in the TransType2 project[2] [SchlumbergerSema S.A. & Intituto Tecnológico de Informática$^+$ 01].

Clearly, the best approach would be to start a new search for every given prefix. However, in these kind of systems, response time is a crucial factor for a human translator as delays higher than a fraction of a second are not acceptable. With nowadays' search algorithms and available computing power, these time restrictions can not be met when doing a whole new search for each prefix, so the performance achieved with this strategy will be an upper bound of the performance we get in the real system. We present an efficient generation strategy and compare its capability with this upper bound.

[1] We use the terms search and generation interchangeably within this chapter.

[2] TransType2 - Computer Assisted Translation, is an RTD project funded by the European Commission under the IST Programme (IST-2001-32091), `http://tt2.atosorigin.es/`.

8.2 Interactive Machine Translation

In this section, we briefly review the statistical framework for translation our system is based on. Recapitulate that in SMT we are given a source sentence f_1^J which is to be translated into a target sentence e_1^I which maximizes the posterior probability

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} . \tag{8.1}$$

Applying Bayes' Rule, we can modify this equation in order to introduce additional knowledge sources

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} . \tag{8.2}$$

The target language model $Pr(e_1^I)$ describes the well-formedness of the target language sentence. The translation model $Pr(f_1^J | e_1^I)$ links the source language sentence to the target language sentence. The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. Here, we maximize over all possible target language sentences.

In interactive machine translation, we have to find an extension e_{i+1}^I for a given prefix e_1^i. Hence, we constrain the search to those sentences e_1^I which contain e_1^i as prefix:

$$\hat{e}_{i+1}^I = \underset{I, \tilde{e}_{i+1}^I}{\operatorname{argmax}} \left\{ Pr(\tilde{e}_{i+1}^I | e_1^i) \cdot Pr(f_1^J | e_1^i, \tilde{e}_{i+1}^I) \right\} . \tag{8.3}$$

Thus, we maximize over all possible extensions $\tilde{e}_{i+1}^I$. For simplicity, this equation is formulated on the word level. We do not include the case where the prefix contains the first characters of the word e_{i+1}. In that case, we have to optimize over all target language words e_{i+1} that have the same word prefix. In the actual implementation, the method is applied on the character level, and the search for an extension is performed after each keystroke of the human translator.

The crucial factor is an efficient maximization of Equation 8.3 because human translators will only accept response times of fractions of a second. Using state-of-the-art search algorithms this is not achievable without putting up with an unacceptable amount of search errors. To overcome this problem, we can compute a word graph which represents a subset of possible extensions [Ney & Aubert 94, Ueffing & Och$^+$ 02]. The generation is then constrained to this set of extensions.

8.3 Phrase-Based Approach

The base method we use in our translation system is the alignment template approach as described in [Och & Tillmann$^+$ 99, Och & Ney 04]. This approach uses the so-called *alignment templates* which are pairs of source and target language phrases[3] together with the word alignment within the phrases. The alignment templates system was our predecessor SMT implementation which was further developed and became the PBT system which is used for generating translations throughout the other parts of this thesis. The alignment templates are introduced as hidden variables z_1^K when modelling the conditional translation probability $\Pr(f_1^J | e_1^I)$:

$$\Pr(f_1^J | e_1^I) = \sum_{z_1^K, a_1^K} \Pr(a_1^K | e_1^I) \cdot \Pr(z_1^K | a_1^K, e_1^I) \cdot \Pr(f_1^J | z_1^K, a_1^K, e_1^I) . \tag{8.4}$$

[3] In this context, phrases are simply sequences of words. No other linguistic meaning is required.

In Equation 8.4, we introduce the additional hidden variables a_1^K that model the alignment of the alignment templates themselves. As smoothing, automatically trained word classes can be used, and additional costs can easily be introduced by using a log-linear model. More details of this approach can be found in the literature.

8.3.1 Generation

The generation of the best translation for a given source sentence f_1^J is carried out by producing the target sentence in a sequential order. At each step of the generation algorithm, we maintain a set of active hypotheses and choose one of them for extension. A word of the target language is then added to the chosen hypothesis and its costs get updated. This kind of generation fits nicely into a dynamic programming framework, as hypotheses which are indistinguishable by both language and translation models (and that have covered the same source positions) can be recombined. The search space is however too big, and therefore pruning has to be done which leads us to a beam search algorithm. So, the actual search is very similar to our approach presented in Chapter 4.

8.4 Interactive Generation

In order to find the completion for a given prefix, the set of generated hypotheses could be restricted to only those which exactly match the given prefix. However, as our probabilistic models are far from being perfect, this approach is too restrictive. Instead, we penalize the hypotheses by introducing an additional cost in the log-linear model for each word that does not match the prefix. If hypotheses that can generate the given prefix are present in the active set, those do not get any additional costs. In the pruning process, the incompatible hypotheses will be discarded while the correct ones will remain in the set. Of course, the last "word" in the given prefix should be considered in a different way, as it itself can be a prefix of the next word. To ensure the extensions start with this word prefix, the comparison must be done at the character level. One might think about different costs for the mismatch of words within the prefix and for extensions which do not start with the given word prefix. If a word within the prefix can not be produced by the search algorithm, it will obviously not be produced by any further search call. This kind of substitution error is less harmful for producing good hypotheses than unfitting extensions, and should therefore be penalized less.

Using this approach, we can expect to obtain optimal results, as a new search is performed at each stage, and the information provided by the prefix is used to avoid search errors made in previous stages. However, the search process has a high computational cost and in the interactive systems the response time is a critical factor. Therefore, this approach can generally not be used for real life applications and some more time-efficient alternatives have to be found.

8.5 Interactive Generation With Word Graphs

In [Och & Zens$^+$ 03], an efficient algorithm for interactive generation using word graphs was presented. Hereby, a word graph is defined as a weighted directed acyclic graph, in which each node represents a set of partial translation hypotheses and each edge is labeled with a word of the target sentence and is weighted according to the language and translation model scores. [Ueffing & Och$^+$ 02] gave a more detailed description of word graphs and showed how

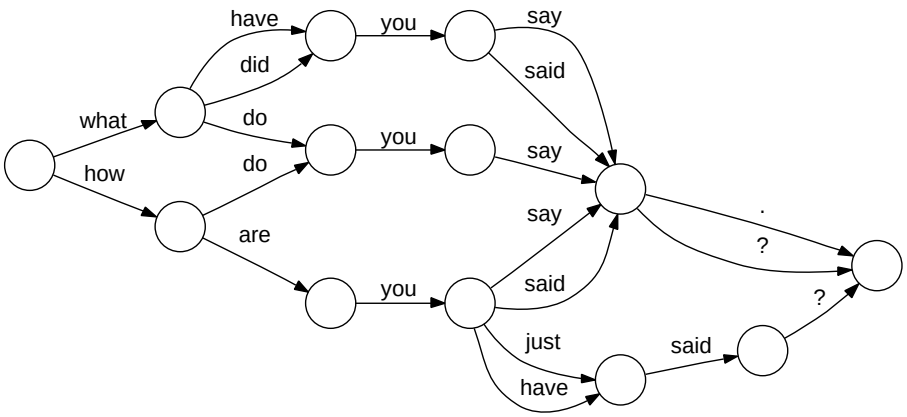

Figure 8.1. Example of a word graph for the German input sentence "was hast du gesagt?" (English reference translation: "what did you say?").

they can be easily produced as a sub-product of the search process. In fact, they presented an extension of the A^* algorithm resulting in possibly multiple incoming edges to the nodes in the search graph respectively word graph. An example of a word graph for the German input sentence "*was hast du gesagt?*" (English reference translation: "*what did you say?*") is shown in Figure 8.1.

It is obvious that each node in the word graph defines a set of prefixes of possible translations for the given source sentence. The main idea behind this approach is to find the node[4] that corresponds to the given prefix and to generate the best completion starting from this node. This can be easily accomplished using a forward-backward algorithm.

Since the word graph is a representation of a subset of the possible translation candidates for the source sentence, it can happen that the given prefix can not be found in the word graph. In this case, we look for the node with the minimum edit distance to the prefix and select the completion path with best backward score. The algorithm for computing the edit distance between a string and such a graph is a straightforward extension of the well-known Levenshtein algorithm for computing the distance between two strings.

The computational cost of this approach is much lower than that of the one presented in Section 8.4, as the whole search for the translation has to be carried out only once, and the generated word graph can be reused for further completion requests. This is also, of course, its main limitation, as the word graph does not automatically get adapted to the new information provided by the prefix. This can be alleviated to some extent by allowing a more flexible alignment of the generated sentences to the given prefix using the edit distance measure.

Another refinement can be added to the system. Usually, if the translation system was not able to find a completion in the generated word graph that is compatible with the last partial word in the prefix, the user has to type the whole completion. Instead, we now try to find the completion with highest probability using only the language model. This simple heuristic slightly increases the performance of the system, as words that were rejected in the pruning process can be recovered.

[4]In the general case, there can be more than one node that represents the same prefix, but with an appropriate determinization this case can be avoided.

8.5.1 Combination of Both Strategies

In order to overcome the limitations of the generation with word graphs, we can try to combine both strategies. We start by generating a word graph for the translations of the given source sentence and then use it for searching for completions. If, at a certain point, we determine that the generated graph does not correspond with the prefix typed by the user, we generate a new word graph tailored to this prefix with the method described in Section 8.4. An important point is how to decide if the word graph should not be used anymore and that a new one has to be generated. In our experiments, we use a simple heuristic: if the last word in the prefix is not complete (i.e. the prefix does not end with a blank space) and the selected node in the word graph does not produce a completion for this word, the word graph gets regenerated. This simple criterion already leads to an improved performance over the standard search strategy using word graphs. What still has to be determined is if the response time of the system, increased by the overhead of regenerating the word graphs, remains acceptable for interactive use under real-life conditions. Off-line experiments seem to indicate that this is the case (see Section 9.1.10).

8.6 Summary

In this chapter, we explained how the generation strategy for a state-of-the-art statistical machine translation system can be adapted for use in an interactive environment. The first approach consists in outputting only the translations compatible with the given prefix. Because this approach needs to perform a new search after each keystroke of the user, the real-time constraints of an interactive machine translation system do not allow to use this generation algorithm in practice. Furthermore, we reviewed an efficient generation process which first generates a word graph for a given source sentence and subsequently looks for completions of the prefixes within this word graph. The performance of the system degrades slightly but the search is performed in a much more efficient way. In the end, a combination of both strategies was proposed which improves the translation quality while still keeping an eye on the severe constraints of interactive MT. First off-line experiments in Section 9.1.10 show that the response time can be adequate for real-time responsiveness, although this has not been tested yet under real-life conditions.

Chapter 9

Experimental Results

In this chapter, we present the experimental findings obtained in the course of this thesis. For the most part, these are results produced for various types of translation tasks. We perform a detailed analysis for the Arabic-English IWSLT and GALE tasks, and also report on our decoder's performance for other tasks and language pairs. Additional experiments in the area of interactive machine translation are carried out on the Spanish-English and German-English Xerox corpora. In the second part of this chapter, we compare and combine different data-driven approaches to Arabic name transliteration. Experimental results are reported for the "10001 Arabic Names" and the "NGA Name Database" transliteration tasks.

9.1 Translation Tasks

To make the presentation of our approach to statistical machine translation complete, we begin with a brief sketch of the implemented training procedure. Training refers to estimation of the free model parameters from bilingual training data, and together with the modeling itself and the search problem the training makes up the three problems one has to deal with in SMT according to [Ney 01]. Next, we comment on the current research in the field of (automatic) evaluation of machine translation systems, describe the investigated translation tasks, and eventually close the part on translation with a detailed analysis of the experimental findings.

9.1.1 Training

Given a preprocessed training corpora for the source and target language, we first generate the word alignment using the IBM models [Brown & Della Pietra$^+$ 93] plus the Hidden-Markov alignment model (HMM) [Vogel & Ney$^+$ 96]. For that purpose, we employ the GIZA++ tool [Och & Ney 03] to perform the expectation maximization (EM) training of the word alignment models. We train a sequence of models which are of increasing complexity:

- In IBM-1, all alignments have the same initial probability, i.e. the distribution is uniform with value $\frac{1}{I+1}$.

- IBM-2 uses a zero-order alignment model $p(a_j|j, I, J)$ where different alignment positions are independent from each other.

- The HMM assumes a first-order model $p(a_j|a_{j-1}, I)$ where the alignment position a_j depends on the previous alignment position a_{j-1}. In the homogeneous version, the distance (or distortion) of the positions is modeled, i.e. $p(\ |a_j - a_{j-1}|\ |\ I)$.

- IBM-3 introduces an (inverted) zero-order alignment model $p(j|a_j, I, J)$ with an additional fertility model $p(\phi|e)$ which describes the number of words ϕ aligned to an English word e.

- In IBM-4, we have an (inverted) first-order alignment model $p(j|j', I, J)$ and a fertility model $p(\phi|e)$.

The models IBM-3 and IBM-4 are deficient. [Brown & Della Pietra$^+$ 93] therefore defined IBM-5 as a reformulation of IBM-4 to avoid deficiency. In our experiments, IBM-5 was not used. The training is performed in both translation directions, source-to-target and target-to-source.

As a result, we obtain two alignments a_1^J and b_1^I for each pair of sentences in the training corpus. Let $A_1 = \{(a_j, j) \,|\, a_j > 0\}$ and $A_2 = \{(i, b_i) \,|\, b_i > 0\}$ denote the sets of alignments in the two Viterbi alignments (source-to-target and target-to-source). In order to compile a generalized alignment and to increase the quality of the two Viterbi alignments, we combine A_1 and A_2 into a single alignment matrix A using the symmetrization heuristics described by [Och & Ney 03]:

- intersection: $A = A_1 \cap A_2$

- union: $A = A_1 \cup A_2$

- refined heuristic: We start determining the intersection $A = A_1 \cap A_2$, and then extend the alignment matrix A iteratively by adding alignments (i, j) occurring only in the alignment A_1 or in the alignment A_2 if neither f_j nor e_i has an alignment in A, or if (i, j) has a horizontal neighbor $(i-1, j), (i+1, j)$ or a vertical neighbor $(i, j-1), (i, j+1)$ that is already in A. We distinguish three variants of this refined heuristic that slightly differ in the merge operation.

The elements of the intersection result from both Viterbi alignments and are therefore very reliable. Obviously, this intersection yields an alignment consisting of only one-to-one alignments with a higher precision and a lower recall than either one separately. In contrast, the union of the two alignments yields a higher recall and a lower precision of the combined alignment than either one separately. The refined heuristic is intended to balance this trade-off. Whether a higher precision or a higher recall is preferred depends on the final application of the word alignment. In our SMT experiments, the refined method turned out to produce the best translation hypotheses.

Next, this generalized alignment matrix is the starting point for the phrase extraction. We determine the pairs of source and target phrases that are consistent with the word alignment from the bilingual training corpus using the extraction algorithm of [Zens & Och$^+$ 02]. The criterion is identical to the alignment template criterion described by [Och & Tillmann$^+$ 99], i.e. two phrases are considered to be translations of each other, if the words are aligned only within the phrase pair and not to words outside. Also, the phrases have to be contiguous. More details on the phrase extraction can be found in [Och 02].

At last, we train the free parameters of our decoder models (cf. Section 4.2) and of the rescoring models which are applied within our reranking framework (cf. Section 5.2). Today, the state-of-the-art is to perform minimum error rate training (MERT) [Och 03], i.e. the scaling factors are directly optimized with respect to some MT evaluation criterion on a development set and n-best development list, respectively. In our experiments, we applied MERT as well.

9.1.2 Evaluation Criteria

The (automatic) evaluation of machine translation systems is a research field on its own and has recently brought up a variety of different evaluation measures, where each of them has advantages and shortcomings. We distinguish between error rates and accuracy measures.

- Error rates:
  - WER (*Word Error Rate*):
    The WER has a long history and is the predominant evaluation measure in automatic speech recognition. It is computed as the minimum number of substitution, insertion and deletion operations that have to be performed to convert the translation hypothesis into the reference translation.
  - PER (*Position-independent word Error Rate*):
    [Tillmann & Vogel$^+$ 97] argued that the WER relies heavily on the perfect word order. Nevertheless, a hypothesis can be an acceptable translation even if its word order differs from that of the reference. In this case, the WER (alone) is somewhat misleading. The PER compares the words in the two sentences ignoring the word order.
  - TER (*Translation Edit Rate*):
    The TER is an extension of the WER and was defined by [Snover & Dorr$^+$ 06]. In addition to the standard edit operations substitutions, insertions and deletions, a new shift operation was introduced; shifts of whole phrases are permitted and counted as a single edit operation.

- Accuracy measures, i.e. larger scores indicate better translations:
  - Bleu score (*BiLingual Evaluation Understudy*):
    This score measures the precision of unigrams, bigrams, trigrams and fourgrams with respect to a reference translation. To increase the recall, it also adds a penalty for too short sentences [Papineni & Roukos$^+$ 02].
  - NIST score:
    The NIST score [Doddington 02] is similar to the Bleu score. It is a weighted n-gram precision in combination with a penalty for too short sentences.
  - METEOR (*Metric for Evaluation of Translation with Explicit Ordering*):
    For a given pair of hypothesis and reference strings, the evaluation proceeds in a sequence of stages, with different criteria being used at each stage to find and score unigram matches. By default, at the first stage all exact matches are detected between the two strings, while in the second stage the words not matched in the first stage are stemmed using the Porter stemmer[1] and then matches are found between these stemmed words. For further details, please refer [Banerjee & Lavie 05, Lavie & Agarwal 07].
  - GTM (*General Text Matcher*)
    GTM measures the similarity between texts in terms of the well-known mea-

[1]The Porter stemming algorithm (or Porter stemmer) is a procedure for removing the common morphological and inflexional endings from words in English. It is mainly used as part of a term normalization process that is usually done when setting up Information Retrieval systems, `http://tartarus.org/~martin/PorterStemmer/`.

sures precision, recall and the F-measure. For further details, please refer [Turian & Shen+ 03].

In this work, we mainly use the Bleu score and the TER to analyze the translation performance. Although criticized for favoring statistical systems over non-statistical systems [Callison-Burch & Osborne+ 06], the Bleu score is the official criterion for many MT evaluations. [Callison-Burch & Osborne+ 06] also showed that the Bleu score is appropriate for comparing variants of the same system, and is thus a good choice for the experimental analysis of this work. The TER, on the other hand, is the automatic counterpart of the HTER (*human* TER) which is the main criterion within the GALE project. As our systems were developed in the course of the GALE project, we focus on this criterion as well. If not stated otherwise, we report all error measures *case-insensitive*, and if available, we use multiple references to compute the evaluation criteria. Except for the IWSLT 2008 evaluation set, all translation hypotheses are scored using the RWTH "*EvalTrans - Fast Evaluation for MT Research*" toolkit[2]. Since the reference translations for the IWSLT 2008 evaluation set are not yet available, the scoring was done via the official IWSLT 2008 Automatic Evaluation Server[3].

As far as interactive machine translation is being investigated, the evaluation is based on the so-called *keystroke ratio* (KSR) introduced by [Och & Zens+ 03], which divides the number of keystrokes needed to produce the single reference translation (using the interactive translation system) by the number of keystrokes needed to simply type the reference translation. Hence, a keystroke ratio of 1 means that the system was never able to suggest a correct extension, whereas a small keystroke ratio means that the produced extensions are often correct. The KSR value is an indicator of the possible effective gain that can be achieved if this interactive translation system is used in a real translation task. Although the keystroke ratio is very optimistic with respect to the efficiency gain of a user, it was shown in the TransType 2 project [SchlumbergerSema S.A. & Intituto Tecnológico de Informática+ 01] that these measurements are correlated.

9.1.3 Task Descriptions and Corpus Statistics

BTEC

The *Basic Travel Expression Corpus* (BTEC) [Takezawa & Sumita+ 02] is a multilingual speech corpus which contains tourism-related sentences similar to those found in phrase books. The average source sentence length is around eight to nine words for all languages, so the task is rather limited and very domain-specific. The advantage, however, is that many different (reranking) experiments with varying settings can be carried out easily and quickly in order to analyze the effects of the different models, etc. BTEC is a rather clean corpus, so for the target language the preprocessing consisted mainly of separating punctuation marks[4] from words and replacing contractions such as *it's* or *I'm*. In the experiments, we made use of two different subsets of this corpus:

[2]The EvalTrans software was developed at RWTH Aachen University - Department of Computer Science by Gregor Leusch and Sonja Niessen, and is freely available: `http://www-i6.informatik.rwth-aachen.de/web/Software/EvalTrans/index.html`

[3]The official IWSLT 2008 Automatic Evaluation Server: `https://www.slc.atr.jp/EVAL/IWSLT08/automatic/testset_IWSLT08/`

[4]Partly, the provided data sets contain punctuation marks and partly they are missing as the focus of the IWSLT campaign is more and more moving towards the translation of recognized speech.

- We used the Arabic-English BTEC data made available to the IWSLT 2008 evaluation participants to analyze different (Arabic) preprocessing approaches, to compare the case handling strategies for English, and to perform a detailed analysis of the search process. For the IWSLT 2008 BTEC Arabic-English task, the training corpus contains 20 000 sentences. We decided to use the IWSLT 2004 evaluation set as development set, to use the IWSLT 2005 evaluation set for blind tests, and added the remaining test sets to the training data. We also report results for the official IWSLT 2008 evaluation set. The corpus statistics are shown in Table 9.1. The numbers of running and out-of-vocabulary (OOV) words for the English test sets are calculated for the full 16 reference translations. As the recent evaluation sets reflect the campaign's focus on speech translation, they do not contain any punctuation marks for the source language part and the official scoring accounts only for the main sentence-end punctuation for the target language. Thus, we proceeded as described in Section 6.4, i.e. we trained the word alignment as usual with punctuation marks present in the source and target part of the bilingual training corpus. Then, we removed all the punctuation marks from the source part of the corpus and the unaccounted ones from the target part of the corpus, adjusting the word alignment indices. In addition, we used the English part of the so-called HIT-corpus[5] to enhance the LM training data. The HIT-corpus is a multi-source Chinese-English parallel corpus (including a proportion of spoken language) intended for speech translation.

- To investigate the use of syntactically motivated feature functions within a reranking framework, we employed the multilingual BTEC data provided for the IWSLT 2005 Supplied Data Track. Translation directions include Arabic-English, Chinese-English and Japanese-English. As for the 2008 evaluation, the training corpus contains 20 000 sentences. However, these sentences are distinct from the 2008 data release. Two test sets are available: we used the C-Star 2003 set for development and tuning of the system's parameters. After that, the IWSLT 2004 set was used as a blind test set in order to measure the performance of the models. There was no special preprocessing for the source language corpora. The corpus statistics are shown in Table 9.2.

For both IWSLT tasks, the translation hypotheses were scored using multiple references.

GALE

The DARPA funded GALE (*Global Autonomous Language Exploitation*) program aims at developing and applying computer software technologies to absorb, translate, analyze, and interpret huge volumes of speech and text in multiple languages. This also involves the processing of noisy and unstructured input data, e.g. data collected from newsgroups or weblogs, or automatic transcriptions of arbitrary broadcast conversations. Within GALE, three research teams participate and evaluate their MT engines against each other. As said before, the SMT system presented in this work is the primary MT engine of the SRI Nightingale team for the Arabic-English tasks. In the series of GALE evaluations, the systems had to translate:

- text input out of two different domains: newswire (NW) texts as were used for most previous MT evaluations and web texts (WT) derived from newsgroups and weblogs, as well as

[5]The HIT-corpus was released to IWSLT 2008 participants, `http://mitlab.hit.edu.cn/index.php/resources/233-information-for-iwslt-2008-participants.html`.

Table 9.1. Corpus statistics of the Arabic-English BTEC task (IWSLT 2008 data sets, BTEC Task, OOV: out-of-vocabulary words).

			Arabic	English
Train		Sentence pairs	23 940	
		Running words	188 892	197 179
		Vocabulary size	19 640	14 672
		Singletons	9 679	7 216
Dev	2004	Sentences	500	
		Running words	3 261	62 517
		OOV	146	2 335
Eval	2005	Sentences	506	
		Running words	3 171	63 525
		OOV	151	2 440
	2008	Sentences	507	
		Running words	2 585	–
		OOV	218	–

- recorded speech out of two different domains: broadcast news (BN) and broadcast conversations (BC) which are focused more on discussions and call-ins that have a conversational style of speech.

The large-scale and multi-domain characteristics of the GALE project already pose a lot of problems to the MT systems. Nevertheless, the Program Manager aimed high when defining the project's performance goals[6]:

> "The ultimate performance targets are to translate Arabic and Chinese speech and text with 95% accuracy and an extremely high degree of consistency (90-95%), and to extract and deliver key information with proficiency matching or exceeding that of humans. GALE systems must be able to perform at these high levels of accuracy and consistency for foreign language information from a wide range of domains and genres, and be able to cope with informal and colloquial language."

To strive for these goals, we made use of all Arabic-English bilingual corpora provided by the Linguistic Data Consortium (LDC)[7]:

- UN Arabic English Parallel Text (LDC2004E13)
- Arabic News Translation Text Part 1 (LDC2004T17)
- Arabic English Parallel News Part 1 (LDC2004T18)
- Arabic Treebank English Translation (LDC2005E46)

[6] Global Autonomous Language Exploitation (GALE): `http://www.darpa.mil/ipto/programs/gale/gale_goals.asp`

[7] The Linguistic Data Consortium supports language-related education, research and technology development by creating and sharing linguistic resources: data, tools and standards, `http://www.ldc.upenn.edu/`.

Table 9.2. Corpus statistics of the multilingual BTEC task (IWSLT 2005 data sets, Arabic-English, Chinese-English and Japanese-English translation directions, Supplied Data Track, OOV: out-of-vocabulary words, after preprocessing).

			Arabic	Chinese	Japanese	English
Train		Sentence pairs	20 000			
		Running words	180 075	176 199	198 453	189 927
		Vocabulary size	15 371	8 687	9 277	6 870
		Singletons	8 319	4 006	4 431	2 888
Dev	2003	Sentences	506			
		Running words	3 552	3 630	4 130	3 823
		OOV	133	114	61	65
Eval	2004	Sentences	500			
		Running words	3 597	3 681	4 131	3 837
		OOV	142	83	71	58

- eTIRR Arabic English News Text (LDC2004E72)
- Multiple-Translation Arabic (MTA) Part 1 (LDC2003T18)
- Multiple-Translation Arabic (MTA) Part 2 (LDC2005T05)
- TIDES MT 2004 Arabic evaluation data (LDC2006E44)
- TIDES MT 2005 Arabic evaluation data (LDC2006E39)
- Arabic Treebank (LDC2005T20)
- Buckwalter Arabic Morphological Analyzer (LDC2004L02)
- ISI Arabic-English Corpus (LDC2007E07)
- Word-Aligned corpus (LDC2006G09)

In addition, we included the data sources that were specifically provided for the GALE participants, e.g. the FBIS, SAKHR, INTERIM, EXTRADRYRUN corpora, some domain-specific training data releases and some Arabic name lists. The corpus statistics of the Arabic-English GALE task are depicted in Table 9.3. All the test sets contain just single references. The huge vocabulary sizes and the immense numbers of singletons show that the GALE task is not feasible for SMT without further preprocessing and simplification of the training data. OOV numbers are rather small for the NW task, still manageable for the speech input (BN and BC), and might be problematic only for the WT domain.

The language model was trained on the English part of the bilingual training corpus and additional monolingual data available, e.g. GigaWord v2, TDT, BBN data, ... (cf. Section 6.2). We ran modified Kneser-Ney smoothing as implemented in the SRILM toolkit [Stolcke 02] and used the default setting for discarding low-frequency n-grams, which means that singletons are discarded for order three and higher.

Table 9.3. Corpus statistics of the Arabic-English GALE task (GALE 2007 MT training data, GALE 2007 MT development sets, GALE 2006 MT evaluation sets, OOV: out-of-vocabulary words).

			Arabic	English
Train		Sentence pairs	7.6 M	
		Running words	158 M	180 M
		Vocabulary size	2 M	1.3 M
		Singletons	1 M	687 K
Dev	NW	Sentences	686	
		Running words	22 225	30 308
		OOV	172	105
	WT	Sentences	907	
		Running words	18 805	27 848
		OOV	485	119
	BN	Sentences	1 112	
		Running words	22 916	34 371
		OOV	388	81
	BC	Sentences	640	
		Running words	11 331	17 150
		OOV	200	26
Eval	NW	Sentences	474	
		Running words	11 529	15 553
		OOV	86	75
	WT	Sentences	527	
		Running words	9 656	14 583
		OOV	548	74
	BN	Sentences	956	
		Running words	11 906	18 199
		OOV	294	87
	BC	Sentences	529	
		Running words	11 518	17 070
		OOV	217	39

Development and tuning of our decoder's parameters was done for the GALE 2007 MT development sets. The GALE 2006 MT evaluation sets were used as a blind test set. We ran a series of experiments to investigate the key topics of this thesis.

As our SMT system was trained on lower case data, we had to restore the case information after translation for the official evaluation campaigns. This was done using the SRI disambig tool and a fourgram language model (trained with case information, respectively).

XEROX

To compare different search strategies for interactive machine translation, various experiments were performed on the Spanish-English and the German-English Xerox corpora which consist of the translation of technical Xerox manuals. We have chosen to use these corpora in the raw

Table 9.4. Corpus statistics of the Spanish-English Xerox task (Raw Data Task, OOV: out-of-vocabulary words).

		Spanish	English
Train	Sentence pairs	55 761	
	Running words	752 166	666 700
	Vocabulary size	16 362	13 541
	Singletons	5 046	3 725
Dev	Sentences	1 012	
	Running words	15 999	14 352
	OOV	95	55
Eval	Sentences	1 125	
	Running words	10 226	8 521
	OOV	250	222

Table 9.5. Corpus statistics of the German-English Xerox task (Raw Data Task, OOV: out-of-vocabulary words).

		German	English
Train	Sentence pairs	49 376	
	Running words	537 464	589 531
	Vocabulary size	23 845	13 223
	Singletons	9 443	3 681
Dev	Sentences	964	
	Running words	10 462	10 642
	OOV	147	29
Test	Sentences	996	
	Running words	11 704	12 298
	OOV	485	141

data format. The corpora allocations are summarized in Table 9.4 and Table 9.5.

Our extensions to enable different search strategies for interactive MT were incorporated into an older SMT decoder [Bender & Zens$^+$ 04] based on the work of [Och & Tillmann$^+$ 99, Och 02]. After training and optimization of the model scaling factors, the SMT engine was run to translate the test corpus. Using the same parameter settings, a simulation of the interactive mode was carried out. This simulation mode was described by [Och & Zens$^+$ 03]. The system with the same parameter settings was also successfully used by human translators to evaluate it under real-life conditions. Due to the high effort that human evaluations require, only the word-graph based generation strategy was tested. The response time of the system was adequate.

Table 9.6. Official results of the NIST 2006 Machine Translation Evaluation (Arabic-to-English task, Large Data Track, NIST subset, BLEUr4n4 [%], case-sensitive evaluation).

System	BLEU
Google	42.8
IBM	39.5
Information Sciences Institute	39.1
This work	39.1
Applications Technology Inc.	38.7
Language Weaver	37.4
BBN Technologies	36.9
NTT Communication Science Laboratories	36.8
ITC-irst	34.7
Carnegie Mellon University & University of Karlsruhe	33.7
University of Maryland & Johns Hopkins University	33.3
University of Edinburgh	33.0
Sakhr Software Co.	33.0
National Institute of Information and Communications Technology	29.3
Queen Mary University of London	29.0
Language Computer	27.8
Universitat Politecnica de Catalunya	27.4
Columbia University	24.7
University of California Berkeley	19.8
The American University in Cairo	15.3
Dublin City University	9.5
Kansas State University	5.2

9.1.4 Comparison With Other Research Groups

In this subsection, we compare the results obtained in this work with the results of other research groups. Table 9.6 shows the official results of the NIST 2006 Machine Translation Evaluation (MT-06) for the Arabic-to-English task (Large Data Track, NIST subset)[8]. The series of NIST evaluations is generally accepted to be the main evaluation campaign in the MT community and therefore gives the best overview of the state-of-the-art in current MT systems. Moreover, the NIST subset consisted of documents drawn from the newswire, newsgroup and broadcast news domains. Next, Table 9.7 contains the results obtained by the individual research groups that make up the SRI Nightingale team while preparing for the re-evalution as part of the 2007 GALE Phase 2 Translation evaluation (GALE-07). The SRI Nightingale team was ranked second of the three GALE teams in GALE-07. For system combination, the GALE 2006 evaluation sets were used as development data while the GALE 2007 development sets served as "test" data. The scores for the system combination correspond to the approach to the official GALE-07 Nightingale submission. Finally, Table 9.8 compares the results of the IWSLT 2008 evaluation campaign with this work. Although the official RWTH submission was a combination of several

[8]For an overview of the entire MT-06 evaluation results, see `http://www.itl.nist.gov/iad/mig/tests/mt/2006/doc/mt06eval_official_results.html`.

Table 9.7. Results of the individual SRI Nightingale team members for the Arabic-English GALE task (GALE Phase 2 Translation evaluation, GALE 2007 MT development and GALE 2006 MT evaluation sets, BLEUr4n4/TER [%]).

		GALE 06		DEV 07	
TASK	SYSTEM	BLEU	TER	BLEU	TER
NW	Applications Technology Inc.	15.5	64.1	20.5	57.9
	Columbia University	23.6	53.4	25.3	51.2
	University of Washington	24.5	58.1	16.7	81.2
	SRI International	29.2	54.0	31.9	50.1
	RWTH hierarchical system	25.9	57.4	28.4	52.6
	This work	30.6	51.6	32.1	48.7
	System combination	32.1	48.7	33.1	46.4
WT	Applications Technology Inc.	8.5	72.1	15.7	64.3
	Columbia University	11.7	66.8	17.2	60.4
	University of Washington	14.1	69.3	6.5	105.9
	SRI International	17.9	63.7	23.9	61.2
	RWTH hierarchical system	15.1	71.8	20.4	64.0
	This work	17.9	66.0	25.2	58.2
	System combination	19.4	60.4	25.4	55.5
BN	Applications Technology Inc.	8.9	73.7	–	–
	Columbia University	13.6	65.3	22.0	55.9
	University of Washington	16.9	65.4	24.0	56.3
	SRI International	22.4	60.9	32.3	49.5
	This work	22.9	60.8	33.1	48.8
	System combination	23.2	58.7	33.2	47.1

MT systems including n-best rescoring, we were able to slightly improve these scores by the use of single-best translations only.

In general, we observe that the system presented in this work is competitive with the best systems on all the tasks.

9.1.5 Comparison of Different Preprocessing Approaches

To begin with a detailed analysis of the methods presented in this work, we first of all compare the investigated approaches to Arabic preprocessing which were presented in Chapter 3:

- IFSA:
  the improved finite state automaton-based approach as described in [Isbihani & Khadivi$^+$ 06],

- MADA:
  the method applying the disambiguation tool [Habash & Rambow 05], here we further analyze:
  - the D2 scheme,

Table 9.8. Official results of the IWSLT 2008 evaluation campaign (BTEC Arabic-English task, Correct Recognition Result, (BLEU+METEOR)/2, BLEU, METEOR). The results are in percentage and scored case-sensitive.

System	(B+M)/2	BLEU	METEOR
MIT Lincoln Laboratory	63.7	56.5	70.8
UPC, TALP Research Center	60.6	52.6	68.5
RWTH	60.0	53.5	66.5
University of Le Mans, LIUM	58.5	49.4	67.7
TBITAK-UEKAE	57.9	48.0	67.9
University J. Fourier, GETALP, LIG	56.7	46.0	67.5
Dublin City University, School of Computing	56.0	47.2	64.9
Pohang University of Science and Technology	50.3	38.8	61.8
Queen Mary University of London	40.7	29.0	52.5
University of Caen Basse-Normandie, GREYC	33.9	22.1	45.6
This work	60.3	53.8	66.8

  - the Atb scheme,
  - the effect of Norm, i.e. normalization of *Yaa* and *Alef*,
  - the effect of noOrth where we disable the orthographical normalization module for word stems,

- and MTAG:
  the word segmentation inferred from the morphological POS tagger of [Mansour & Sima'an$^+$ 07].

In Table 9.9 and Table 9.10, the resulting corpus statistics and the effect of the different preprocessing approaches on the translation quality are shown for the Arabic-English BTEC task. Translation quality is measured in terms of the case-sensitive Bleu score. The IFSA, MADA-Atb and MTAG methods perform a quite rigorous word splitting in comparison to the other MADA variants. This can be seen at the relatively high number of running words but rather small vocabulary size and number of singletons at the same time. Moreover, the IFSA method seems to be prone to different corpus characteristics as the number of out-of-vocabulary words (OOVs) is twice as high for the IWSLT 2004 development set in comparison to the number of OOVs for the other approaches. The statistics for the other two test sets are more balanced. When looking at the translation quality, these observations are also reflected in the Bleu scores. MTAG and MADA-Atb yield the best translation hypotheses with an outlier for the IWSLT 2005 evaluation set for which the IFSA method achieves the highest score. Considering the results for all three eval sets, the MTAG approach is the first choice. Therefore, all of the following experiments carried out on the Arabic-English BTEC task are based on MTAG-preprocessed data.

Furthermore, we wanted to analyze the effect on the translation quality not only for the source language preprocessing but also for different case handling strategies for English. Clearly, the most obvious strategies are to either retain the case information as provided in the data sets or to lowercase the target text. The first approach keeps the full information while accepting

Table 9.9. Arabic corpus statistics of the BTEC task for different preprocessing approaches (IWSLT 2008 data sets, OOV: out-of-vocabulary words).

			IFSA	MADA				MTAG
				D2	Atb	Norm	noOrth	
Train		Sentence pairs	23 940					
		Running words	198 142	167 079	185 592	167 079	167 079	186 424
		Vocabulary size	14 039	15 531	13 744	15 308	15 555	13 925
		Singletons	6 119	7 040	5 936	6 917	7 054	6 036
Dev	2004	Sentences	500					
		Running words	3 175	2 780	3 149	2 780	2 780	3 142
		OOV	190	99	82	96	99	93
Eval	2005	Sentences	506					
		Running words	3 254	2 700	3 041	2 700	2 700	3 031
		OOV	96	113	91	110	114	99
	2008	Sentences	507					
		Running words	2 996	2 730	3 059	2 730	2 730	3 047
		OOV	125	144	111	139	146	116

Table 9.10. Effect of different approaches to Arabic preprocessing on the translation quality (BLEUr4n4 [%]) for the BTEC task (IWSLT 2004, 2005 and 2008 evaluation sets, case-sensitive evaluation).

	Dev	Eval	
Preprocessing	2004	2005	2008
IFSA	52.8	59.0	49.3
MADA-D2	54.6	56.5	50.7
MADA-Atb	55.2	56.2	51.2
MADA-Norm	54.4	55.9	49.8
MADA-noOrth	54.4	56.0	50.4
MTAG	55.4	57.0	51.5

multiple vocabulary and lexicon entries which only differ in their case. The latter method on the other hand removes all of these ambiguities but instead requires to restore the correct case information in a postprocessing step. In addition, we kept the "true case" information but for each word, the most frequent case was determined and substituted. The case-sensitive Bleu scores for the three strategies are presented in Table 9.11. For the Arabic-English BTEC task, the frequent-case strategy nicely pays off as there is a significant improvement in translation quality over the true case and lower case strategies.

If we look at the equivalent experiments for the Arabic-English GALE task, the findings are different. Table 9.12 and Table 9.13 show the resulting corpus statistics and the effect of the different preprocessing approaches on the translation quality for this task. Bleu scores are reported case-insensitive. Comparing the statistics of the raw GALE data, see Table 9.3, to the numbers obtained after running the different preprocessing approaches, it can be seen that each of the methods is able to reduce the complexity of the GALE task significantly. All methods

Table 9.11. Effect of different case handling strategies for English on the translation quality (BLEUr4n4 [%]) for the BTEC task (IWSLT 2004, 2005 and 2008 evaluation sets, case-sensitive evaluation).

	Dev	Eval	
Case Handling	2004	2005	2008
true case	53.7	55.4	50.9
lower case	53.9	56.3	50.8
frequent case	55.4	57.0	51.5

roughly lower the vocabulary size and the number of singletons by a factor of four. Still, the subtle differences in the word splitting and orthographic normalization strategies can be noticed when looking at the details or when considering the translation results. For instance, the MTAG approach which yielded the best translations for the clean and small BTEC data clearly falls behind the other approaches for this task. Apparently, the morphological POS tagger fails when being confronted with diverse, noisy and unstructured input. As can be seen from Tables 9.12 and 9.13, the number of OOVs is considerably higher and the translation quality is inferior to the one obtained by the best methods. Here, the MADA-Atb method is the first choice for the text domains, whereas the IFSA approach achieved the best translations for the speech input. For the moment, speech input is to be perceived as the correct transcriptions of the acoustic data, we go into the actual details of speech translation in Subsection 9.1.9. Moreover, the translation quality obtained on the text data using the IFSA method is nearly on the same level as the quality of the best MADA-Atb translations, the Bleu score for the GALE 2006 NW evaluation set is even the single best score on that set. At the same time, the IFSA method is completely unsupervised and a rather straightforward but very efficient FSA for splitting off prefixes and suffixes. In contrast, the MADA variants additionally involve a POS tagger and a classifier for disambiguation of the morphological analyses. This causes the MADA runs to be very expensive in terms of computing power and time, especially when dealing with millions of training sentences as for the GALE task.

Unfortunately, we were not able to carry over the nice improvements obtained from the frequent case handling for BTEC to large-scale translations tasks such as GALE. Here, either strategy was more or less on par with the others.

9.1.6 Effect of Different Heuristics for Alignment Symmetrization

Inspired by [Och & Ney 00, Och & Ney 03] who reported improvements in translation quality due to the proper choice of the heuristic for symmetrization of the word alignments (source-to-target and target-to-source), we investigated these effects also for the Arabic-English BTEC task. The Bleu and TER scores for the different heuristics are shown in Table 9.14. Hereby, the baseline is defined as the system build on the single word alignment trained for the standard translation direction ($f \rightarrow e$). Except for the union of the two alignments, all the symmetric alignments enhance either the Bleu or TER score, or both. This is contrary to the statement of [Och & Ney 00] that a higher recall (as a result of the union) is more important than a high precision (yielded by the intersection) for SMT tasks. However, the refined heuristic intended to balance this trade-off generated the best translation hypotheses. In a last experiment, we

Table 9.12. Arabic corpus statistics of the GALE task for different preprocessing approaches (GALE 2007 MT training data, GALE 2007 MT development sets, GALE 2006 MT evaluation sets, OOV: out-of-vocabulary words).

			IFSA	MADA				MTAG
				D2	Atb	Norm	noOrth	
Train		Sentence pairs	7.4 M					
		Running words	186 M	188 M	194 M	188 M	188 M	194 M
		Vocabulary size	565 K	564 K	493 K	557 K	566 K	515 K
		Singletons	263 K	243 K	219 K	241 K	243 K	228 K
Dev	NW	Sentences	686					
		Running words	26 583	26 010	27 219	26 010	26 010	27 856
		OOV	124	85	73	85	85	352
	WT	Sentences	907					
		Running words	22 608	21 883	23 548	21 883	21 883	24 009
		OOV	310	226	172	220	226	319
	BN	Sentences	1 112					
		Running words	26 673	26 294	27 651	26 294	26 294	28 076
		OOV	188	154	142	149	154	374
	BC	Sentences	640					
		Running words	12 880	12 772	13 471	12 772	12 772	13 577
		OOV	108	80	70	79	80	128
Eval	NW	Sentences	474					
		Running words	14 095	13 866	14 436	13 866	13 866	14 882
		OOV	71	48	43	47	48	215
	WT	Sentences	527					
		Running words	11 597	11 324	12 275	11 324	11 324	12 468
		OOV	326	293	236	291	293	293
	BN	Sentences	956					
		Running words	13 833	13 547	14 290	13 547	13 547	14 418
		OOV	171	179	161	177	179	240
	BC	Sentences	529					
		Running words	13 465	13 101	13 903	13 101	13 101	14 074
		OOV	132	139	126	138	139	224

simply repeated the training corpus several times and concatenated the baseline and symmetric alignments. We then trained our system on the duplicated data. In terms of the translation quality, the concatenation was helpful as there is an improvement over all single scores.

The corresponding results for the GALE task are similar, i.e. also for large-scale domains the refined heuristics outperform the baseline standard alignment, but the concatenation of the single alignments does not result in any further improvement of the translation quality.

9.1.7 Analysis of the Search

To understand the search for SMT in more detail, we begin with an analysis of the effect of the different models used during decoding. These first-pass models were delineated in Section 4.2. Again, we start with the Arabic-English BTEC task; the Bleu and TER scores for the IWSLT 2004 and 2005 evaluation sets are reported in Table 9.15. The initial experiments just used a standard n-gram language model as well as the word and phrase penalties to score the

Table 9.13. Effect of different approaches to Arabic preprocessing on the translation quality (BLEUr4n4 [%]) for the GALE task (GALE 2007 MT development and GALE 2006 MT evaluation sets).

	Dev				Eval			
Preprocessing	NW	WT	BN	BC	NW	WT	BN	BC
IFSA	29.6	18.9	30.0	28.5	27.6	15.1	22.0	21.8
MADA-D2	29.7	18.7	29.5	27.7	27.1	14.9	21.2	21.6
MADA-Atb	30.3	19.2	29.5	28.1	27.4	15.3	21.2	21.6
MADA-Norm	29.5	18.7	29.7	27.6	26.9	14.0	20.9	21.7
MADA-noOrth	29.6	18.2	29.3	27.7	27.0	14.1	20.7	21.7
MTAG	28.2	18.7	28.1	27.2	24.9	14.5	20.5	21.2

Table 9.14. Effect of different heuristics for alignment symmetrization on the translation quality (BLEUr4n4/TER [%]) for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

	Dev 2004		Eval 2005	
Alignment	BLEU	TER	BLEU	TER
Baseline				
standard alignment ($f \rightarrow e$)	55.2	34.4	57.1	32.6
Symmetric Alignment				
intersection	56.0	34.1	57.1	32.9
union	54.4	35.1	56.0	32.4
refined method a	54.9	33.9	57.0	32.3
refined method b	55.7	34.1	57.0	32.5
refined method c	56.3	33.9	58.1	31.8
Concatenation of Alignments	56.8	33.6	58.5	31.3

translation candidates during search. We experimented with n-grams up to order eight. This setup was intended to measure the complexity of the translation task, and the numbers indicate that the BTEC task is indeed rather simple. For the actual translation setups, the simplest system used the language model plus the "standard" phrase model $p(\tilde{f}|\tilde{e})$. We then added the "inverted" phrase model $p(\tilde{e}|\tilde{f})$ and the penalties one by one. As can be drawn from the table, each of the added models improved the translation quality, both in terms of Bleu and TER score, and for both test sets. Next, we see that the Noisy-Or word lexica outperform the standard IBM word lexica. Furthermore, there is an additional benefit from using a combination of the lexica obtained for both translation directions. This is in contrast to the conclusion drawn by [Zens 08] for Chinese-English translation tasks, where the Noisy-Or word lexica also outperform the IBM word lexica but where there is no further improvement by combining the word lexica for the two translation directions. Note that so far, these are only phrase/lexicon models which are more or less independent from reordering. The effect of alternative word reorderings unveils

Table 9.15. Effect of different models on the translation quality (BLEUr4n4/TER [%]) for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

	Dev 2004		Eval 2005	
Models	BLEU	TER	BLEU	TER
Language Model				
LM + WP	31.0	62.7	32.5	60.4
LM + WP + PP	40.7	51.8	40.8	50.9
+ Phrase Models				
LM + $p(\tilde{f}\|\tilde{e})$	54.9	34.8	57.3	33.5
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$	56.2	34.2	58.2	32.2
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP	57.0	33.5	59.4	31.4
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP + PP	57.3	33.1	59.8	30.9
+ Lexicon Models				
IBM-1 $p(f\|\tilde{e})$	56.9	33.5	59.4	31.5
IBM-1 $p(e\|\tilde{f})$	57.1	33.2	59.7	30.9
IBM-1 both	57.3	33.0	59.9	30.9
Noisy-Or $p(f\|\tilde{e})$	58.0	32.4	60.1	30.6
Noisy-Or $p(e\|\tilde{f})$	56.9	34.0	59.3	31.6
Noisy-Or both	58.2	32.3	60.1	30.4
+ Distortion Model				
distortion penalty	58.9	32.4	60.3	30.9

when enabling the distortion model. However, at least for the Arabic-English BTEC task the reordering problem seems to be a minor one; the Bleu score increases by just 0.7% on the development set and even less by 0.2% for the test data, the TER scores are similar.

Although the log-linear modeling approach enables us to easily combine lots of models and to integrate any additional feature function, fewer models are preferable in practice. In our experimental setup, we try for not using more than ten different models. Otherwise, the Downhill-Simplex training will slow down significantly as it will more likely fail to converge to the optimal values. Consequently, the SMT system will encounter suboptimal parameter settings. Moreover, the less models have to be trained, the less is the chance of overfitting. This becomes even more important when applying the various types of rescoring models.

We repeated these experiments for the Arabic-English GALE MT from text tasks; Table 9.16 contains the translation scores obtained for the newswire (NW) domain, and Table 9.17 shows the corresponding numbers for the web text (WT) documents. The GALE MT from text tasks are much harder as can be seen from the fact that it is not possible to generate any useful translations without applying the phrase/lexicon models. The huge vocabulary size and the unrestricted domains simply do not allow for any reliable guess based only on the n-gram statistics. Otherwise, the general observations made for the BTEC task remain more or less the same for the GALE translations, although there is much more fluctuation in the scores due to the diverse and noisy nature of the GALE documents. As the individual systems were tuned for maximum Bleu scores, the single models helped to increase the Bleu score one by one

Table 9.16. Effect of different models on the translation quality (BLEUr4n4/TER [%]) for the Arabic-English GALE newswire task (GALE 2007 MT development set, GALE 2006 MT evaluation set).

	Dev		Eval	
Models	BLEU	TER	BLEU	TER
Language Model				
LM + WP	4.7	214.3	4.5	202.8
LM + WP + PP	14.1	95.7	13.6	99.5
+ Phrase Models				
LM + $p(\tilde{f}\|\tilde{e})$	24.1	59.3	22.6	61.6
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$	25.2	56.6	23.2	58.9
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP	26.8	58.2	25.0	63.5
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP + PP	27.3	57.3	25.7	60.5
+ Lexicon Models				
IBM-1 $p(f\|\tilde{e})$	29.6	54.7	28.3	56.1
IBM-1 $p(e\|\tilde{f})$	30.0	51.1	28.3	54.0
IBM-1 both	29.5	52.7	27.7	57.5
Noisy-Or $p(f\|\tilde{e})$	29.7	53.3	28.3	54.0
Noisy-Or $p(e\|\tilde{f})$	29.6	51.3	28.2	55.9
Noisy-Or both	30.1	50.8	28.7	54.2
+ Distortion Model				
distortion penalty	31.3	50.5	29.2	53.5

but these gains did not always carry over to reductions in TER. Nevertheless, the best settings achieved significant improvements for both evaluation measures, for both test sets, and for both translation tasks.

When taking the results for all three tasks into account, we can summarize that the phrase models, the word penalty, and the proper choice of the word lexicon share a particular impact on the translation quality for the Arabic-English language pair. Interestingly, the IBM word lexica perform better for the inverted direction $p(e|\tilde{f})$ whereas the Noisy-Or models work better for the standard direction $p(f|\tilde{e})$. In general, the combination of the two (Noisy-Or) word lexica results in an additional performance gain. Of course, it does not make sense to generate translations without using the language model. By comparison, the phrase penalty and the distortion penalty model only have a minor influence on the translation hypotheses. According to that, reordering indeed seems to be a less critical issue for Arabic-English SMT; we come back to this point shortly when dealing with the different pruning strategies.

Another way to analyze the contribution of the single models to the achieved translation quality is via the effect of the corresponding model scaling factors of our decoder. In the previous experiments, the primal evaluation measure was the Bleu score, therefore we now focused on the TER score and ran a series of experiments for the Arabic-English BTEC task. For the IWSLT 2004 and 2005 evaluation sets, we took the optimized values for the decoder parameters, kept all but one fixed, and produced translations for several values of the free parameter. We repeated this procedure for each of the decoder's model scaling factors and

Table 9.17. Effect of different models on the translation quality (BLEUr4n4/TER [%]) for the Arabic-English GALE web text task (GALE 2007 MT development set, GALE 2006 MT evaluation set).

	Dev		Eval	
Models	BLEU	TER	BLEU	TER
Language Model				
LM + WP	2.9	221.2	3.2	219.8
LM + WP + PP	5.3	174.6	4.8	186.0
+ Phrase Models				
LM + $p(\tilde{f}\|\tilde{e})$	17.0	67.7	13.2	72.1
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$	16.5	70.0	13.1	74.3
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP	18.6	69.2	15.1	72.7
LM + $p(\tilde{f}\|\tilde{e})$ + $p(\tilde{e}\|\tilde{f})$ + WP + PP	18.9	68.4	15.6	76.8
+ Lexicon Models				
IBM-1 $p(f\|\tilde{e})$	19.3	69.6	16.0	76.3
IBM-1 $p(e\|\tilde{f})$	20.6	63.8	16.7	70.1
IBM-1 both	20.9	64.5	16.9	71.0
Noisy-Or $p(f\|\tilde{e})$	20.7	65.5	16.5	68.8
Noisy-Or $p(e\|\tilde{f})$	19.2	67.6	15.7	72.3
Noisy-Or both	21.0	60.9	16.7	66.1
+ Distortion Model				
distortion penalty	21.8	61.6	16.8	65.9

plotted the obtained TER score against the value of the investigated model scaling factor. In this way, the plots shown in Figures 9.1 to 9.10 were produced. During optimization, the model scaling factors are normalized such that the absolute values sum up to one (L_1 norm), i.e. the optimal TER scores are always achieved for small values around zero. As can be drawn from the various curves, the overall findings correspond to those obtained for the previous experiments; the biggest impact on the produced translations can be made by varying the scaling factors for the phrase models, the language model, the word-based lexicon models, and the word penalty model. For each of these models, the range of scaling factor values that generate high quality translations is sharply defined. The effect of the phrase penalty and the phrase count models is rather small. Therefore, the phrase count models should not be used in practice to speed up the training and optimization processes and to reduce the risk of overfitting. The phrase penalty model can be beneficial for adjusting the length of the translation hypotheses though.

Next, we underline the improvements for beam search in SMT that have been achieved in the course of this work, and thereto analyze the experimental results obtained for the pruning strategies presented in Section 4.3. In [Bender & Zens$^+$ 09], we stated that it is important to focus on alternative reorderings, whereas already a small number of lexical alternatives is sufficient to achieve good translation quality. We carried out experiments on the investigated Arabic-English translation tasks as well as for the Chinese-English GALE newswire task[9] in

[9]Special thanks to Arne Mauser and Richard Zens for providing us with the data and decoder settings to be able to run the experiments for the Chinese-English translation direction.

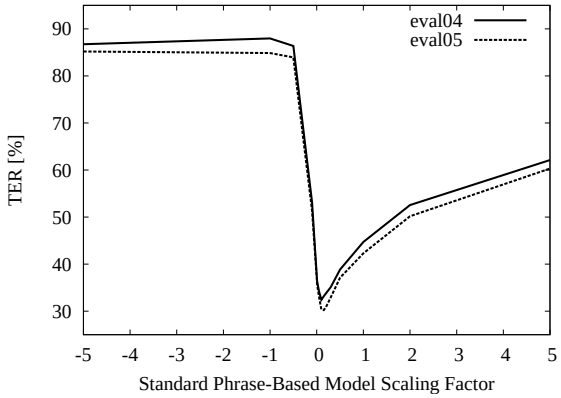

Figure 9.1. Effect of the standard phrase-based model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

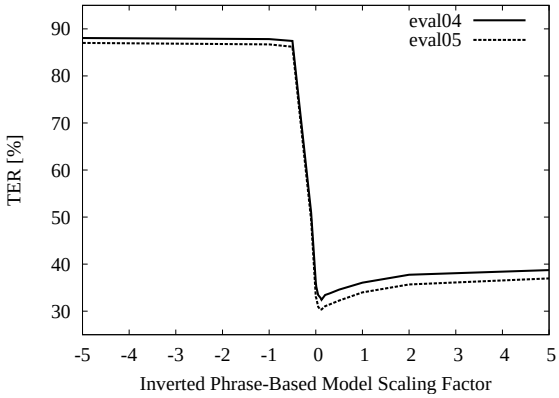

Figure 9.2. Effect of the inverted phrase-based model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

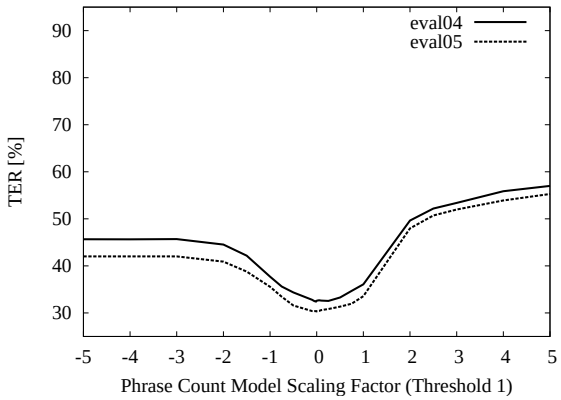

Figure 9.3. Effect of the phrase count model scaling factor on the TER for the Arabic-English BTEC task (threshold 1, IWSLT 2004 and 2005 evaluation sets).

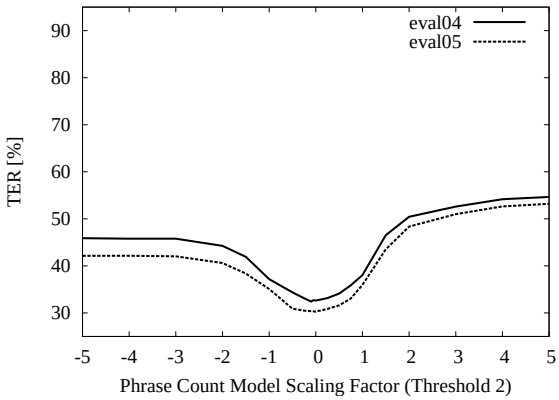

Figure 9.4. Effect of the phrase count model scaling factor on the TER for the Arabic-English BTEC task (threshold 2, IWSLT 2004 and 2005 evaluation sets).

order to support this claim. The corpus statistics and the translation scores for the Chinese-English GALE newswire task are given in Table 9.18. In comparison to the corresponding Arabic-English GALE task, the vocabulary size for the source language is smaller by a factor of two but at the same time, the Chinese-English language pair is much harder to translate as the Bleu and TER score demonstrate.

In Figures 9.11 to 9.14, we separated the effect of the number of lexical and reordering hypotheses on the translation quality. The plots demonstrate that already a small number of lexical alternatives is sufficient to achieve good translation quality. For each curve, we limited the number of reordering hypotheses and varied the maximum number of lexical hypotheses per reordering hypothesis. Thus, along the x-axis we increased the search space by allowing for more lexical choice, whereas from curve to curve we allowed for more reordering. The overall

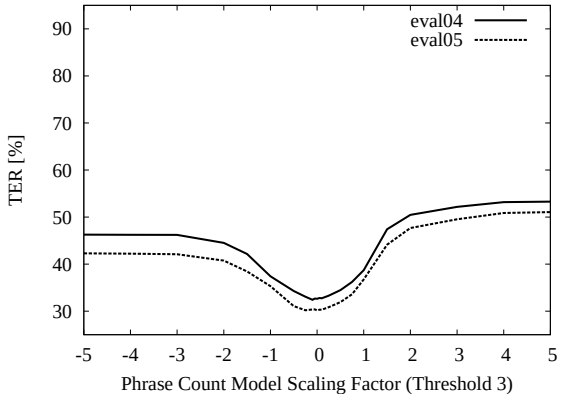

Figure 9.5. Effect of the phrase count model scaling factor on the TER for the Arabic-English BTEC task (threshold 3, IWSLT 2004 and 2005 evaluation sets).

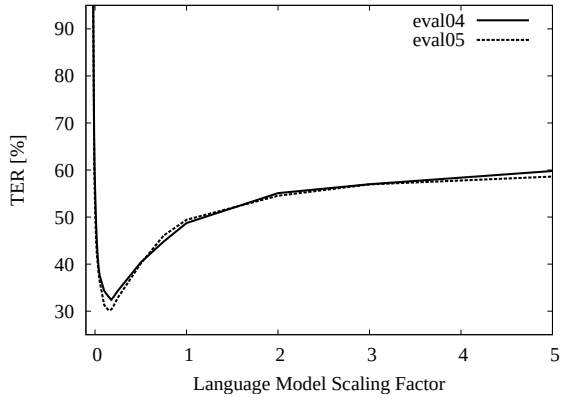

Figure 9.6. Effect of the language model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

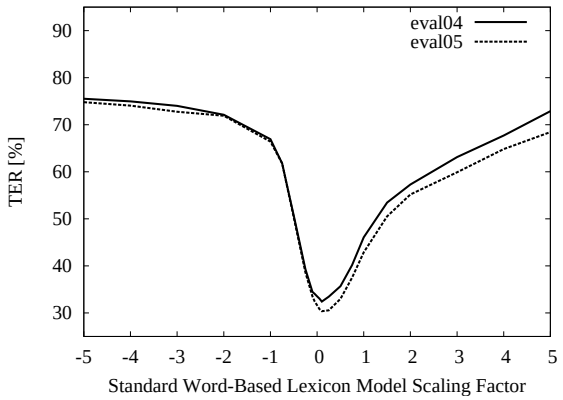

Figure 9.7. Effect of the standard word-based lexicon model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

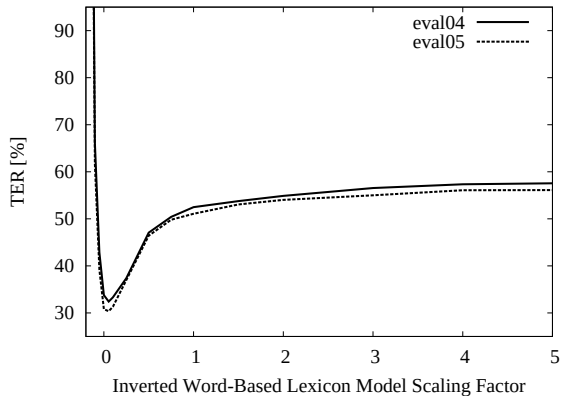

Figure 9.8. Effect of the inverted word-based lexicon model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

search space is limited by the product of the two numbers, i.e. we varied the beam size from 1 to 64K. All figures point out that there is no benefit from increasing the number of lexical choices beyond 16 candidates per reordering hypothesis. If we look at the maximum number of reordering choices, we see again that reordering seems to be only a minor problem for the Arabic-English language pair. In combination with the implicit within-phrase word reorderings, just 4 reordering alternatives times 16 lexical hypotheses for each of them are sufficient to exhaust the search space. By contrast, the maximum number of coverage hypotheses has a much bigger effect on the Bleu score for the Chinese-English translation direction, see Figure 9.14. There is a considerable improvement by increasing the number of coverage hypotheses up to 64. Furthermore, the improvement achieved by taking more reordering alternatives into account exceeds the improvement due to more lexical choices. Two conclusions are important: first,

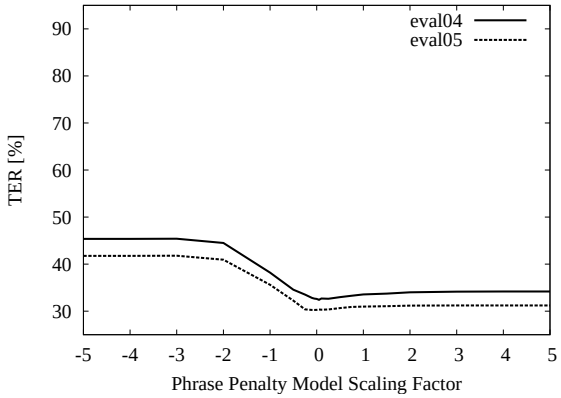

Figure 9.9. Effect of the phrase penalty model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

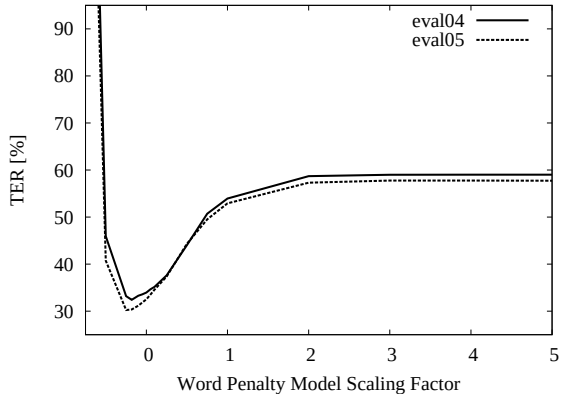

Figure 9.10. Effect of the word penalty model scaling factor on the TER for the Arabic-English BTEC task (IWSLT 2004 and 2005 evaluation sets).

Table 9.18. Corpus statistics and translation quality (BLEUr4n4/TER [%]) for the Chinese-English GALE newswire task (GALE 2007 MT development set).

		Chinese	English
Train	Sentence pairs	8.7 M	
	Running words	230 M	248 M
	Vocabulary size	242 K	439 K
Test NW	Sentences	554	
	Words	19 K	21 K
	Bleu	18.66	
	TER	68.29	

we showed that reordering is only a moderate problem in Arabic-English SMT, at least in terms of the automatic evaluation measures, and second, our pruning statement about the improved translation quality due to a separation of lexical and reordering hypotheses holds. The numbers obtained for the Chinese-English GALE task make clear that it is important to focus on alternative reorderings.

We carried out experiments to analyze the effect of the domain adaptation for the applied language models which are summarized in Table 9.19. For both the newswire and the web text evaluation sets, the domain adapted LMs resulted in an 1% absolute improvement for the Bleu and TER numbers. Thus, the perplexity reductions reported in Section 6.2 also make a difference in translation quality. Though, the improvements for the NW task are due to the additional LM training material whereas for the WT task the adaptation to the specific structure of the web texts especially payed off. This is comprehensible because the bilingual MT training corpora already consist of newswire data for the most part; consequently, it does not make sense to further constrain the training data and overly tune the LMs. But as the additional LM data used for the DMIX-GS* models has been gathered mainly on news data as well, these type of model perfectly fits the NW documents. On the other hand, the DMIX-GS

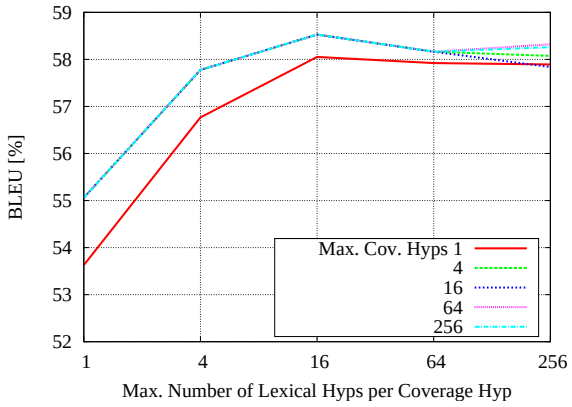

Figure 9.11. Effect of the number of lexical and coverage hypotheses on the Bleu score for the Arabic-English BTEC task (IWSLT 2004 evaluation set).

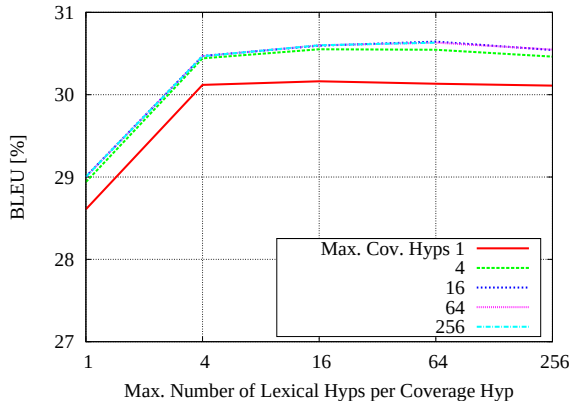

Figure 9.12. Effect of the number of lexical and coverage hypotheses on the Bleu score for the Arabic-English GALE newswire task (GALE 2007 MT development set).

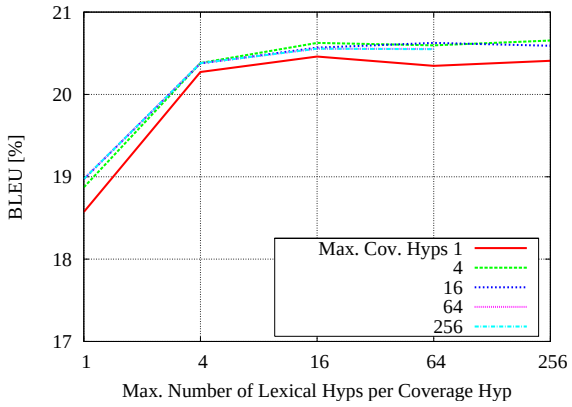

Figure 9.13. Effect of the number of lexical and coverage hypotheses on the Bleu score for the Arabic-English GALE web text task (GALE 2007 MT development set).

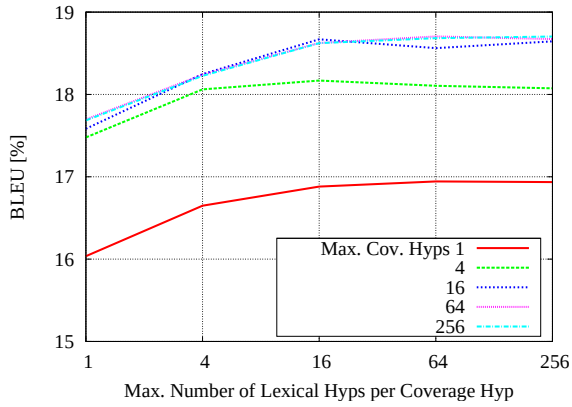

Figure 9.14. Effect of the number of lexical and coverage hypotheses on the Bleu score for the Chinese-English GALE newswire task (GALE 2007 MT development set).

model alleviates the discrepancy between the nature of the WT test sets and of the MT training data. The SMT system is thereby able to generate translation hypotheses that correspond to the structure of the web texts. The additional, mainly NW style LM data does not help to further enhance the translation quality for the same reason. On the development sets, the effect of the domain adapted LMs is less because these systems are highly tuned for the baseline model.

In Table 9.20 and 9.21, we show some translation examples for the Arabic-English GALE tasks. These sentences, on the one hand, illustrate the specific NW and WT tasks, and on the other hand compare monotone vs. non-monotone search (Table 9.20) and exemplify the effect of domain adaptation for the applied LMs (Table 9.21). In the case of monotone search, the single word translations are often correct but the word order is only correct for the non-monotone search (although reordering seems to be only a moderate problem w.r.t. the automatic trans-

Table 9.19. Effect of different language models on the translation quality (BLEUr4n4/TER [%]) for the Arabic-English GALE task (GALE 2007 MT development and GALE 2006 MT evaluation sets).

		Dev		Eval	
Task	Model	BLEU	TER	BLEU	TER
NW	base	31.3	50.5	29.2	53.5
	dmix-gs	30.8	51.0	28.4	54.1
	dmix-gs*	31.7	49.6	30.0	52.3
WT	base	21.8	61.6	16.8	65.9
	dmix-gs	22.1	61.8	17.9	65.1
	dmix-gs*	22.1	61.9	17.5	66.1

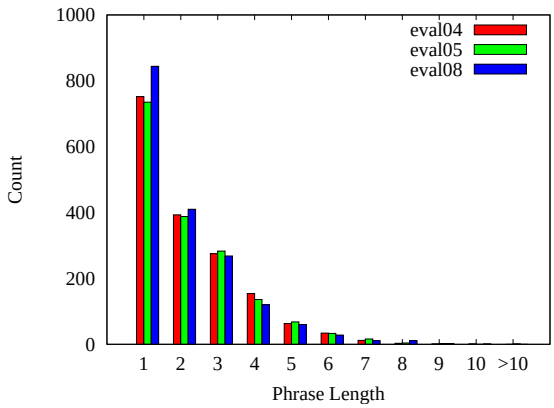

Figure 9.15. Histogram of the phrase lengths used during search for the Arabic-English BTEC task (IWSLT 2004, 2005 and 2008 evaluation sets).

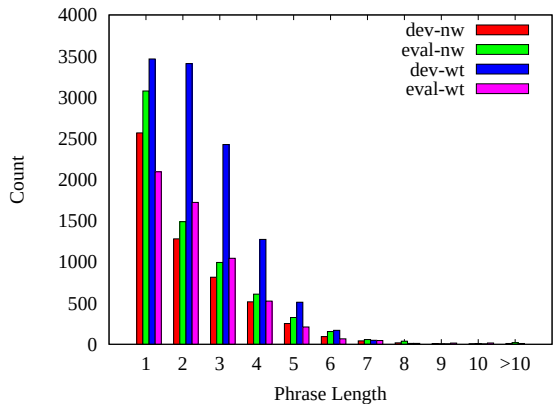

Figure 9.16. Histogram of the phrase lengths used during search for the Arabic-English GALE task (GALE 2007 MT development and GALE 2006 MT evaluation sets).

lation scores). The WT examples demonstrate that the WT task is significantly harder to translate and that the dmix-gs model definitely helps to increase the n-gram coverage.

Next, we comment on the statistics of the actually used phrases to generate the translation hypotheses. In all the above-mentioned experiments, the average length of the phrases that made up the best translations is about two words per phrase, both for the target and source phrases. In addition, we collected the target phrase length counts and plotted the corresponding histograms for the various test sets. As can be seen in Figures 9.15 and 9.16, the maximum phrase count is always achieved for the single-word phrases but if we pool the two to four word phrases, it becomes obvious that these are the phrases that primarily build the translation hypotheses. Thus, not only the translation scores but also these histograms emphasize the advantage of the phrase-based approach over single-word based translation. Furthermore, the plots suggest that it does not make sense to account for phrases consisting of more than ten words.

Before we move on to the reranking experiments, we want to ensure that the word graphs and n-best lists applied at several points of this thesis are of high quality and contain translation

Table 9.20. Translation examples for the Arabic-English GALE newswire task: monotone vs. non-monotone search (GALE 2007 MT development set, lowercase MT hypotheses/references).

MON	*and said if i was not able to protect the ministry employees which led by.*
NON-MON	*he said, if i was not able to protect the personnel of the ministry, which i head.*
REF	*he said, if i am not able to protect the employees of the ministry which i head.*
MON	*news agency reported russian plane that egyptian president hosni mubarak, landed at the vnukovo airport and 2 in the russian capital.*
NON-MON	*news agency reported that the egyptian president's plane landed at the vnukovo airport and 2 in the russian capital.*
REF	*novosti news agency stated that the egyptian president's plane landed in vnukovo - 2 airport in the russian capital.*
MON	*at the end of the election campaign that dominated by debate over the war in iraq and hopes for the opposition democratic party in obtaining 15 additional seats in the parliament that will allow getting a majority for the first time since 1994.*
NON-MON	*at the end of the election campaign that were dominated by the controversy over the war in iraq, the opposition democratic party hopes to gain 15 additional seats in the house of representatives that will allow getting a majority for the first time since 1994.*
REF	*at the end of the election campaign that was dominated by the controversy over the war in iraq, the opposition democratic party hopes to get 15 additional seats in the house of representatives which will enable it to have a majority there for the first time since 1994.*

candidates that are significantly better than the single-best hypotheses. Empirically, word graphs and n-best lists should have an appropriate size such that the oracle error rate, i.e. the error rate of the best hypothesis with respect to an error measure is approximately half the baseline error rate of the system. We proceeded as presented in Section 4.5 to generate the word graphs and to extract the n-best translation candidates and plotted the oracle Bleu score against the word graph density and the n-best list size. Figure 9.17 shows these curves for the Arabic-English BTEC task, and Figure 9.18 depicts the effect of the word graph density and n-best list size for the GALE task (NW). The word graph density was thereby computed as the number of edges in the graph divided by the number of words in the source language input. Note that the word graph density can be less than one since the edges are labeled with phrases. To compute the oracle Bleu score, we selected for each reference sentence the hypothesis which results in the best Bleu score from the word graph or n-best list, respectively, taking into account that this is indeed a far too optimistic upper bound. For both tasks, the oracle Bleu score grows logarithmic with the word graph density and the Bleu error rate, i.e. $1-$ Bleu score, could be successfully reduced to reach half the baseline error rate (BTEC: Bleu

Table 9.21. Translation examples for the Arabic-English GALE web text task: standard vs. domain adapted genre-specific language models (GALE 2007 MT development set, lowercase MT hypotheses/references).

BASE	*and why they refused to pay extortion money the citizen to informers and said to them : do not*
DMIX-GS	*and the reason is the rejection of the citizen pay royalty to informers and said to them : do not*
REF	*the reason is the refusal by the citizen to pay a bribe to the informants. he said "no".*
BASE	*he added that there are many say that henry may was seen praying in a mosque in london and that it was continuing to their religious obligations*
DMIX-GS	*he added that there are many people confirm that henry was seen in the islamic mosque in london and that he will continue to their religious obligations*
REF	*there are many who confirm that henry was seen praying at the islamic mosque in london, and that he practices his religious duties regularly.*
BASE	*because there is reason look for all new for women to enable them to show that all of the people, is proceeding at noon and the body is that of girls in the street and part of her abdomen naked*
DMIX-GS	*because the devil of reason searches for every new for women to be able to demonstrate every some of the people, noon and the body, which is that the girl is moving in the street and part of her abdomen naked*
REF	*because the devil of reason seeks everything new for the woman so she can display everything she's got to the world, now the half-body top has emerged, which is when the girl walks in the street with a bare stomach.*

ER 40% → 17%, GALE: Bleu ER 69% → 36%). For the n-best lists, we could reach the designated reduction of the Bleu error rate only for the BTEC task, though. The GALE n-best translation candidates allow for just a moderate increase of the Bleu score. Furthermore, there seems to be no benefit from exploring more than 10k hypotheses. Thus, the potential gain due to the application of further rescoring models is limited from the first. Here, future work has to elaborate more clever strategies to extract n-best candidate translations from the word graph that bear more potential for improving the translation quality.

To conclude this subsection, we exemplarily show how the presented methods contributed to the overall system improvements for the IWSLT 2008 Arabic-English BTEC task. The results given in Table 9.22 were scored via the official automatic evaluation server for IWSLT 2008, i.e. they are case-sensitive. Here, we report the evaluation measures that were used for the official campaign, thus the TER scores are missing. The Baseline system used the default features of our decoder and was tuned to maximize an equally combined Bleu and TER score. In past evaluations, it turned out that this combination corresponds well to subjective scores obtained by human MT evaluators. When investigating the effect of the different heuristics for alignment symmetrization, we found out that a concatenation of the individual alignments improves translation quality. As always, additional LM data enhances the system as long as

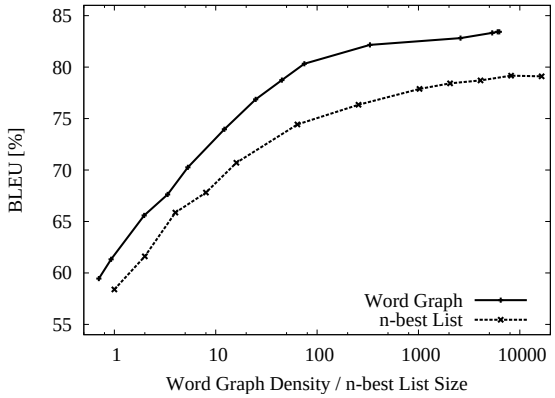

Figure 9.17. Effect of the word graph density and n-best list size on the oracle Bleu score for the Arabic-English BTEC task (IWSLT 2004 evaluation set).

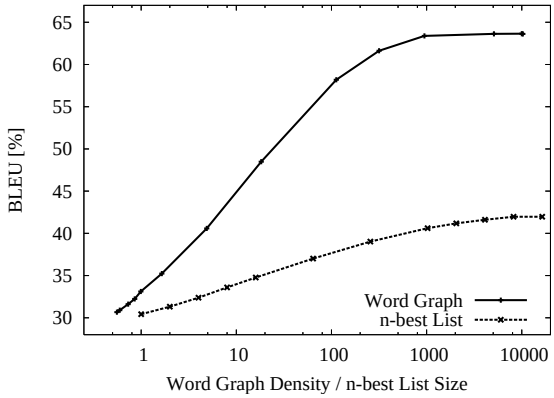

Figure 9.18. Effect of the word graph density and n-best list size on the oracle Bleu score for the Arabic-English GALE newswire task (GALE 2007 MT development set).

Table 9.22. Overview of system improvements for the IWSLT 2008 Arabic-English BTEC task (case-sensitive evaluation, scored via the official automatic evaluation server for IWSLT 2008). The results are in percentage except for the NIST score.

System	METEOR	BLEU	NIST	WER	PER	GTM
Baseline	65.8	51.5	8.57	38.3	34.3	75.8
+ Concat. of Alignm.	66.2	52.4	8.65	38.4	34.5	76.2
+ Add. LM Data (HIT corp)	66.5	53.4	8.73	37.4	33.5	76.7
+ Distortion Model	66.8	53.8	8.91	37.1	33.1	77.0

the data fits the task. To ensure the adequacy of the LM data, we chose the added sentences from the HIT corpus depending on the fraction of words that are already contained in the word-based translation lexicon. At last, there is a small but consistent improvement due to the distortion model in combination with the presented pruning strategies for beam search.

9.1.8 Reranking Experiments

To investigate the value of syntactically motivated feature functions within a SMT reranking concept, we let our decoder generate word graphs containing the most likely translation hypotheses during the search process. Reranking experiments were conducted for the multilingual IWSLT 2005 Supplied Data Track, i.e. we translated the BTEC test sets for the Chinese-English, Arabic-English, and Japanese-English language pairs. Out of these compact representations, we extracted n-best lists as described by [Zens & Ney 05]. Subsequently, these n-best lists served as a starting point for our experiments; the rescoring methods presented in Section 5.2 produced scores that were used as additional features for the n-best lists.

The use of n-best lists in machine translation has several advantages. It alleviates the effects

Table 9.23. Effect of successively adding syntactic features to the Chinese-English n-best list for the BTEC task (C-Star 2003 development set and IWSLT 2004 evaluation set). The results are in percentage except for the NIST score.

C-Star 2003	NIST	BLEU	WER	PER
Baseline	8.17	46.2	48.6	41.4
with supertagging/LDA	8.29	46.5	48.4	41.0
with link grammar	8.43	45.6	47.9	41.1
with supertagging/LDA + link grammar	8.22	47.5	47.7	40.8
with ME chunker	8.65	47.3	47.4	40.4
with all models	8.42	47.0	47.4	40.5
IWSLT 2004	NIST	BLEU	WER	PER
Baseline	8.67	45.5	49.1	39.8
with supertagging/LDA	8.68	45.4	49.8	40.3
with link grammar	8.81	45.0	49.0	40.2
with supertagging/LDA+link grammar	8.56	46.0	49.1	40.6
with ME chunker	9.00	44.6	49.3	40.6
with all models	8.89	46.2	48.1	39.6

of the huge search space which is represented in word graphs by using a compact excerpt of the n best hypotheses generated by the system. Especially for limited domain tasks, the size of the n-best list can be rather small but still yield good oracle error rates. Empirically, n-best lists should have an appropriate size such that the oracle error rate, i.e. the error rate of the best hypothesis with respect to an error measure (such as WER or PER) is approximately half the baseline error rate of the system. N-best lists are suitable for easily applying several rescoring techniques since the hypotheses are already fully generated. In comparison, word graph rescoring techniques need specialized tools which can traverse the graph accordingly. Since a node within a word graph allows for many histories, one can only apply local rescoring techniques, whereas for n-best lists, techniques can be used that consider properties of the whole sentence.

For the Chinese-English and Arabic-English task, we set the n-best list size to $n = 1500$. For Japanese-English, $n = 1000$ produced oracle error rates that are deemed to be sufficiently low, namely 17.7% and 14.8% for WER and PER, respectively. The single-best output for Japanese-English has a word error rate of 33.3% and position-independent word error rate of 25.9%.

For the experiments, we added additional features to the initial models of our decoder that have shown to be particularly useful in the past, such as IBM model 1 score, a clustered language model score and a word penalty that prevents the hypotheses to become too short. A detailed definition of these additional features was given in [Zens & Bender$^+$ 05]. Thus, the baseline we started with is already a very strong one. The log-linear interpolation weights λ_m of our decoder (cf. Equation 1.13) were directly optimized using the Downhill-Simplex method from the Numerical Recipes [Press & Teukolsky$^+$ 02] on a linear combination of WER (word error rate), PER (position-independent word error rate), NIST and Bleu score.

In Table 9.23, we show the effect of adding the presented features successively to the baseline.

Table 9.24. Translation examples for the Chinese-English BTEC task (IWSLT 2004 evaluation set): baseline system (BASE) vs. rescored hypotheses (RESC) vs. reference translation (REFE).

BASE	*Any messages for me?*
RESC	*Do you have any messages for me?*
REFE	*Do you have any messages for me?*
BASE	*She, not yet?*
RESC	*She has not come yet?*
REFE	*Lenny, she has not come in?*
BASE	*How much is it to the?*
RESC	*How much is it to the local call?*
REFE	*How much is it to the city center?*
BASE	*This blot or.*
RESC	*This is not clean.*
REFE	*This still is not clean.*

Separate entries for experiments using supertagging/LDA and link grammars demonstrate that a combination of these syntactic approaches always yields some gain in translation quality (regarding Bleu score). The performance of the maximum-entropy based chunking is comparable. Furthermore, a combination of all three models still yields a small but consistent improvement. Table 9.24 depicts some translation examples for the Chinese-English IWSLT 2004evaluation set. The rescored translations are syntactically coherent, though semantical correctness cannot be guaranteed. On the test data, we achieved an overall improvement of 0.7%, 0.5% and 0.3% in Bleu score for Chinese-English, Japanese-English and Arabic-English, respectively (cf. Tables 9.25 and 9.26).

From the tables, it can be seen that the use of syntactically motivated feature functions within a reranking concept helps to slightly reduce the number of translation errors of the overall translation system. Although the improvement on the IWSLT 2004 set is only moderate, the results are nevertheless comparable or better to the ones from [Och & Gildea$^+$ 04], where, starting from an IBM model 1 baseline, an additional improvement of only 0.4% Bleu was achieved using even more complex methods.

For the maximum entropy based chunking approach, n-grams with $n = 4$ worked best for the chunker that was trained on WSJ data. The domain-specific rescoring model which resulted from the chunker being trained on the BTEC corpora turned out to prefer higher order n-grams, with $n = 6$ or more. This might be an indicator of the domain-specific rescoring model successfully capturing more local context. The training of the other models, i.e. supertagging/LDA and link grammar, was also performed on out-of-domain data. Thus, further improvements should be possible if the models were adapted to the BTEC domain. However, this would require the preparation of an annotated corpus for the supertagger and a specialized link grammar, which are both time-consuming tasks and out of the focus of this work.

The syntactically motivated methods (supertagging/LDA and link grammars) performed similarly to the maximum entropy based chunker which only pays attention to the syntax in a more shallow manner. It seems that both approaches successfully exploit structural properties of language. However, one outlier is the ME chunking performance on the Chinese-English

Table 9.25. Effect of successively adding syntactic features to the Japanese-English n-best list for the BTEC task (C-Star 2003 development set and IWSLT 2004 evaluation set). The results are in percentage except for the NIST score.

C-Star 2003	NIST	BLEU	WER	PER
Baseline	9.09	57.8	31.3	25.0
with supertagging/LDA	9.13	57.8	31.3	24.8
with link grammar	9.46	57.6	31.9	25.3
with supertagging/LDA + link grammar	9.24	58.2	31.0	24.8
with ME chunker	9.31	58.7	30.9	24.4
with all models	9.21	58.9	30.5	24.3
IWSLT 2004	NIST	BLEU	WER	PER
Baseline	9.22	54.7	34.1	25.5
with supertagging/LDA	9.27	54.8	34.2	25.6
with link grammar	9.37	54.9	34.3	25.9
with supertagging/LDA + link grammar	9.30	55.0	34.0	25.6
with ME chunker	9.27	55.0	34.2	25.5
with all models	9.27	55.2	33.9	25.5

Table 9.26. Effect of successively adding syntactic features to the Arabic-English n-best list for the BTEC task (C-Star 2003 development set and IWSLT 2004 evaluation set). The results are in percentage except for the NIST score.

C-Star 2003	NIST	BLEU	WER	PER
Baseline	10.18	64.3	23.9	20.6
with supertagging/LDA	10.13	64.6	23.4	20.1
with link grammar	10.06	64.7	23.4	20.3
with supertagging/LDA + link grammar	10.20	65.0	23.2	20.2
with ME chunker	10.11	65.1	23.0	19.9
with all models	10.23	65.2	23.0	19.9
IWSLT 2004	NIST	BLEU	WER	PER
Baseline	9.75	59.8	26.1	21.9
with supertagging/LDA	9.77	60.5	25.6	21.5
with link grammar	9.74	60.5	25.9	21.7
with supertagging/LDA + link grammar	9.86	60.8	26.0	21.6
with ME chunker	9.71	59.9	25.9	21.8
with all models	9.84	60.1	26.4	21.9

test data, where we observe a lower Bleu but a larger NIST score. An explanation could be the optimization on a linear combination of the error measures, which, for this specific case, favors NIST performance. For the Arabic-English translation direction, the combination of all methods does not seem to generalize well on the test set. In that case, supertagging/LDA and link grammar outperformed the ME chunker; the overall improvement is 1% absolute in terms

of Bleu score.

Summing up, we added syntactically motivated features to a SMT system in a reranking framework. The goal was to analyze whether shallow parsing techniques help in identifying ungrammatical hypotheses. We showed that some improvements are possible by utilizing supertagging, lightweight dependency analysis, a link grammar parser and a maximum entropy based chunk parser. Adding feature scores to the n-best lists and discriminatively training the system on a development set helped to gain up to 0.7% in Bleu score on the test set. Future work could include developing an adapted LTAG for the BTEC domain or incorporating n-gram models into the link grammar concept in order to derive a long-range language model [Lafferty & Sleator$^+$ 92]. However, we feel that the current improvements are not significant enough to justify these efforts. Additionally, there is a need to apply these reranking methods to larger corpora in order to study the effects on longer sentences from more complex domains.

9.1.9 Speech Translation Experiments

In Chapter 6, we covered the specific requirements that come along with the translation of automatically recognized speech. Here, we focus on the translation of Arabic transcripts; experiments were carried out for the current GALE MT from speech tasks, i.e. we translated transcripts of recognized broadcast news (BN) and broadcast conversations (BC). Consequently, we had to cope with the effects of spontaneous speech, misrecognitions from the automatic speech recognizer (ASR), but also worry about unknown words or constructs which are due to the unconstrained domains. In comparison to the GALE system designed for the translation of text input, the speech translation system slightly differs in a number of details, including the data sets. The corpus statistics of the preprocessed data are shown in Table 9.27. We trained our models on approximately seven million sentence pairs, and used the GALE 2007 MT development set to tune the system w.r.t. the Bleu score. The GALE 2006 MT evaluation set was used as a blind test corpus.

For the source language, preprocessing consisted of the steps described in Section 6.3 plus the removal of sentence-internal punctuation. For the target language, we mainly tokenized the corpora, i.e. separated punctuation marks from words. Additionally, we expanded English contradictions like *it's* or *I'm* and removed the case information in order to reduce the vocabulary size and to improve the training. Regular expression were applied to categorize the corresponding numbers, URLs and e-mail addresses. The automatic transcripts for the test sets were obtained using a system combination of three speech recognizers based on SRI's ASR system architecture.[10] For details about the ASR architecture the reader is referred to [Stolcke & Chen$^+$ 06]. The ROVER combination of the individual systems resulted in word error rates of 13.0% on the development set and 23.8% on the test set. More precisely, the combined system achieved error rates of 10.8% (BN) and 16.9% (BC) on the genre-specific parts of the development set.

As we lowercased the training corpus during preprocessing, we needed to restore the correct case information. Therefore, we built a disambiguation language model. True-casing was done in a postprocessing step using the DISAMBIG tool from the SRILM toolkit. Compared to the correct case of the references, true-casing has an error rate of less than 2% on the dev set and about 3% on the test set. Furthermore, we used the ICSI/UW algorithm to

[10] We thank SRI International for providing us with the ASR transcripts.

Table 9.27. Corpus statistics of the Arabic-English GALE MT data as used to build the speech translation system (GALE 2007 MT training data, GALE 2007 MT development sets, GALE 2006 MT evaluation sets, OOV: out-of-vocabulary words), after preprocessing.

			Arabic	English
Train		Sentence pairs	7 M	
		Running words	176 M	181 M
		Vocabulary size	681 K	492 K
		Singletons	304 K	243 K
Dev	BN	Sentences	565	
		Running words	13 424	17 729
		OOV	292	186
	BC	Sentences	315	
		Running words	7 707	10 009
		OOV	590	70
Eval	BN	Sentences	956	
		Running words	13 397	18 204
		OOV	294	87
	BC	Sentences	529	
		Running words	13 033	17 073
		OOV	332	246

automatically segment the ASR transcripts into sentences. Obviously, this results in a sentence segmentation that is different from the segmentation of the reference translations. On document level, we aligned the system translations to the reference translations using our automatic sentence segmentation tool [Matusov & Leusch$^+$ 05] which traces back the decisions of the Levenshtein edit distance algorithm. The translation results for the GALE 2007 development set (Dev-BC/BN) and for the GALE 2006 evaluation set (Eval-BC/BN) are presented in Table 9.28. For the reranking experiments, we used the 10 K best translation candidates.

Applying the domain adapted genre-specific LMs improved the system performance on the dev set as well as on the test set for both domains. The perplexity reductions reported in Section 6.2 hence also make a difference in (speech) translation quality. While we were able to further improve the scores on the dev set by adding additional LM data and reranking the translation candidates, these improvements carried over only to the BN part of the test set. This is comprehensible because the additional LM data used in both passes has been gathered only on news data. Furthermore, the improvements due to the additional rescoring models are comparatively small as the test sets provide just single reference translations and therefore do not allow for a large tolerance in the MT output which can be exploited in the reranking pass.

To show the progress made, we compare the results with the official scores obtained in the 2006 GALE MT evaluation. W.r.t. to the Bleu score, the baseline already outperforms the 2006 system. This is due to the use of additional training data as well as new models in combination with a thorough re-optimization of the entire system. Certainly, the advances of the ASR system account for better translations of the transcripts as well. Domain adapted LMs and the rescoring models further contributed to improve overall translation quality. We achieved improvements of 2.82% Bleu (BC) and 2.50% Bleu (BN) absolute. However, there are

Table 9.28. Translation results (BLEUr4n4/TER [%]) for the Arabic-English GALE speech translation tasks (GALE 2007 MT development sets, GALE 2006 MT evaluation sets); comparison to the RWTH system used in the official GALE 2006 evaluation and overview of current improvements.

	Dev-BC		Dev-BN		Eval-BC		Eval-BN	
System	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
GALE 2006	–	–	–	–	12.3	75.6	16.0	69.3
GALE 2007								
- base	20.0	68.8	23.6	61.1	13.6	83.1	16.9	70.5
- dmix-gs	22.0	64.3	24.9	58.8	15.3	78.4	17.9	68.0
- dmix-gs*	22.8	64.2	25.6	58.4	15.0	79.2	18.2	68.3
GALE 2007								
- rescoring	23.5	62.8	27.0	56.9	15.2	78.4	18.5	67.0

Table 9.29. Translation results (BLEUr4n4/TER [%]) for the Arabic-English GALE speech translation tasks (GALE 2007 MT development sets, GALE 2006 MT evaluation sets) for different input; correct transcripts vs. raw ASR output vs. normalized ASR output.

	Dev-BC		Dev-BN		Eval-BC		Eval-BN	
Input	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
correct transcripts	27.9	53.6	28.8	53.3	22.1	58.7	20.7	62.1
raw ASR output	16.1	67.5	19.2	62.3	12.9	75.2	13.7	69.6
norm. ASR output	22.8	64.2	25.6	58.4	15.0	79.2	18.2	68.3

still shortcomings in our system. Regarding the TER scores, we only improved our system on the BN part of the test set. On the BC part, TER scores even deteriorated. We still have to analyze the translations in more detail but future steps require additional models and LM data that better match the BC domain.

Table 9.29 contains the results for different types of input. Given the correct transcripts, the system would be able to generate translations for the BC and BN sets that perform more or less at the same level. In practice, transcribing BC is a substantially harder task which is also reflected in the ASR error rates (10.8% on BN vs. 16.9% on BC for the dev set). Of course, this affects the MT performance. Translating automatically transcribed input, the Bleu scores drop from 28.79% to 25.64% on BN and from 27.90% to 22.84% on BC. Nonetheless, the numbers demonstrate how important the adjustment steps described in Section 6.3 are. The system performance shows clear deterioration for the translations of the unnormalized ASR output.

To subsume the speech translation experiments, we described our spoken language translation system that was also used in the official GALE translation evaluations. The system uses a two pass approach; in the first pass, we use a dynamic programming beam search decoder to generate n-best translation candidates. In the second pass, these translations are reranked. We proved significant improvements compared to the initial GALE 2006 system, mainly achieved by applying domain adapted genre-specific language models, adding additional data and reranking of the candidate translations. We also demonstrated that our work on adjusting the ASR and

Table 9.30. Translation results for the (raw) Xerox Spanish-English and German-English task (generated by the alignment template approach). The results are in percentage except for the NIST score.

Language Pair	WER	PER	BLEU	NIST
Spanish - English	40.2	34.4	57.2	8.7
English - Spanish	33.4	28.3	62.0	9.5
German - English	67.9	56.6	25.7	6.0
English - German	76.6	68.7	20.7	5.1

SMT vocabularies in a preprocessing step to MT and on predicting punctuation marks that are missing from automatically transcribed speech in the translation process actually pays off in improved translation quality. Future work will focus on how to further adapt to domains that contain very noisy data and data being highly diverse from traditional newswire text, like broadcast conversations or web texts.

9.1.10 Comparison of Search Strategies for Interactive Machine Translation

Chapter 8 dealt with search strategies for interactive (statistical) machine translation systems. Here, we present the experimental results for the comparison of the proposed search strategies. The numbers indicate that the response time can be adequate for real-time responsiveness, although this has been tested only partly yet under real-life conditions.

Before addressing the special aspects of interactive MT, we first outline the experimental setup. As stated in Chapter 8, we conducted these experiments resorting to the alignment template approach as described in [Och & Tillmann$^+$ 99, Och & Ney 04]. After training and optimization of the model scaling factors, the SMT engine [Bender & Zens$^+$ 04] was used to translate the test sets for the four translation directions (Es – En, En – Es, Ge – En, and En – Ge); the translation results in terms of the standard evaluation measures for machine translation (word error rate, position independent word error rate, Bleu score and NIST score) are given in Table 9.30. Using the same parameter settings, a simulation of the interactive mode was carried out. This simulation mode was described by [Och & Zens$^+$ 03]. The system with the same parameter settings was also successfully used by human translators to evaluate it under real-life conditions. Due to the high effort that human evaluations require, only the word-graph based generation strategy was tested. The response time of the system was adequate.

Table 9.31 contains the average extension times and keystroke ratios for the investigated generation strategies, both for the Spanish-English and the German-English corpora, in both translation directions. As can be seen, in nearly all translation directions the best performance in terms of keystroke ratio is achieved when carrying out a new search for every prefix. The only exception is for the English to Spanish direction, where the interactive search with word graphs performed slightly better than the new search for every prefix. This can be due to the rich morphology of the Spanish language, where the correct form of some words can not be generated by the search procedure and thus, the flexibility provided by the use of the Levenshtein distance when searching allows for a better keystroke ratio. This effect is also seen in a smaller scale on the English to German direction. On the other hand, the average extension time for full search is far from being acceptable for real translation tasks.

The values for the system that used the interactive search with word graphs (the one consid-

Table 9.31. Average extension time [ms] and keystroke ratio (KSR) [%] for the investigated generation strategies, both for the Spanish-English and the German-English Xerox (raw) task.

	Translation Direction							
	Es – En		En – Es		Ge – En		En – Ge	
Generation Strategy	time	KSR	time	KSR	time	KSR	time	KSR
interactive	2 489	20.3	3 283	21.8	2 747	38.8	2 661	39.1
combined	130	20.6	332	21.8	112	39.5	105	39.5
interactive with word graphs	17	21.1	13	21.7	25	39.9	28	40.0

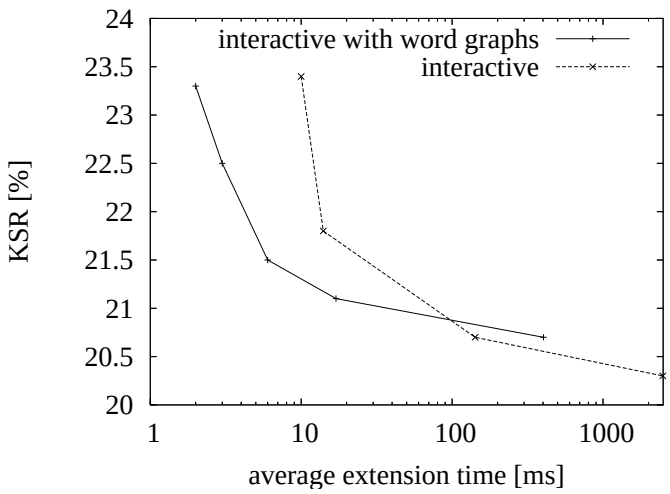

Figure 9.19. Keystroke ratio (KSR) as a function of the average extension time for the interactive and the interactive with word graphs generation strategies for the Spanish-English Xerox (raw) task.

ered in the human evaluation) were obtained for the same parameters as the other systems, but the beam size was further reduced in order to get a better response time. The average extension time could be significantly reduced while the keystroke ration increased only slightly, about 7% relative in the worst case (direction German to English). Figure 9.19 depicts the keystroke ratio as a function of the average extension time (controlled varying the size of the beam). As expected, the best keystroke ratio values are obtained at the expense of a high extension time. Nevertheless, the interactive search with word graphs already achieves adequate results for low extension times, i.e. this strategy fits the tight response time constraints of real-life systems. In addition, Table 9.32 shows the different word graph densities associated with different extension times. The combined generation strategy helps alleviating the performance loss in all the cases and provided a keystroke ratio value between the one of the whole search and the keystroke ratio of the search with word graphs. The average completion time seems to indicate that this strategy could also be used in the interactive environment, but this has still to be tested under

Table 9.32. Keystroke ratio (KSR) [%] and average extension time [ms] for different word graph densities (WGD) for the Spanish-English Xerox (raw) task.

WGD	KSR	time
4	23.3	2
6	22.5	3
57	21.5	6
234	21.1	17
3400	20.7	403

real-life conditions[11].

Table 9.33 gives an example of a sentence from the English-German test corpus that is translated using the interactive generation with word graphs and by applying the pure interactive generation (full search) strategy. The part of the translation that has been accepted by the user is taken as prefix for the search for the next extension. We see that the correct result is obtained much faster with the full search method: only four steps of system-user interaction are necessary instead of seven. The gain is due to the fact that the initial word graph does not contain the word "DocuColor". Hence, the correct extension can not be found directly but has to be produced more or less character by character using the language model heuristic. In contrast, the interactive strategy performs a new search given the prefix "Komponenten des Do" and is then able to produce the correct extension in one step. The number of keystrokes to type the reference decreases from 11 to 6 (with 38 reference characters); KSR decreases from 28.9% to 15.8%. This benefit can also be achieved using the combined generation strategy. Here, a new word graph is computed for the given prefix "Komponenten des Do", resulting in the same one step production of the correct extension.

In summary, the experimental findings for the investigated search strategies for interactive SMT confirm the conclusions drawn at the end of Chapter 8. The most efficient search process first generates a word graph for a given source sentence and subsequently looks for completions of the prefixes within this word graph. The performance of the system degrades slightly but the search is performed in a much more efficient way. In contrast, the approach outputting only the translations compatible with the given prefix needs to perform a new search after each keystroke of the user. Thus, the real-time constraints of an interactive machine translation system do not allow to use this generation algorithm in practice. Furthermore, a combination of both strategies was proposed which improves the translation quality while still complying with the severe constraints of interactive MT.

9.2 Transliteration Tasks

In the second part of this chapter, we focus on Arabic name transliteration. We carried out experiments on two different corpora and investigated different statistical approaches to the transliteration task. Our motivation was to apply only purely data-driven methods that do not

[11] Consider also the subjective impression of a "long" wait only when selecting a new sentence to translate and then nearly instant completions (word-graph based search) against "random" waiting times when typing the translation (combined strategy).

Table 9.33. Comparison of the interactive generation with word graphs and the interactive generation strategies for an example from the English-German Xerox test set; simulated interactive mode.

SOURCE		`Component parts of the DocuColor 12 Printer`
REFERENCE		`Komponenten des DocuColor 12 Druckers`
interactive generation with word graphs	prefix	
	extension	`Komponenten der DocuColor 12`
	prefix	`Komponenten des`
	extension	
	prefix	`Komponenten des D`
	extension	`er DocuColor 12`
	prefix	`Komponenten des Do`
	extension	`cument`
	prefix	`Komponenten des DocuC`
	extension	`olor`
	prefix	`Komponenten des DocuColor 1`
	extension	`2`
	prefix	`Komponenten des DocuColor 12 D`
	extension	`ruckers`
interactive generation	prefix	
	extension	`Komponenten der DocuColor 12`
	prefix	`Komponenten des`
	extension	`Komponenten der DocuColor 12`
	prefix	`Komponenten des D`
	extension	`er DocuColor 12`
	prefix	`Komponenten des Do`
	extension	`cuColor 12 Druckers`

require any additional knowledge but just a set of training name pairs. In doing so, we performed an analysis of several methods and showed how they compare to the current state-of-the-art for (proper) name transliteration. Additionally, we investigated the benefit of the individual approaches when applied within a system combination framework. As for the translation tasks, we first comment on the error measures used to evaluate the transliterations, then briefly sketch the investigated tasks, and finally analyze the experimental findings in detail.

9.2.1 Evaluation Criteria

As for the translation tasks, there exists no standard, commonly agreed on evaluation measure for name transliteration. Moreover, the problem of name transliteration is often studied within the higher-level problem, thus also applying the evaluation measures corresponding to the parent task. Here, we survey the problem of transliteration apart and can therefore use the following simple error criteria:

- character error rate (CER):
  which is the equivalent to the widely used word error rate, e.g. in speech recognition or MT, but defined on the character level, see Subsection 9.1.2, and

Table 9.34. Corpora characteristics for the Arabic names and their English renderings; "10 001 Arabic Names" (LDC2005G02) and "NGA Name Database" (LDC2005G01) corpora.

	Name Pairs			# Chars		Mean Length	
Corpus	train	dev	eval	Arabic	English	Arabic	English
"10 001 Arabic Names"	8 084	1 000	1 000	31	28	4.9	6.5
"NGA Name Database"	58 557	2 000	2 000	48	33	9.0	12.6

- word error rate (WER):
  which corresponds to the common sentence error rate on character level as well.

We believe that the distinction between character and word error rate is advantageous over just the transliteration accuracy or the corresponding F-measure score in order to get an idea of the transliteration capabilities of the individual approaches.

9.2.2 Task Descriptions and Corpus Statistics

We performed experiments on the "10 001 Arabic Names" (LDC2005G02) and on the "NGA Name Database" (LDC2005G01) corpora. These data sets are among the LDC resources exclusively released to GALE participants. The 10k corpus contains Arabic forenames and surnames encoded according to the Standard Arabic Technical Transliteration System (SATTS). The Arabic names are written as they would appear in conventional written Arabic, i.e. they are lacking short vowels or any other diacritics. After filtering, 10 084 name pairs remain which were randomly split to form train (8 084), development (1 000) and evaluation (1 000) sets. The NGA database is built from Arabic script (UTF-8) renderings of common Arabic names. Again after filtering, 62 557 name pairs remain which were split into train (58 557), dev (2 000) and eval (2 000) sets. In Table 9.34, we give some characteristics for the Arabic names and their English renderings which are composed of the conventional 26 alphabetical characters plus some characters to bind the names together. The simple UTF-8 rendering for the NGA database leads to a higher number of individual Arabic source characters. Furthermore, the average length of the names is nearly doubled in comparison to the 10k corpus since the NGA data sets do not contain just forenames and surnames but a choice of common Arabic names, e.g. مەمد الموسَى → Muhammad Al Musa, or سبخپ ەسيَان المنبقه → Sabkhat Hasyan Al.

9.2.3 Comparison of Different Approaches to Arabic Name Transliteration

At first, we present our experiments for the "10 001 Arabic Names" (10k) task. We started using our phrase-based SMT decoder and then investigated and analyzed each of the methods covered in Section 7.2. Recapitulate that only the phrase-based MT (PBT), the grapheme-to-phoneme (G2P) and the deep belief networks (DBN) methods allow for independently building a transliteration system, including training of the character alignment or segmentation, respectively. To be able to also apply the log-linear approaches, i.e. the maximum entropy (ME) models and the conditional random fields (CRF), we utilized the output of the G2P toolkit and took the segmentation as determined by the joint multi-grams. The experimental results in terms of character error rate (CER) and word error rate (WER) are shown in Table 9.35. As can be seen, the methods perform more or less equal except for the DBN approach. Furthermore, they

Table 9.35. Comparison of different transliteration approaches for the "10 001 Arabic Names" (10k) task. PERC denotes the perceptron-trained edit model of [Freitag & Khadivi 07]. The results are in percentage.

	Dev		Eval		Train	
System	CER	WER	CER	WER	CER	WER
PBT	12.9	45.8	13.3	47.2	2.2	7.0
G2P	12.2	43.8	12.1	43.4	1.5	5.0
ME	12.3	45.5	12.4	44.7	0.4	2.2
CRF	12.0	44.2	12.0	43.2	0.1	0.5
DBN	23.5	70.3	23.1	69.1	1.3	6.6
PERC	12.0	45.7	12.2	43.1	–	–

achieve state-of-the-art results as becomes clear when comparing the numbers with the results for the perceptron-trained edit model of [Freitag & Khadivi 07], for instance. Our ME tagger is in principle able to use an arbitrary window length for the context features; in practice, we always have to adjust the trade-off between model size and tagging accuracy. For the 10k task, we chose a window size of up to six positions. However, to better compare the ME and CRF approach, we trained a ME model with exactly the same set of features that are supported by the CRF++ toolkit. Doing so, the performance of the ME system only deteriorated marginally by just 0.1% in CER. The CRF and G2P models always generate the best transliterations. In the last two columns, the error rates are given that were obtained for testing the models on the training data. By these experiments we could prove that all approaches capture the correspondences between the characters in the Arabic and English names. Moreover, the log-linear methods are especially good at this task which is consistent with the theory. When not discarding any features at all and not smoothing the parameters, the ME approach reaches a CER of less than 0.01% on the training data. Nevertheless, the systems differ in their generalization abilities on unseen test data.

By further analysis, we figured out that the segmentation seems to be the major problem. When going from single-best to n-best transliterations and thereby exploiting multiple segmentations, the oracle rates showed significant room for improvements; in fact, the oracle CER could be reduced to reach less than 5% when computed for 10-best lists. In a first experiment, we therefore wanted to check whether our standard translation rescoring models, see Subsection 5.2.1, are also valuable for transliteration. Expectedly, this experiment turned out to not be successful. Only the DBN transliteration candidates could be further improved applying the rescoring models; there, the CER could be reduced from 23.5% to 21.3% for the dev set, and from 23.1% to 20.1% for the eval set, still falling behind the other systems by far. For all the other approaches, just very small rescoring improvements on the dev set could be achieved which did not really carry over to the eval sets and thus proved to be mainly overfitting effects.

In another experiment, we investigated the potential benefit of the individual systems' transliterations when fed into a system combination framework. Thereto, we performed light-weighted system combination using the Recognizer Output Voting Error Reduction (ROVER) approach which is known to work well in speech recognition [Fiscus 97]. We wanted to see if the ROVER approach also yields improvements when combining the transliteration candidates. Moreover, we were motivated by the fact that, although being clearly outperformed by

Table 9.36. Comparison of transliteration results for different ROVER combinations on the "10 001 Arabic Names" (10k) task. The results are in percentage.

	Dev		Eval	
System	CER	WER	CER	WER
Baseline				
CRF	12.0	44.2	12.0	43.2
Rover				
5-way equal weights	11.7	42.9	11.9	42.8
4-way setting w/o DBN	11.9	43.3	11.9	42.7
3-way setting w/o DBN	11.9	43.2	11.8	42.6
best setting w/ DBN	11.0	42.8	11.0	43.3
+ tuned weights	10.9	42.7	10.9	43.0

the other approaches, the DBN approach differs decisively from the other statistical approaches we applied to the transliteration task, and thus investigated the potential benefit of the diverse nature of the DBN transliterations. To get an idea of the theoretically achievable performance gain, we calculated the oracle character error rate on the dev set which equals 6.15% CER, thus providing room for improvements. A comparison of the transliteration scores obtained for the single-best CRF baseline and for different ROVER combinations is shown in Table 9.36.

If we look at combinations of systems without the DBN approach, we observe only marginal improvements of around 0.1 to 0.2% CER. Interestingly, a combination of all four models (PBT, G2P, ME, CRF) works as good as individual 3-way combinations (the same 11.9% CER on the dev set were obtained). This can be interpreted as a potential "similarity" of the approaches. Adding e.g. ME to a combination of PBT, G2P and CRF does not improve results because the transliteration hypotheses are too similar. If we simply put together all five systems including DBN with equal weights, we have a similar trend. Since all systems are equally weighted and at least three of the systems are similar in individual performance (G2P, ME, CRF have all around 12% CER on the tested data sets), the DBN approach does not get a large impact on the overall performance.

If we drop similar systems and tune for 3-way combinations, we observe a large reduction in CER if DBN comes into play. Compared to the best individual system (CRF) of 12% CER, we now arrived at a CER of 11.0% for a combination of PBT, CRF and DBN which is significantly better than each of the individual methods. Our interpretation of this is that the DBN system has different hypotheses compared to all other systems and that the hypotheses from the other systems are too similar to be apt for combination. We got another, rather small improvement of 0.1% CER by tuning the system weights using Powell's method. So, although DBN is much worse than the other approaches, it obviously helps in the system combination; the overall results could be improved by 1% absolute. Using the rescored variant of the DBN transliterations, performance was equal to the one obtained for the DBN baseline. Another argument for the "similarity" of the approaches is that one of the system weights of the similar approaches (G2P, ME, CRF) always gets set to zero when tuning the weights for minimum CER on the dev set.

We repeated the same experiments for the "NGA Name Database" (NGA) task and displayed

Table 9.37. Comparison of different transliteration approaches for the "NGA Name Database" (NGA) task. PERC denotes the perceptron-trained edit model of [Freitag & Khadivi 07]. The results are in percentage.

	Dev		Eval	
System	CER	WER	CER	WER
PBT	5.9	39.4	5.7	37.7
G2P	5.8	39.2	5.7	38.3
ME	6.1	41.4	6.1	41.2
CRF	5.8	38.9	5.6	38.1
DBN	25.3	90.0	24.5	90.2
PERC	6.4	42.7	6.1	41.8

the obtained transliteration results (CER/WER) in Table 9.37. As can be drawn from the table, the main findings remain the same. At first, we were able to achieve state-of-the-art performance as the comparison with [Freitag & Khadivi 07] proves. Furthermore, the CRF and G2P methods again generated the best transliterations. However, our phrase-based SMT system seems to best scale with larger training data as the performance gain w.r.t. the scores achieved for the 10k task demonstrate. The ME and PERC systems could be certainly further optimized somewhat but this would not change the general observations.

Unlike the 10k task, the DBN transliterations do not fall behind the other approaches here, but they are completely out of the useful range. Even after approximately 500 training iterations and 30 days of computing time, the model was still far from converging to a feasible setting and the intermediate system obtained just the poor scores. One might argue that the higher average length of the names in the NGA data sets pose a particular problem for the DBN system, but future work definitely has to elaborate more efficient ways of training the DBN models. As the ROVER experiments for the 10k task show, the poor performing DBN transliteration candidates significantly shorten the performance gain expected due to system combination, at the same time. And, in fact, we were only able to slightly increase the quality of the transliterations by performing system combination; the CER scores could be reduced from 5.8% (CRF) to 5.5% on the dev set and from 5.6% to 5.5% on the eval set for a tuned combination of the CRF, PBT, and DBT system outputs.

Concluding the section on the transliteration experiments, we presented the results obtained for several purely data-driven approaches to the task of Arabic name transliteration. We carried out experiments on two different corpora and showed that our methods achieve state-of-the-art performance by a comparison with the results of [Freitag & Khadivi 07]. While we were able to significantly improve the overall results by 1% absolute due to adding DBN-based transliterations within a ROVER system combination framework for the 10k task, we could not produce similar results for the NGA task. Here, more efficient ways of training the DBN models are required. Furthermore, our standard rescoring models were not capable to utilize the potential improvements contained in the n-best candidate transliterations. Hence, future work should consider rescoring models that fit the transliteration task. Since, we so far only consider the single-best output of each system, ROVER is just a simple majority voting on character level after a Levenshtein alignment of all system outputs has been performed. In this respect, an investigation of n-best list system combination as was proposed by [Stolcke & Konig$^+$ 97] for

speech recognition would be certainly worth the effort. Finally, the log-linear approaches are still lacking the modeling of the segmentation. A reformulation of the ME and CRF models including the character level alignment should be done and incorporated into our implementations.

9.3 Summary

In this chapter, we presented the main experimental findings to stress and exemplify the core statements of this thesis. We already subsumed the experiments and pointed out problems that should be faced in the future throughout this chapter. We analyzed both passes of our phrase-based decoder in detail and showed how the proper choice of the preprocessing approach helps to increase the translation quality. We covered the special requirements of unstructured and automatically recognized input and multi-domain tasks. Furthermore, we addressed the task of transliterating proper names, and investigated search strategies for interactive machine translation, respectively. In the next chapter, we summarize the main achievements of this thesis. Finally, we conclude and outline directions for future work in Chapter 11.

Chapter 10

Scientific Contributions

The aim of this work was to build a robust SMT system for multi-domain translation tasks. We focused on the Arabic-English language pair and, in this context, analyzed the search problem for phrase-based SMT in detail. Striving for our goals, we also showed how the proper choice of the preprocessing approach helps to increase the translation quality. In addition, we addressed the task of improving the translation quality by means of syntactically motivated feature functions within a reranking concept. Finally, we commented on the task of transliterating proper names, and on search strategies for interactive machine translation. Throughout this thesis, we gave concluding remarks and pointed out problems that should be faced in the future. In this chapter, we summarize the most important contributions:

- **Detailed analysis of the search problem:**
  [Zens 08] concluded that word reordering is a substantial problem in Chinese-English machine translation, and that it is important to focus the search on alternative coverage hypotheses, i.e. alternative reorderings. In Arabic-English SMT, however, reordering is only a moderate problem at least in terms of the automatic evaluation measures. Still, reordering can make the difference for subjective analysis of the translation hypotheses. Referring to pruning and improved translation quality due to a separation of lexical and reordering hypotheses, [Zens 08]'s statement holds but the improvements by non-monotone translation are rather small. We also investigated not only limiting the maximum number of hypotheses but also defining a minimum number of lexical and coverage hypotheses kept in the beam at all times. Unfortunately, these minimum beam settings did not make a difference in practice.

  Further experiments showed that, for Arabic-English SMT, the phrase models, the word penalty, and the proper choice of the word lexicon share a particular impact on the translation quality. By comparison, the phrase penalty and the distortion penalty model only have a minor influence on the translation hypotheses. The effect of the phrase count models on the translation quality is negligible; therefore, they should not be used in practice to speed up the training and optimization processes, and to reduce the risk of suboptimal parameter settings.

  Finally, the refined heuristic for symmetrization of the word alignments usually improves the translation quality. For small translation tasks such as BTEC, it is helpful to concatenate the different symmetric alignments and to train the SMT system on the duplicated data.

- **Robustness and multi-domains:**
  Obviously, the findings obtained by analysis of the search problem could be listed w.r.t. robustness and multi-domains as well, since all of the experiments were conducted on translation tasks that correspond to the targeted goals of this thesis.

In addition, we addressed open domains primarily by use of domain adaptation to the language models. More precisely, we trained domain-specific LMs for each genre which were applied during search. The experimental results show that domain adaptation to the language models is especially useful for data that are diverse from traditional newswire text; thus, the perplexity reductions reported in Section 6.2 also led to a difference in translation quality. In contrast, traditional newswire translations could be further improved by additional (rescoring) LMs, e.g. count-based models trained on huge corpora.

The task of cleaning and harmonizing noisy input was partly tackled as a preprocessing step to MT. For SLT, we adapted the ASR output to a more conventional SMT task which enabled us to use our decoder as principal translation engine for different speech recognizers. When translating automatically transcribed speech, adjusting the ASR and SMT vocabularies and predicting missing punctuation marks leads to significant improvements in translation quality. Moreover, approaching sentence-end and sentence-internal punctuation marks differently is beneficial for Arabic-English speech translation.

- **Morpho-syntactic Arabic preprocessing:**
  Arabic is a highly inflected language compared to languages like English which have very little morphology. In this work, we analyzed varying word segmentation approaches and normalization strategies, and examined the impact of the applied preprocessing on the translation quality. Out of the investigated methods, the IFSA, MADA-ATB and MTAG methods performed a quite rigorous word splitting in comparison to the other MADA variants. This can be seen at the relatively high number of running words but rather small vocabulary size and number of singletons at the same time. In terms of the quality of the translations, the choice of the preprocessing strategy depends on the translation domain and on the structure of the input data. Rather clean and limited data sets allow for more syntactical and morphological methods while more ambitious, real-life data should be tackled using more robust techniques. For instance, the MTAG approach which yielded the best translations for the clean and small BTEC data clearly fell behind the other approaches for the GALE tasks. The number of OOVs was considerably higher and the translation quality was inferior to the one obtained by the best methods. Here, the MADA-ATB method is the first choice for the text domains, whereas the IFSA approach achieved the best translations for the speech input. Apparently, the morphological POS tagger failed when being confronted with diverse, noisy and unstructured input.

- **Incorporation of structural properties:**
  Given the fact that statistically driven MT systems still make a lot of errors, we showed how to set up a reranking framework to enhance the MT quality by exploiting multiple translation candidates using additional rescoring models and discriminatively training the system on a development set. Those additional models are typically hard to apply during the search, either because of the high computational demands or because they require that the translation hypothesis is fully generated. We started using standard models that have shown to be particularly useful for rescoring of MT hypotheses in the past and then added syntactically motivated features; supertagging/lightweight dependency analysis (LDA), link grammars and maximum entropy based chunking. The goal was to analyze whether shallow parsing techniques help in identifying ungrammatical hypotheses. The experimental findings show that the use of syntactically motivated feature functions within a reranking concept helps to slightly reduce the number of translation errors of

the overall translation system. Although the improvement was only moderate, the results are nevertheless comparable or better than the ones from [Och & Gildea$^+$ 04].

- **Transliteration of proper names:**
  Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Often, out-of-vocabulary (OOV) terms are the names of entities and the intention was to preserve the names by transliteration. In contrast to the (phonetic) matching of source language names against a large list of candidate transliterations and to many recent publications, we focused on purely data-driven approaches that do not require any additional knowledge but just a set of training name pairs. We showed how these approaches that have been successfully applied to NLP or general machine learning tasks, can be reformulated in order to be used for machine transliteration. By comparison with the results for the perceptron-trained edit model of [Freitag & Khadivi 07], we proved that the systems achieved state-of-the-art performance. Moreover, the transliterations could be significantly improved by adding DBN-based transliterations within a ROVER system combination framework, at least for one specific transliteration task.

- **Search strategies for interactive machine translation:**
  Here, we dealt with search strategies for interactive (statistical) MT systems. Clearly, the best approach would be to start a new search for every given prefix. However, in these kind of systems, response time is a crucial factor for a human translator as delays higher than a fraction of a second are not acceptable. With today's algorithms and available computing power these time restrictions can not be met when doing a full search for each prefix. Thus, we reviewed an efficient generation process [Och & Zens$^+$ 03] which first generates a word graph for a given source sentence and subsequently looks for completions of the prefixes within this word graph. The performance of the system degraded slightly but the search was performed in a much more efficient way. In the end, a combination of both strategies was proposed which improved the translation quality while still keeping an eye on the severe constraints of interactive MT. First off-line experiments showed that the response time can be adequate for real-time responsiveness, although this has not been tested yet under real-life conditions.

Chapter 11

Conclusions

At the beginning of this work, we set ourselves the target to extend the state-of-the-art in phrase-based SMT in order to build a robust SMT system for multi-domain translation tasks. Thereby, the main focus was on Arabic-English translation tasks. In conclusion, we attained our goals as can be derived from the fact that the presented system was successfully used in various evaluation campaigns, including speech translation tasks and multi-domain input, e.g.:

- the NIST Open Machine Translation Evaluation 2006 (MT06) [Bender & Isbihani$^+$ 06],
- the NIST Machine Translation Evaluation for GALE 2007 Phase 2 Evaluation [Bender & Matusov$^+$ 07], or
- the series of IWSLT evaluation campaigns [Zens & Bender$^+$ 05, Mauser & Zens$^+$ 06, Mauser & Vilar$^+$ 07, Vilar & Stein$^+$ 08],

and has thus been proven to achieve and outperform the state-of-the-art for the intended tasks. For all evaluations, our system was ranked among the top submissions. We compared the obtained translation results with those obtained by other research groups working on MT in Subsection 9.1.4.

The scientific contributions presented in this work enabled us to set up a translation system for Arabic-English translation tasks that,

- on the one hand, achieves state-of-the-art performance,
- is at the same time robust regarding unstructured and automatically recognized input, and
- pays attention to the special requirements of multi-domain tasks as well.

11.1 Outlook

As for any data-driven approach, the most obvious way to further improve the presented system is by means of more and better data. Of course, this also poses the problem of automated collection of bilingual training data. Due to the availability of multilingual data, e.g. the UN corpus or the EU bulletin corpus, in combination with clever ways to automatically gather training data from RSS news feeds, e.g. [Fry 05], the amount of training material significantly increased. However, the problem of performing a robust sentence alignment and filtering out bad translation examples persists.

Next, refined models can of course contribute to improvements in translation quality as well. Nevertheless, as the "Google approach" to MT shows, the improvements due to better models usually fall behind the improvements achieved by large-scale upgrades, either in

the employed data or the actual implementation, by far. At present, there are attempts to incorporate syntactical knowledge into the search process. These approaches parse the sentences in one or both of the involved languages and the translations are then performed on tree structures, e.g. [Galley & Graehl$^+$ 06, Chiang 07, DeNeefe & Knight$^+$ 07]. Factored translation models [Koehn & Hoang$^+$ 07] distinguish words at the stem or POS level and are of special interest for scarce resources and/or morphologically rich languages. Another shortcoming of the phrase-based approach is the lack of long-range dependencies. [Hasan & Ganitkevitch$^+$ 08] address this problem by introducing triplet lexicon models. Although improvements are reported for all of these methods, the improvements are merely achieved for specific test sets or subsets of commonly used test sets on closer examination. To our knowledge, there exists no refined model which is advantageous over the phrase-based approach for generic translation tasks as a general rule.

Refined methods yielding a better adaptation to different, potentially unstructured and noisy domains would be beneficial. So far, we only consider the applied language models, but certainly the phrase and lexicon models should be adapted to the specific domain as well. Moreover, we tackle speech translation as a serial coupling of a speech recognizer and a translation system. Yet, from a viewpoint of statistical decision theory, an integrated approach would be desirable [Ney 99]. However, the integrated approach resulted in improved translation scores only for simple speech translation tasks so far. Future work has to elaborate appropriate models for the integrated search. The same holds for the task of machine transliteration which we investigated apart. Here, we should look at an analysis of the entire translation system which applies the transliteration component, similar to the approach of [Hermjakob & Knight$^+$ 08].

Appendix A

Symbols and Acronyms

A.1 Mathematical Symbols

A.1.1 Mathematical Symbols Used for Translation

$\mathbf{f} = f_1^J = f_1, ..., f_j, ..., f_J$	source language sentence
$\mathbf{e} = e_1^I = e_1, ..., e_i, ..., e_I$	target language sentence
$\mathbf{a} = a_1^J = a_1, ..., a_j, ..., a_J$	word alignment
$A = \{(a_j, j) \mid a_j > 0\}$	word alignment matrix
$Pr(\cdot)$	general probability distribution with (nearly) no specific assumptions
$p(\cdot)$	model-based probability distribution
λ	model scaling factor
$h(\cdot)$	component of log-linear model
$\tilde{f}$	source phrase
$\tilde{e}$	target phrase
s_1^K	segmentation into K phrase pairs
b_k	start position of kth source phrase
j_k	end position of kth source phrase
i_k	end position of kth target phrase
$Q(C, \tilde{e}, j)$	score of hypothesis $(C, \tilde{e}, j)$, i.e. the hypothesis with coverage C, LM history $\tilde{e}$ and source sentence position j
$E(j, j')$	translation candidates for source phrase $f_j, \ldots, f_{j'}$
$B(C, \tilde{e}, j)$	back pointer of hypothesis $(C, \tilde{e}, j)$
$A(C, \tilde{e}, j)$	maximizing argument of hypothesis $(C, \tilde{e}, j)$
L_s	maximum source phrase length

$R(C, j)$	rest score estimate
$q_{\mathbf{TM}}(\tilde{e}, j, j')$	weighted translation model score for translating source phrase $f_j, \ldots, f_{j'}$ with target phrase $\tilde{e}$
$q_{\mathbf{TM}}(j, j')$	best translation model score for translating source phrase $f_j, \ldots, f_{j'}$, i.e. $q_{\mathrm{TM}}(j, j') = \max_{\tilde{e}} q_{\mathrm{TM}}(\tilde{e}, j, j')$
$q_{\mathbf{LM}}(\tilde{e}\|\tilde{e}')$	weighted LM score of phrase $\tilde{e}$ given LM history $\tilde{e}'$
$q_{\mathbf{DM}}(j, j')$	weighted distortion score for a jump from source position j to source position j'
N_O	histogram size for observation pruning
N_C	histogram size for coverage pruning per cardinality
τ_C	threshold for coverage pruning per cardinality
N_L	histogram size for lexical pruning per coverage
τ_L	threshold for lexical pruning per coverage
\$	sentence start or sentence end symbol
$g_1^I = g_1, ..., g_i, ..., g_I$	sequence of POS tags (for the target language sentence)
$c_1^I = c_1, ..., c_i, ..., c_I$	sequence of chunk tags (for the target language sentence): the actual text chunks are derived from these chunk tags, cf. [Bender & Macherey$^+$ 03]

A.1.2 Mathematical Symbols Used for Transliteration

$s_1^M = s_1, ..., s_m, ..., s_M$	sequence of characters for source language word
$t_1^N = t_1, ..., t_n, ..., t_N$	sequence of characters for corresponding target language word
$Pr(\cdot)$	general probability distribution with (nearly) no specific assumptions
$p(\cdot)$	model-based probability distribution
λ	model scaling factor
$h(\cdot)$	component of log-linear model
G	the set of graphemes, also referred to as characters, or letters
Φ	the set of phonemes, i.e. phoneme symbols used for phonemic transcription
$\boldsymbol{g} \in G^*$	an orthographic form (sequence of letters)
$\boldsymbol{\varphi} \in \Phi^*$	a pronunciation (phoneme sequence)

$q = (\boldsymbol{g}, \boldsymbol{\varphi}) \in Q \subseteq G^* \times \Phi^*$	a grapheme-phoneme joint multi-gram, or graphone
Q	the inventory of graphones
$\boldsymbol{q} \in Q^*$	s sequence of graphones
$\delta(\cdot, \cdot)$	the Kronecker-function
σ	the free parameter for Gaussian prior smoothing
$S_{1,2,3}$	number of neurons in the source encoders
$T_{1,2,3}$	number of neurons in the target encoders
S_F	binary input vector for DBN
D_F	dimensionality of the binary input vector S_F
$O_{(\cdot)}$	binary output vector of DBN
$w_{(\cdot)}$	weight matrix for DBN
$b_{(\cdot)}$	bias vector for DBN

A.2 Acronyms

ACE Automatic Content Extraction

ACL Association for Computational Linguistics

AE Arabic-English

ASR Automatic Speech Recognition

AT Alignment Template Approach to SMT

ATB Arabic Tree Bank

BAMA Buckwalter Arabic Morphological Analyzer

BLEU BiLingual Evaluation Understudy

BC Broadcast Conversations

BN Broadcast News

BTEC Basic Travel Expression Corpus

CAT Computer-Assisted Translation

CE Chinese-English

CoNLL Computational Natural Language Learning

CRF Conditional Random Field

DARPA Defense Advanced Research Projects Agency

DBN Deep Belief Network

DP Dynamic Programming

EAMT European Association for Machine Translation

EBMT Example-Based Machine Translation

EM Expectation Maximization

FSA Finite State Automaton

FST Finite State Transducer

G2P Grapheme-to-Phoneme

GALE Global Autonomous Language Exploitation

GIS Generalized Iterative Scaling

GTM General Text Matcher

HIT Harbin Institute of Technology

HMM	Hidden Markov Model
ICSI	International Computer Science Institute
IFSA	Improved Finite State Automaton approach to Arabic preprocessing
ITG	Inversion Transduction Grammars
IWSLT	International Workshop on Spoken Language Translation
KSR	Keystroke Ratio
LDC	Linguistic Data Consortium
LM	Language Model
MADA	Morphological Analysis and Disambiguation for Arabic
MAP	Maximum-a-Posteriori
MBR	Minimum Bayes Risk
ME	Maximum Entropy
MERT	Minimum Error Rate Training
METEOR	Metric for Evaluation of Translation with Explicit OrdeRing
ML	Maximum Likelihood
MT	Machine Translation
MTAG	Arabic preprocessing method based on Morphological TAGging
NIST	National Institute of Standards and Technology
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
NW	Newswire Texts
OOV	Out-Of-Vocabulary
PBT	Phrase-Based Translation
PER	Position-independent word Error Rate
POS	Part Of Speech
PP	Phrase Penalty
RBM	Restricted Boltzmann Machine
ROVER	Recognizer Output Voting Error Reduction

SMT	Statistical Machine Translation
SLT	Spoken Language Translation
SST	Speech-to-Speech Translation
SVM	Support Vector Machine
TC-STAR	Technology and Corpora for Speech to Speech Translation
TER	Translation Edit Rate
TIDES	Translingual Information Detection, Extraction and Summarization
TM	Translation Model
TOKAN	General Arabic Tokenizer
UW	University of Washington
WER	Word Error Rate
WP	Word Penalty
WSJ	Wall Street Journal
WT	Web Texts
YAMCHA	Yet Another Multipurpose CHunk Annotator

List of Figures

1.1 Levels of linguistic analysis in an MT system. 2
1.2 Architecture of the direct approach to SMT. 4
1.3 Example of a word alignment and the corresponding phrase segmentation. 5
1.4 Illustration of the phrase segmentation process. 6

3.1 Finite state automaton (FSA) for stripping prefixes off Arabic words. 18
3.2 Enhanced tagging algorithm of *MorphTagger*. 22

4.1 Monotone search algorithm for phrase-based translation. 28
4.2 Illustration of the (non-monotone) search process. 30
4.3 Detailed non-monotone search algorithm for phrase-based translation. 32
4.4 Illustration of the IBM constraints. 34
4.5 Illustration of the ITG reordering constraints. 35

5.1 LDA: example of a derivation tree. 39
5.2 Link grammar: example of a valid linkage satisfying all constraints. 40
5.3 Chunking and POS tagging: example of a chunk parse. 41

7.1 Examples of Arabic names with their corresponding English transcriptions. . . . 50
7.2 G2P co-segmentations for the pronunciation of the example word “mixing”. . . . 51
7.3 A schematic representation of our DBN for transliteration. 57
7.4 Lattice representation for the ROVER combination for the Arabic example name “تِر”. 59

8.1 Example of a word graph for the German input sentence “was hast du gesagt?” (English reference translation: “what did you say?”). 64

9.1 Effect of the standard phrase-based model scaling factor on the TER for the BTEC task. 86
9.2 Effect of the inverted phrase-based model scaling factor on the TER for the BTEC task. 86
9.3 Effect of the phrase count model scaling factor on the TER (threshold 1) for the BTEC task. 86
9.4 Effect of the phrase count model scaling factor on the TER (threshold 2) for the BTEC task. 86
9.5 Effect of the phrase count model scaling factor on the TER (threshold 3) for the BTEC task. 87
9.6 Effect of the language model scaling factor on the TER for the BTEC task. . . . 87
9.7 Effect of the standard word-based lexicon model scaling factor on the TER for the BTEC task. 87

9.8 Effect of the inverted word-based lexicon model scaling factor on the TER for the BTEC task. 87
9.9 Effect of the phrase penalty model scaling factor on the TER for the BTEC task. 88
9.10 Effect of the word penalty model scaling factor on the TER for the BTEC task. . 88
9.11 Effect of the number of lexical and coverage hypotheses for the BTEC task. . . . 89
9.12 Effect of the number of lexical and coverage hypotheses for the AE GALE NW task. 89
9.13 Effect of the number of lexical and coverage hypotheses for the AE GALE WT task. 89
9.14 Effect of the number of lexical and coverage hypotheses for the CE GALE NW task. 89
9.15 Histogram of the phrase lengths used during search for the BTEC task. 90
9.16 Histogram of the phrase lengths used during search for the GALE task. 90
9.17 Effect of the word graph density and n-best list size for the BTEC task. 93
9.18 Effect of the word graph density and n-best list size for the GALE task. 93
9.19 Keystroke ratio as a function of the average extension time. 101

List of Tables

3.1 Arabic prefixes handled in this work and their English meanings. 17
3.2 Arabic suffixes handled in this work and their English meanings. 17
3.3 Possible MADA analyses for the word وَالي. 20

6.1 LM perplexities on the GALE test sets. 45

9.1 Corpus statistics of the Arabic-English BTEC task. 72
9.2 Corpus statistics of the multilingual BTEC task. 73
9.3 Corpus statistics of the Arabic-English GALE task. 74
9.4 Corpus statistics of the Spanish-English Xerox task. 75
9.5 Corpus statistics of the German-English Xerox task. 75
9.6 Official results of the NIST 2006 Machine Translation Evaluation. 76
9.7 Results of the individual SRI Nightingale team members for the GALE task. . . 77
9.8 Official results of the IWSLT 2008 evaluation campaign. 78
9.9 Arabic corpus statistics of the BTEC task for different preprocessing approaches. 79
9.10 Effect of different approaches to Arabic preprocessing on the translation quality for the BTEC task. 79
9.11 Effect of different case handling strategies for English on the translation quality for the BTEC task. 80
9.12 Arabic corpus statistics of the GALE task for different preprocessing approaches. 81
9.13 Effect of different approaches to Arabic preprocessing on the translation quality for the GALE task. 82
9.14 Effect of different heuristics for alignment symmetrization on the translation quality for the BTEC task. 82
9.15 Effect of different models on the translation quality for the BTEC task. 83
9.16 Effect of different models on the translation quality for the AE GALE NW task. 84
9.17 Effect of different models on the translation quality for the AE GALE WT task. 85
9.18 Corpus statistics and translation quality for the Chinese-English GALE newswire task. 88
9.19 Effect of different language models on the translation quality for the GALE task. 90
9.20 Translation examples for the Arabic-English GALE NW task. 91
9.21 Translation examples for the Arabic-English GALE WT task. 92
9.22 Overview of system improvements for the IWSLT 2008 Arabic-English BTEC task. 93
9.23 Effect of syntactic feature functions for the Chinese-English BTEC task. 94
9.24 Translation examples for the Chinese-English BTEC task. 95
9.25 Effect of syntactic feature functions for the Japanese-English BTEC task. 96
9.26 Effect of syntactic feature functions for the Arabic-English BTEC task. 96
9.27 Corpus statistics of the Arabic-English GALE MT data for speech translation. . 98
9.28 Translation results for the AE GALE speech translation tasks. 99

9.29 Translation results for different forms of speech input. 99
9.30 Translation results for the Xerox Spanish-English and German-English task. . . . 100
9.31 Average extension time and keystroke ratio for the investigated generation strategies. 101
9.32 Keystroke ratio and average extension time for different word graph densities. . . 102
9.33 Comparison of the interactive search strategies for an example from the Xerox task. 103
9.34 Corpora characteristics for the Arabic names and their English renderings. 104
9.35 Comparison of different transliteration approaches for the 10k task. 105
9.36 Transliteration results of the ROVER combination for the 10k task. 106
9.37 Comparison of different transliteration approaches for the NGA task. 107

Bibliography

[Ackley & Hinton+ 85] D. Ackley, G. Hinton, T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, Vol. 9, No. 1, pp. 147–169, 1985.

[Akiba & Federico+ 04] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, J. Tsujii. Overview of the IWSLT04 evaluation campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 1–12, Kyoto, Japan, September/October 2004.

[Al-Onaizan & Curin+ 99] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. D. Lafferty, I. D. Melamed, D. Purdy, F. J. Och, N. A. Smith, D. Yarowsky. Statistical machine translation, final report, JHU workshop, 1999. `http://www.clsp.jhu.edu/ws99/projects/mt/final_report/mt-final-report.ps`.

[Al-Onaizan & Knight 02a] Y. Al-Onaizan, K. Knight. Machine transliteration of names in Arabic text. In *Proceedings of the ACL-02 workshop on computational approaches to semitic languages*, pp. 34–46, Philadelphia, PA, July 2002.

[Al-Onaizan & Knight 02b] Y. Al-Onaizan, K. Knight. Translating named entities using monolingual and bilingual resources. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 400 – 408, Philadelphia, PA, July 2002.

[Al-Onaizan & Mangu 07] Y. Al-Onaizan, L. Mangu. IBM Arabic ASR and MT integration for GALE. In *Proceedings of the ICASSP 2007, Special Session on Integrating Speech Recognition and Machine Translation*, pp. 1285 – 1288, Honolulu, Hawaii, USA, April 2007.

[Alshawi 96] H. Alshawi. Head automata and bilingual tiling: Translation with minimal representations. In *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 167 – 176, Santa Cruz, CA, June 1996.

[Arnold & Balkan+ 94] D. Arnold, L. Balkan, L. L. Humphreys, S. Meijer, L. Sadler. *Machine Translation.* Blackwell Publishers, 1994.

[Banerjee & Lavie 05] S. Banerjee, A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pp. 65–72, Ann Arbor, MI, June 2005.

[Bangalore & Joshi 99] S. Bangalore, A. K. Joshi. Supertagging: An approach to almost parsing. *Computational Linguistics*, Vol. 25, No. 2, pp. 237–265, 1999.

[Bangalore & Riccardi 00] S. Bangalore, G. Riccardi. Stochastic finite-state models for spoken language machine translation. In *Proc. of the Workshop on Embedded Machine Translation Systems, NAACL*, pp. 52–59, Seattle, WA, May 2000.

[Bangalore 00] S. Bangalore. A lightweight dependency analyzer for partial parsing. *Computational Linguistics*, Vol. 6, No. 2, pp. 113–138, 2000.

[Bar-Haim & Winter 05] R. Bar-Haim, Y. Winter. Choosing an optimal architecture for segmentation and POS-tagging of Modern Hebrew. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Computational Approaches to Semitic Languages*, pp. 39–46, Ann Arbor, MI, June 2005.

[Barrachina & Bender$^+$ 09] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, J.-M. Vilar. Statistical approaches to computer-assisted translation. *Computational Linguistics*, Vol. 35, No. 1, pp. 3–28, March 2009.

[Baxter 64] G. Baxter. On fixed points of the composite of commuting functions. *American Mathematical Society*, Vol. 15, No. 6, pp. 851–855, December 1964.

[Bellman 57] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[Bender & Hasan$^+$ 05] O. Bender, S. Hasan, D. Vilar, R. Zens, H. Ney. Comparison of generation strategies for interactive machine translation. In *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT)*, pp. 33–40, Budapest, Hungary, May 2005.

[Bender & Isbihani$^+$ 06] O. Bender, A. E. Isbihani, S. Hasan, S. Khadivi, J. Xu, R. Zens, Y. Zhang, H. Ney. The RWTH statistical machine translation system. In *Proceedings of the NIST Machine Translation Workshop*, 4 pages, Washington, DC, September 2006.

[Bender & Macherey$^+$ 03] O. Bender, K. Macherey, F. J. Och, H. Ney. Comparison of alignment templates and maximum entropy models for natural language understanding. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 11–18, Budapest, Hungary, April 2003.

[Bender & Matusov$^+$ 07] O. Bender, E. Matusov, S. Hahn, S. Hasan, S. Khadivi, H. Ney. The RWTH Arabic-to-English spoken language translation system. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 396–401, Kyoto, Japan, December 2007.

[Bender & Och$^+$ 03] O. Bender, F. J. Och, H. Ney. Maximum entropy models for named entity recognition. In *Proc. 7th Conf. on Computational Natural Language Learning (CoNLL)*, pp. 148–151, Edmonton, Canada, May 2003.

[Bender & Zens$^+$ 04] O. Bender, R. Zens, E. Matusov, H. Ney. Alignment Templates: the RWTH SMT System. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 79–84, Kyoto, Japan, September 2004.

[Bender & Zens$^+$ 09] O. Bender, R. Zens, H. Ney. Improvements for beam search in statistical machine translation. Advances in Machine Translation and Distillation, 2009. In print.

[Bender 02] O. Bender. Untersuchung zur Tagging-Aufgabenstellung in der Sprachverarbeitung. Diploma thesis (in German), Lehrstuhl für Informatik 6, RWTH Aachen University, Aachen, Germany, October 2002.

[Berger & Brown$^+$ 96] A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. R. Gillett, A. S. Kehler, R. L. Mercer. Language translation apparatus and method of using context-based translation models, United States Patent 5510981, April 1996.

[Berger & Della Pietra$^+$ 96] A. L. Berger, S. A. Della Pietra, V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, pp. 39–72, March 1996.

[Bertoldi 05] N. Bertoldi. *Statistical Models and Search Algorithms for Machine Translation.* Ph.D. thesis, University of Trento, Trento, Italy, March 2005.

[Birch & Callison-Burch$^+$ 06] A. Birch, C. Callison-Burch, M. Osborne, P. Koehn. Constraining the phrase-based, joint probability statistical translation model. In *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, pp. 154–157, New York City, NY, June 2006.

[Bisani & Ney 02] M. Bisani, H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. of the 7th Int. Conf. on Spoken Language Processing (ICSLP'02)*, pp. 105–108, Denver, CO, USA, September 2002.

[Bisani & Ney 03] M. Bisani, H. Ney. Multigram-based grapheme-to-phoneme conversation for LVCSR. In *European Conference on Speech Communication and Technology*, Vol. 2, pp. 933–936, Geneva, Switzerland, September 2003.

[Bisani & Ney 08] M. Bisani, H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, Vol. 50, No. 5, pp. 434–451, May 2008.

[Borthwick & Sterling$^+$ 98] A. Borthwick, J. Sterling, E. Agichtein, R. Grisham. NYU: Description of the MENE named entity system as used in MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 6 pages, Fairfax, VA, April 1998. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

[Bozarov & Sagisaka$^+$ 05] A. Bozarov, Y. Sagisaka, R. Zhang, G. Kikui. Improved speech recognition word lattice translation by confidence measure. In *Interspeech'2005 - Eurospeech, 9th European Conference on Speech Communication and Technology*, pp. 3197–3200, Lisbon, Portugal, September 2005.

[Brown & Cocke$^+$ 88] P. F. Brown, J. Cocke, S. Della Pietra, V. J. Della Pietra, F. Jelinek, R. L. Mercer, P. S. Roossin. A statistical approach to language translation. In *COLING '88: The 12th Int. Conf. on Computational Linguistics*, pp. 71–76, Budapest, Hungary, August 1988.

[Brown & Cocke$^+$ 90] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, Vol. 16, No. 2, pp. 79–85, June 1990.

[Brown & Della Pietra$^+$ 93] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, June 1993.

[Buckwalter 02] T. Buckwalter. Arabic Morphological Analyzer Version 1.0, 2002.

[Bulyko & Matsoukas+ 07] I. Bulyko, S. Matsoukas, R. Schwartz, L. Nguyen, J. Makhoul. Language model adaptation in machine translation from speech. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 117–120, Honolulu, HI, April 2007.

[Callison-Burch & Osborne+ 06] C. Callison-Burch, M. Osborne, P. Koehn. Re-evaluating the role of BLEU in machine translation research. In *Proc. 11th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 249–256, Trento, Italy, April 2006.

[Casacuberta & Llorens+ 01] F. Casacuberta, D. Llorens, C. Martínez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, J. M. Vilar. Speech-to-speech translation based on finite-state transducers. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 613–616, Salt Lake City, UH, May 2001.

[Chen & Goodman 98] S. F. Chen, J. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, 63 pages, August 1998.

[Chen & Rosenfeld 99] S. F. Chen, R. Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon University, Pittsburgh, PA, 1999.

[Chiang 05] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 263–270, Ann Arbor, Michigan, June 2005.

[Chiang 07] D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, Vol. 33, No. 2, pp. 201–228, June 2007.

[Darroch & Ratcliff 72] J. N. Darroch, D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, Vol. 43, pp. 1470–1480, 1972.

[Day 07] D. Day. Entity translation 2007 pilot evaluation (ET07). Workshop on Automatic Content Extraction (ACE). College Park, MD, 2007. `http://www.nist.gov/speech/tests/ace/2007/et`.

[DeNeefe & Knight+ 07] S. DeNeefe, K. Knight, W. Wang, D. Marcu. What can syntax-based MT learn from phrase-based MT? In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 755–763, Prague, Czech Republic, June 2007.

[Deselaers & Hasan+ 09] T. Deselaers, S. Hasan, O. Bender, H. Ney. A deep learning approach to machine transliteration. In *Proceedings of the EACL 2009 Workshop on Statistical Machine Translation*, pp. 233–241, Athens, Greece, March 2009.

[Diab & Hacioglu+ 04] M. Diab, K. Hacioglu, D. Jurafsky. Automatic tagging of Arabic text: From raw text to base phrase chunks. In D. M. Susan Dumais, S. Roukos, editors, *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 149–152, Boston, MA, May 2004. Association for Computational Linguistics.

[Diab 00] M. Diab. An unsupervised method for multilingual word sense tagging using parallel corpora: A preliminary investigation. In *Proceedings of the ACL-2000 workshop on Word Senses and Multilinguality*, pp. 1–9, Hong Kong, October 2000.

[Ding & Palmer 05] Y. Ding, M. Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 541–548, Ann Arbor, MI, June 2005.

[Doddington 02] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pp. 138–145, San Diego, CA, March 2002.

[Dorr 93] B. Dorr. *Machine Translation.* MIT Press, Cambridge, MA, 1993.

[Duda & Hart 73] R. O. Duda, P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, New York, NY, 1973.

[Eck & Hori 05] M. Eck, C. Hori. Overview of the IWSLT 2005 evaluation campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 1–22, Pittsburgh, PA, October 2005.

[Eppstein 01] D. Eppstein. Bibliography on k shortest paths and other "k best solutions" problems, `http://www.ics.uci.edu/~eppstein/bibs/kpath.bib`. URL, March 2001.

[Farwell & Gerber$^+$ 98] D. Farwell, L. Gerber, E. H. Hovy, editors. *Machine Translation and the Information Soup*, Vol. 1529 of *Lecture Notes in Computer Science*, Langhorne, PA, USA, October 1998. Springer Verlag.

[Fiscus 97] J. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 347–354, Santa Barbara, CA, USA, December 1997.

[Fordyce 07] C. S. Fordyce. Overview of the IWSLT 2007 evaluation campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 1–12, Trento, Italy, October 2007.

[Foster & Isabelle$^+$ 96] G. Foster, P. Isabelle, P. Plamondon. Word completion: A first step toward target-text mediated IMT. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pp. 394–399, Copenhagen, Denmark, August 1996.

[Foster & Isabelle$^+$ 97] G. Foster, P. Isabelle, P. Plamondon. Target-text mediated interactive machine translation. *Machine Translation*, Vol. 12, No. 1, pp. 175–194, 1997.

[Foster & Langlais$^+$ 02] G. Foster, P. Langlais, G. Lapalme. User-friendly text prediction for translators. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 148–155, Philadelphia, USA, July 2002.

[Foster 02] G. Foster. *Text Prediction for Translators.* Ph.D. thesis, Université de Montréal, May 2002.

[Freitag & Khadivi 07] D. Freitag, S. Khadivi. A sequence alignment model based on the averaged perceptron. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 238–247, Prague, Czech Republic, June 2007.

[Fry 05] J. Fry. Assembling a parallel corpus from RSS news feeds. In *MT Summit X: Workshop on Example-based Machine Translation*, pp. 59–62, Phuket, Thailand, September 2005.

[Galley & Graehl+ 06] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, I. Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Assoc. for Computational Linguistics (COLING/ACL)*, pp. 961–968, Sydney, Australia, July 2006.

[Galley & Hopkins+ 04] M. Galley, M. Hopkins, K. Knight, D. Marcu. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 273–280, Boston, MA, May 2004.

[Germann & Jahr+ 01] U. Germann, M. Jahr, K. Knight, D. Marcu, K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 228–235, Toulouse, France, July 2001.

[Germann & Jahr+ 04] U. Germann, M. Jahr, K. Knight, D. Marcu, K. Yamada. Fast decoding and optimal decoding for machine translation. *Artificial Intelligence*, Vol. 154, pp. 127–143, 2004.

[Graham & Knuth+ 94] R. L. Graham, D. E. Knuth, O. Patashnik. *Concrete Mathematics*. Addison-Wesley Publishing Company, Reading, MA, 2nd edition, 1994.

[Habash & Rambow 05] N. Habash, O. Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 573–580, Ann Arbor, MI, June 2005.

[Habash & Sadat 06] N. Habash, F. Sadat. Arabic preprocessing schemes for statistical machine translation. In *Proc. of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 49–52, New York City, NY, June 2006. Association for Computational Linguistics.

[Haizhou & Min+ 04] L. Haizhou, Z. Min, S. Jian. A joint source-channel model for machine transliteration. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 159–166, Barcelona, Spain, July 2004.

[Hasan & Bender+ 06] S. Hasan, O. Bender, H. Ney. Reranking translation hypotheses using structural properties. In *Proceedings of the EACL06 Workshop on Learning Structured Information in Natural Language Applications*, pp. 41–48, Trento, Italy, April 2006.

[Hasan & Ganitkevitch+ 08] S. Hasan, J. Ganitkevitch, H. Ney, J. Andrés-Ferrer. Triplet lexicon models for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pp. 372–381, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[Hasan & Zens+ 07] S. Hasan, R. Zens, H. Ney. Are very large N-best lists useful for SMT? In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting (HLT-NAACL), Companion Volume: Short Papers*, pp. 57–60, Rochester, NY, April 2007.

[Hermjakob & Knight+ 08] U. Hermjakob, K. Knight, H. D. III. Name translation in statistical machine translation - learning when to transliterate. In *Proc. of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pp. 389–397, Columbus, OH, June 2008.

[Hinton & Osindero+ 06] G. Hinton, S. Osindero, Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, Vol. 18, pp. 1527–1554, 2006.

[Huang & Vogel+ 03] F. Huang, S. Vogel, A. Waibel. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models*, pp. 9–16, Sapporo, Japan, July 2003.

[Huang & Vogel+ 04] F. Huang, S. Vogel, A. Waibel. Improving named entity translation combining phonetic and semantic similarities. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 281–288, Boston, MA, May 2004.

[Hutchins & Somers 92] W. J. Hutchins, H. L. Somers. *An Introduction to Machine Translation.* Academic Press, Cambridge, 1992.

[Isbihani & Khadivi+ 06] A. E. Isbihani, S. Khadivi, O. Bender, H. Ney. Morpho-syntactic Arabic preprocessing for Arabic to English statistical machine translation. In *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, pp. 15–22, New York, NY, June 2006.

[Iyer & Ostendorf 99] R. Iyer, M. Ostendorf. Modeling long distance dependence in language: topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 1, pp. 30–39, 1999.

[Jelinek 98] F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, Cambridge, MA, 1998.

[Ji & Blume+ 07] H. Ji, M. Blume, D. Freitag, R. Grishman, S. Khadivi, R. Zens. NYU-Fair Isaac-RWTH Chinese to English Entity Translation 07 system. In *NIST ET 2007 PI/Evaluation Workshop*, 10 pages, Washington, USA, March 2007.

[Kanthak & Ney 04] S. Kanthak, H. Ney. FSA: An efficient and flexible C ++ toolkit for finite state automata using on-demand computation. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 510–517, Barcelona, Spain, July 2004.

[Kasami 65] T. Kasami. An efficient recognition and syntax analysis algorithm for context-free languages, August 1965.

[Kashani & Popowich+ 07] M. M. Kashani, F. Popowich, A. Sarkar. Automatic transliteration of proper nouns from Arabic to English. In *Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages*, pp. 81–87, Linguistic Institute, Stanford, CA, July 2007.

[Kay 80] M. Kay. The proper place of men and machines in language translation, October 1980.

[Kneser & Ney 95] R. Kneser, H. Ney. Improved backing-off for M-gram language modelling. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 181–184, Detroit, MI, May 1995.

[Knight & Al-Onaizan 98] K. Knight, Y. Al-Onaizan. Translation with finite-state devices. In [Farwell & Gerber$^+$ 98], pp. 421–437.

[Knight & Graehl 97] K. Knight, J. Graehl. Machine transliteration. In *Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 128–135, Madrid, Spain, June 1997.

[Knight & Graehl 98] K. Knight, J. Graehl. Machine transliteration. *Computational Linguistics*, Vol. 24, No. 4, pp. 599–612, December 1998.

[Knight 99] K. Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, Vol. 25, No. 4, pp. 607–615, December 1999.

[Koehn & Hoang$^+$ 07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantine, E. Herbst. Moses: Open source toolkit for statistical machine translation. In *45th Annual Meeting of the Assoc. for Computational Linguistics (ACL): Poster Session*, pp. 177–180, Prague, Czech Republic, June 2007.

[Koehn & Knight 03] P. Koehn, K. Knight. Empirical methods for compound splitting. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 347–354, Budapest, Hungary, April 2003.

[Koehn & Och$^+$ 03] P. Koehn, F. J. Och, D. Marcu. Statistical phrase-based translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 127–133, Edmonton, Canada, May/June 2003.

[Koehn 03] P. Koehn. *Noun Phrase Translation.* Ph.D. thesis, University of Southern California, 2003.

[Koehn 04] P. Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proc. of the 6th Conf. of the Association for Machine Translation in the Americas (AMTA 04)*, pp. 115–124, Washington DC, September/October 2004.

[Kudo & Matsumoto 03] T. Kudo, Y. Matsumoto. Fast methods for kernel-based text analysis. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 24–31, Sapporo, Japan, July 2003.

[Kudo & Yamamoto$^+$ 04] T. Kudo, K. Yamamoto, Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. In D. Lin, D. Wu, editors, *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Kumar & Byrne 03] S. Kumar, W. Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 142–149, Edmonton, Canada, May/June 2003.

[Kumar & Byrne 04] S. Kumar, W. Byrne. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 169–176, Boston, MA, May 2004.

[Kumar & Byrne 05] S. Kumar, W. Byrne. Local phrase reordering models for statistical machine translation. In *Human Language Technology Conf. / Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 161–168, Vancouver, Canada, October 2005.

[Lafferty & McCallum$^+$ 01] J. Lafferty, A. McCallum, F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pp. 282–289, Williamstown, MA, USA, June 2001.

[Lafferty & Sleator$^+$ 92] J. Lafferty, D. Sleator, D. Temperley. Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pp. 89–97, Cambridge, MA, October 1992.

[Langlais & Foster$^+$ 00a] P. Langlais, G. Foster, G. Lapalme. Unit completion for a computer-aided translation typing system. *Machine Translation*, Vol. 15, No. 4, pp. 267–294, 2000.

[Langlais & Foster$^+$ 00b] P. Langlais, G. Foster, G. Lapalme. TransType: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems*, pp. 46–51, Seattle, Wash., May 2000.

[Larkey & Ballesteros$^+$ 02] L. S. Larkey, L. Ballesteros, M. E. Connell. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 275–282, Tampere, Finland, August 2002.

[Lavie & Agarwal 07] A. Lavie, A. Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *45th Annual Meeting of the Assoc. for Computational Linguistics (ACL): Workshop on Statistical Machine Translation*, pp. 228–231, Prague, Czech Republic, June 2007.

[Lee & Chang 03] C.-J. Lee, J. S. Chang. Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, pp. 96–103, Edmonton, Canada, May 2003.

[Lee & Papineni$^+$ 03] Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, H. Hassan. Language model based Arabic word segmentation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 399–406, Sapporo, Japan, July 2003.

[Lee 04] Y.-S. Lee. Morphological analysis for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 57–60, Boston, MA, May 2004.

[Lehmer 70] D. H. Lehmer. Permutations with strongly restricted displacements. In *Combinatorial Theory and its Applications*, Vol. 2, pp. 755–770. North-Holland, Amsterdam, The Netherlands, 1970.

[Lin 04] D. Lin. A path-based transfer model for machine translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pp. 625–630, Geneva, Switzerland, August 2004.

[Maamouri & Bies$^+$ 04] M. Maamouri, A. Bies, T. Buckwalter, W. Mekki. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus, 2004.

[Macherey & Bender$^+$ 03] K. Macherey, O. Bender, H. Ney. Multi-level error handling for tree based dialogue course management. In *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, pp. 123–128, Chateau-d'Oex-Vaud, Switzerland, August 2003.

[Macherey & Bender$^+$ 09] K. Macherey, O. Bender, H. Ney. Applications of statistical machine translation approaches to spoken language understanding. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 4, pp. 803–818, May 2009.

[Macklovitch & Nguyen$^+$ 05] E. Macklovitch, N. Nguyen, R. Silva. User evaluation report, 2005.

[Macklovitch 06] E. Macklovitch. TransType2: The last word. In *Proc. of the Fifth Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 167–172, Genoa, Italy, May 2006.

[Mansour & Sima'an$^+$ 07] S. Mansour, K. Sima'an, Y. Winter. Smoothing a lexicon-based POS tagger for Arabic and Hebrew. In *45th Annual Meeting of the Assoc. for Computational Linguistics (ACL): Workshop on Computational Approaches to Semitic Languages*, pp. 97–103, Prague, Czech Republic, June 2007.

[Mansour 08] S. Mansour. Combining character and morpheme based models for part-of-speech tagging of Semitic languages. Master thesis, Technion - Computer Science Department, Israel Institute of Technology, Haifa, Israel, April 2008.

[Marcu & Wong 02] D. Marcu, W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 133–139, Philadelphia, PA, July 2002.

[Matusov & Hillard$^+$ 07] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tür, M. Ostendorf, H. Ney. Improving speech translation by automatic boundary prediction. In *Interspeech'2007, 8th Annual Conference of the International Speech Communication Association*, pp. 2449–2452, Antwerp, Belgium, August 2007.

[Matusov & Kanthak$^+$ 05] E. Matusov, S. Kanthak, H. Ney. On the integration of speech recognition and statistical machine translation. In *Interspeech'2005 - Eurospeech, 9th European Conference on Speech Communication and Technology*, pp. 3177–3180, September 2005.

[Matusov & Kanthak$^+$ 06] E. Matusov, S. Kanthak, H. Ney. Integrating speech recognition and machine translation: Where do we stand? In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. V–1217–V–1220, Toulouse, France, May 2006.

[Matusov & Leusch$^+$ 05] E. Matusov, G. Leusch, O. Bender, H. Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 148–154, Pittsburgh, PA, October 2005.

[Matusov & Mauser+ 06] E. Matusov, A. Mauser, H. Ney. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 158–165, Kyoto, Japan, November 2006.

[Matusov & Zens+ 06] E. Matusov, R. Zens, D. Vilar, A. Mauser, M. Popović, S. Hasan, H. Ney. The RWTH machine translation system. In *TC-STAR Workshop on Speech-to-Speech Translation*, pp. 31–36, Barcelona, Spain, June 2006.

[Mauser & Vilar+ 07] A. Mauser, D. Vilar, G. Leusch, Y. Zhang, H. Ney. The RWTH machine translation system for IWSLT 2007. In *International Workshop on Spoken Language Translation*, pp. 161–168, Trento, Italy, October 2007.

[Mauser & Zens+ 06] A. Mauser, R. Zens, E. Matusov, S. Hasan, H. Ney. The RWTH statistical machine translation system for the IWSLT 2006 evaluation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 103–110, Kyoto, Japan, November 2006.

[Melamed 04] I. D. Melamed. Statistical machine translation by parsing. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 653–660, Barcelona, Spain, July 2004.

[Meng & Lo+ 01] H. Meng, W.-K. Lo, B. Chen, K. Tang. Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 311–314, Trento, Italy, December 2001.

[Mohri & Pereira+ 02] M. Mohri, F. C. Pereira, M. D. Riley. AT&T FSM Library - finite state machine library, 2002. `http://www.research.att.com/~fsmtools/fsm`.

[Moore & Quirk 07] R. C. Moore, C. Quirk. Faster beam-search decoding for phrasal statistical machine translation. In *Machine Translation Summit XI*, 7 pages, Copenhagen, Denmark, September 2007.

[Moore 03] R. C. Moore. Learning translations of named-entity phrases from parallel corpora. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 259–266, Budapest, Hungary, April 2003.

[Ney & Aubert 94] H. Ney, X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 1355–1358, Yokohama, Japan, September 1994.

[Ney 99] H. Ney. Speech translation: Coupling of recognition and translation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 517–520, Phoenix, AR, March 1999.

[Ney 01] H. Ney. Stochastic modelling: From pattern classification to language translation. In *Proc. Data-Driven Machine Translation Workshop, 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 33–37, Toulouse, France, July 2001.

[Och & Gildea+ 03] 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, D. Radev. Syntax for statistical machine translation. Technical report, Johns Hopkins University 2003 Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, 120 pages, August 2003.

[Och & Gildea+ 04] 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, D. Radev. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 161–168, Boston,MA, May 2004.

[Och & Ney 00] F. J. Och, H. Ney. A comparison of alignment models for statistical machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 1086–1090, Saarbrücken, Germany, August 2000.

[Och & Ney 02] F. J. Och, H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, PA, July 2002.

[Och & Ney 03] F. J. Och, H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, March 2003.

[Och & Ney 04] F. J. Och, H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417–449, December 2004.

[Och & Tillmann+ 99] F. J. Och, C. Tillmann, H. Ney. Improved alignment models for statistical machine translation. In *Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, June 1999.

[Och & Zens+ 03] F. J. Och, R. Zens, H. Ney. Efficient search for interactive statistical machine translation. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 387–393, Budapest, Hungary, April 2003.

[Och 02] F. J. Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph.D. thesis, Lehrstuhl für Informatik 6, Computer Science Department, RWTH Aachen University, Aachen, Germany, October 2002.

[Och 03] F. J. Och. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 160–167, Sapporo, Japan, July 2003.

[Papineni & Roukos+ 97] K. A. Papineni, S. Roukos, R. T. Ward. Feature-based language understanding. In *European Conf. on Speech Communication and Technology*, pp. 1435–1438, Rhodes, Greece, September 1997.

[Papineni & Roukos+ 98] K. A. Papineni, S. Roukos, R. T. Ward. Maximum likelihood and discriminative training of direct translation models. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 189–192, Seattle, WA, May 1998.

[Papineni & Roukos+ 02] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, Philadelphia, PA, July 2002.

[Paul 06] M. Paul. Overview of the IWSLT06 evaluation campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 1–15, Kyoto, Japan, November 2006.

[Pearl 88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988. Revised second printing.

[Press & Teukolsky+ 02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK, 2002.

[SchlumbergerSema S.A. & Intituto Tecnológico de Informática+ 01] SchlumbergerSema S.A., Intituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen - Lehrstuhl für Informatik 6, Recherche Appliquée en Linguistique Informatique Laboratory - University of Montreal, Celer Soluciones, Société Gamma, Xerox Research Centre Europe. TT2. TransType2 - computer assisted translation. Project technical annex., 2001.

[Schultz & Jou+ 04] T. Schultz, S.-C. Jou, S. Vogel, S. Saleem. Using word lattice information for a tighter coupling in speech translation systems. In *Interspeech'2004 - ICSLP, 8th International Conference on Spoken Language Processing*, pp. 41–44, Jeju Island, Korea, October 2004.

[Seymore & Rosenfeld 97] K. Seymore, R. Rosenfeld. Using story topics for language model adaptation. In *European Conf. on Speech Communication and Technology*, pp. 1987–1990, Rhodes, Greece, September 1997.

[Sherif & Kondrak 07] T. Sherif, G. Kondrak. Substring-based transliteration. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 944–951, Prague, Czech Republic, June 2007.

[Sleator & Temperley 93] D. Sleator, D. Temperley. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*, 14 pages, Tilburg/Durbuy, The Netherlands/Belgium, August 1993.

[Snover & Dorr+ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA 06)*, pp. 223–231, Cambridge, MA, August 2006.

[Stalls & Knight 98] B. Stalls, K. Knight. Translating names and technical terms in Arabic text. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 34–41, Montréal, Québec, Canada, August 1998.

[Steinbiss & Tran+ 94] V. Steinbiss, B. Tran, H. Ney. Improvements in beam search. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP'94)*, pp. 2143–2146, September 1994.

[Stolcke & Chen+ 06] A. Stolcke, B. Chen, H. Franco, V. R. R. Gadde, M. Graciarena, M.-Y. Hwang, K. Kirchhoff, N. Morgan, X. Lin, T. Ng, M. Ostendorf, K. Snmez, A. Venkataraman, D. Vergyri, W. Wang, J. Zheng, Q. Zhu. Recent innovations in speech-to-text transcription at SRI-ICSI-UW. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, pp. 1729–1744, September 2006.

[Stolcke & Konig+ 97] A. Stolcke, Y. Konig, M. Weintraub. Explicit word error minimization in N-best list rescoring. In *European Conf. on Speech Communication and Technology*, pp. 163–166, Rhodes, Greece, September 1997.

[Stolcke 02] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proc. of the 7th Int. Conf. on Spoken Language Processing (ICSLP'02)*, Vol. 2, pp. 901–904, Denver, CO, September 2002.

[Takezawa & Sumita$^+$ 02] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, S. Yamamoto. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of the Third Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 147–152, Las Palmas, Spain, May 2002.

[Tillmann & Ney 00] C. Tillmann, H. Ney. Word re-ordering and DP-based search in statistical machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Tillmann & Ney 03] C. Tillmann, H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, Vol. 29, No. 1, pp. 97–133, March 2003.

[Tillmann & Vogel$^+$ 97] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga. A DP-based search using monotone alignments in statistical translation. In *Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 289–296, Madrid, Spain, July 1997.

[Tillmann & Xia 03] C. Tillmann, F. Xia. A phrase-based unigram model for statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting (HLT-NAACL), Companion Volume: Short Papers*, pp. 106–108, Edmonton, Canada, May/June 2003.

[Tillmann 03] C. Tillmann. A projection extension algorithm for statistical machine translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 1–8, Sapporo, Japan, July 2003.

[Tillmann 06] C. Tillmann. Efficient dynamic programming search algorithms for phrase-based SMT. In *Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pp. 9–16, New York City, NY, June 2006.

[Tjong Kim Sang & Buchholz 00] E. F. Tjong Kim Sang, S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 127–132, Lisbon, Portugal, September 2000.

[Tomás & Casacuberta 01] J. Tomás, F. Casacuberta. Monotone statistical translation using word groups. In *Machine Translation Summit VIII*, pp. 357–361, Santiago de Compostela, September 2001.

[Tomás & Casacuberta 04] J. Tomás, F. Casacuberta. Statistical machine translation decoding using target word reordering. In *Structural, Syntactic, and Statistical Pattern Recognition*, Vol. 3138 of *Lecture Notes in Computer Science*, pp. 734–743. Springer-Verlag, Lisbon, Portugal, August 2004.

[Turian & Shen$^+$ 03] J. P. Turian, L. Shen, I. D. Melamed. Evaluation of machine translation and its evaluation. Technical Report Proteus technical report 03-005, Computer Science Department, New York University, 8 pages, 2003.

[Ueffing & Och+ 02] N. Ueffing, F. J. Och, H. Ney. Generation of word graphs in statistical machine translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 156–163, Philadelphia, PA, July 2002.

[Ueffing 05] N. Ueffing. *Confidence Measures for Statistical Machine Translation.* Ph.D. thesis, Lehrstuhl für Informatik 6, Computer Science Department, RWTH Aachen University, Aachen, Germany, 2005.

[Vidal 97] E. Vidal. Finite-state speech-to-speech translation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 111–114, Munich, Germany, April 1997.

[Vilar & Stein+ 08] D. Vilar, D. Stein, Y. Zhang, E. Matusov, A. Mauser, O. Bender, S. Mansour, H. Ney. The RWTH machine translation system for IWSLT 2008. In *International Workshop on Spoken Language Translation 2008*, pp. 190–197, Waikiki, Hawaii, October 2008.

[Vogel & Ney+ 96] S. Vogel, H. Ney, C. Tillmann. HMM-based word alignment in statistical translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pp. 836–841, Copenhagen, Denmark, August 1996.

[Vogel 03] S. Vogel. SMT decoder dissected: Word reordering. In *Int. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pp. 561–566, Beijing, China, October 2003.

[Wahlster 00] W. Wahlster, editor. *Verbmobil: Foundations of speech-to-speech translations.* Springer-Verlag, Berlin, 2000.

[Wang & Waibel 97] Y.-Y. Wang, A. Waibel. Decoding algorithm in statistical machine translation. In *Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 366–372, Madrid, Spain, July 1997.

[Weaver 55] W. Weaver. Translation. In W. N. Locke, A. D. Booth, editors, *Machine Translation of Language*, pp. 15–23. MIT Press, Cambridge, Mass., 1955.

[Wu & Wong 98] D. Wu, H. Wong. Machine translation with a stochastic grammatical channel. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 1408–1414, Montréal, Québec, Canada, August 1998.

[Wu 95] D. Wu. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *14th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1328–1334, Montreal, Canada, August 1995.

[Wu 96] D. Wu. A polynomial-time algorithm for statistical machine translation. In *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 152–158, Santa Cruz, CA, June 1996.

[Wu 97] D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, Vol. 23, No. 3, pp. 377–403, September 1997.

[XTAG Research Group 01] XTAG Research Group. A lexicalized tree adjoining grammar for English, 2001.

[Yamada & Knight 01] K. Yamada, K. Knight. A syntax-based statistical translation model. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 523–530, Toulouse, France, July 2001.

[Yamada & Knight 02] K. Yamada, K. Knight. A decoder for syntax-based statistical MT. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 303–310, Philadelphia, PA, July 2002.

[Younger 67] D. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, Vol. 10, No. 2, pp. 189–208, 1967.

[Zens & Bender$^+$ 05] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, H. Ney. The RWTH phrase-based statistical machine translation system. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 155–162, Pittsburgh, PA, October 2005.

[Zens & Ney 04] R. Zens, H. Ney. Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 257–264, Boston, MA, May 2004.

[Zens & Ney 05] R. Zens, H. Ney. Word graphs for statistical machine translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pp. 191–198, Ann Arbor, MI, June 2005.

[Zens & Ney 06] R. Zens, H. Ney. N-gram posterior probabilities for statistical machine translation. In *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, pp. 72–77, New York City, NY, June 2006.

[Zens & Och$^+$ 02] R. Zens, F. J. Och, H. Ney. Phrase-based statistical machine translation. In *25th German Conf. on Artificial Intelligence (KI2002)*, pp. 18–32, Aachen, Germany, September 2002. Springer Verlag.

[Zens 08] R. Zens. *Phrase-based Statistical Machine Translation: Models, Search, Training.* Ph.D. thesis, RWTH Aachen University, Aachen, Germany, February 2008.

[Zhao & Eck$^+$ 04] B. Zhao, M. Eck, S. Vogel. Language model adaptation for statistical machine translation with structured query models. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pp. 411–417, Geneva, Switzerland, August 2004.

[Zollmann & Venugopal 06] A. Zollmann, A. Venugopal. Syntax augmented machine translation via chart parsing. In *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, pp. 138–141, New York City, NY, June 2006.

Curriculum Vitae

Personal Details

Name	Oliver Bender
Address	Reumontstr. 65, 52064 Aachen
Email	oliver.bender@gmail.com
Date of birth	Sep 13, 1974
Place of birth	Heinsberg, Germany
Nationality	German
Marital status	married

Education

1981 – 1985	Kath. Grundschule, Gangelt-Breberen
1985 – 1994	Bischöfl. Gymnasium St. Ursula, Geilenkirchen Allgemeine Hochschulreife

Studies

Oct 1997 – Nov 2002	Study of Computer Science at RWTH Aachen University minor subject: physics degree: Diplom-Informatiker
Dec 2002 – Mar 2010	PhD student at RWTH Aachen University

Work Experience

Sep 1994 – Aug 1997	Math.-Techn. Assistent at RWTH Aachen University Communication Networks (ComNets)
Sep 1997 – Nov 2002	System Administrator at RWTH Aachen University Human Language Technology and Pattern Recognition Group
Dec 2002 – Oct 2008	Research Assistant at RWTH Aachen University Human Language Technology and Pattern Recognition Group
Jul 2007 – Sep 2007	Visiting researcher at SRI International, USA Speech Technology and Research (STAR) Laboratory
Nov 2008 – present	Research Engineer at Nuance Communications Aachen GmbH R&D – Embedded