Effective Training and Efficient Decoding for Statistical Machine Translation

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
Jörn Wübker
aus Münster, Westfalen

Berichter: Universitätsprofessor Dr.-Ing. Hermann Ney
Universitätsprofessor Dr. Josef van Genabith

Tag der mündlichen Prüfung: 02. Februar 2017

Acknowledgments

I would like to express my gratitude to a number of people that have supported me during my years as a PhD student.

First of all, my thanks go to Prof. Dr.-Ing. Hermann Ney for the opportunity to work with and learn from the best at his institute, and for advising and supporting my research that resulted in this PhD thesis.

I also would like to thank Prof. Dr. Josef van Genabith for his interest in my work and agreeing to review this thesis.

Special thanks to all of my colleagues during my time at i6. First and foremost, my diploma thesis supervisor and mentor Arne should be acknowledged. I would further like to mention Stephan, Saab, Tamer, Carmen, Matthias, Martin, Markus, Patrick, Andy, JTP, Malte, Minwei, Christoph, David, Ralf, Felix and Yunsu. It was a pleasure working with you and I learned a lot from our joint ventures. My thanks also go to Albert, Amr, Björn, Christian, Christian, Christian, Christian, Daniel, David, David, Eugen, Evgeny, Farzad, Georg, Gregor, Harald, Jens, Jia, Jonas, Kazuki, Mahdi, Maja, Markus, Markus, Michal, Muhammad, Oskar, Parnia, Patrick, Pavel, Philippe, Saša, Simon, Stefan, Thomas, Tobias, Volker, Yannick, Yuqi and Zoltán, who all contributed to making the past six years a fun journey.

I further want to thank our system administrators Stefan, Kai, JTP, Pavel, Christian, David and David for their hard work in maintaining the infrastructure that I took for granted too many times, as well as Andrea, Steffi, Katja and Gisela for their efficiency and knowledgeability which made my life a lot easier.

My thanks also go to my short-term colleagues during the internship at Microsoft Research for advice, fruitful discussions, making me feel welcome, and the entire "Microsoft experience": My mentor Mei-Yuh Hwang, team leader Arul Menezes as well as Chris Quirk, Will Lewis, Michel Galley, Hany Hassan and Xiaodong He.

Finally and most importantly, I would like to convey my gratitude towards my wife Rebekka, my parents and my sister. Thank you for bearing with me all the way and for your relentless support and understanding throughout these years!

Abstract

Statistical machine translation, the task of translating text from one natural language into another using statistical models, can be divided into three main problems: modeling, search and training. This thesis gives a detailed description of the most popular approach to statistical machine translation, the phrase-based paradigm, and presents several improvements to the state of the art in all three of the aspects mentioned above.

Regarding the search problem, we propose three novel language model look-ahead techniques which can considerably increase time efficiency of the algorithm with different quality tradeoffs. They are evaluated in detail with respect to their effect on translation quality, translation speed, number of language model queries and number of generated nodes within the search graph. We can show that our final system outperforms the popular Moses toolkit in terms of translation speed.

With regard to the modeling problem we extend the state of the art with novel smoothing models based on word classes. Data sparsity is a common pitfall for statistical models. We leverage word classes that can be learned in an unsupervised fashion in order to re-parameterize the standard phrase-based models, resulting in a smoother probability distribution and reduced sparsity.

The largest part of this work is dedicated to the training problem. We investigate both generative and discriminative training methods, two fundamentally different approaches to learning statistical models. Our generative procedure is inspired by the expectation-maximization algorithm and based on force-aligning the training data with the application of the leave-one-out technique to avoid overfitting. Its advantage over the standard heuristic model extraction is that it provides a framework which uses the same consistent models in training and search. The initial technique is further developed into a length-incremental procedure which does not require initialization with a Viterbi word alignment and is thus not biased by its inconsistencies. Both the learning procedure and the resulting models are analyzed in detail.

As a discriminative training procedure, we employ a gradient-based method to optimize an expected Bleu objective function. Our novel contribution is the application of the resilient backpropagation algorithm, which is experimentally shown to be superior to several previously proposed techniques. It is also significantly more time and memory efficient than previous work, so that we can run training on the largest data set reported in the literature to date.

Our novel techniques are experimentally evaluated against internal and external results on large-scale translation tasks and within public evaluation campaigns. Especially the word class language model and discriminative training procedure prove to be valuable for state-of-the-art large scale translation systems.

Kurzfassung

Als statistische maschinelle Übersetzung bezeichnet man die Problemstellung, mit Hilfe von statistischen Modellen Text aus einer natürlichen Sprache in eine andere zu übersetzen. Man kann sie in drei Unterprobleme unterteilen: Modellierung, Suche und Training. Diese Doktorarbeit beschreibt den populärsten Ansatz für statistische maschinelle Übersetzung, die Phrasen-basierte Übersetzung, im Detail und führt Verbesserungen zum aktuellen Stand der Technik in allen drei der oben erwähnten Aspekte ein.

Für das Suchproblem werden drei neuartige Techniken zur Sprachmodellvorschau (engl.: language model look-ahead) vorgestellt, die die Zeit- und Speichereffizienz des Suchalgorithmus beträchtlich erhöhen können und unterschiedliche Wirkung auf die Qualität der Ausgabe haben. Ihr Einfluss auf die Qualität und Geschwindigkeit der Übersetzungen, sowie auf die Anzahl von Sprachmodellanfragen und generierten Knoten im Suchgraphen wird detailliert ausgewertet. Wir können zeigen, dass unser endgültiges System die weitverbreitete Software "Moses" in ihrer Übersetzungsgeschwindigkeit übertrifft.

In Bezug auf das Problem der Modellierung erweitern wir den Stand der Technik mit neuartigen Glättungsmodellen, die auf Wortklassen basieren. Auch bei großen Datenmengen gibt es bei statistischen Modellen oft viele Parameter, deren Wert nur aus sehr wenigen Beobachtungen geschätzt werden kann. In dieser Arbeit werden die Standardmodelle des Phrasen-basierten Ansatzes zur statistischen maschinellen Übersetzung mit Hilfe von Wortklassen neu parameterisiert, was zu einer glatteren Wahrscheinlichkeitsverteilung und einer besseren Datenlage zur Parameterschätzung führt. Die Wortklassen können unüberwacht gelernt werden.

Der größte Teil dieser Doktorarbeit beschäftigt sich mit dem Trainingsproblem. Wir untersuchen sowohl generative, als auch diskriminative Trainingsverfahren, welche zwei fundamental unterschiedliche Ansätze zum Lernen statistischer Modelle darstellen. Unser generatives Verfahren ist an den Expectation-Maximization-Algorithmus angelehnt und basiert auf einer erzwungenen Alignierung der Trainingsdaten mit dem Suchverfahren, wobei eine "leave-one-out"-Technik angewandt wird um Überanpassung zu vermeiden. Der Vorteil gegenüber der üblichen heuristischen Modellextraktion ist, dass im Training und später in der Suche dieselben Modelle verwendet werden. Diese Technik wird außerdem zu einer Längen-inkrementellen Methode weiterentwickelt, welche nicht mit einem Viterbi-Wortalignment initialisiert wird. Dessen Inkonsistenzen werden daher nicht in die Modelle weiterpropagiert. Sowohl das Lernverfahren, als auch die resultierenden Modelle werden detailliert untersucht.

Als diskriminative Trainingsmethode verwenden wir ein Gradienten-basiertes Verfahren, das den erwarteten Bleu-Wert optimiert. Unser neuer wissenschaftlicher Beitrag ist der Einsatz des Resilient-Backpropagation-Algorithmus, dessen Überlegenheit zu mehreren in der Literatur angewandten Techniken experimentell gezeigt wird. Im Vergleich zu früher verwendeten Methoden zeichnet er sich außerdem durch eine signifikant höhere Zeit- und Speichereffizienz aus, so dass wir unser Training auf dem größten Datensatz durchführen können, von dem in der Literatur bisher

berichtet wurde.

Unsere neuartigen Methoden werden auf großen Datensätzen und in öffentlichen Evaluierungen mit internen und externen Resultaten experimentell verglichen. Dabei zeigt sich, dass insbesondere das Wortklassen-Sprachmodell sowie unser diskriminatives Trainingsverfahren auch für große und moderne System, die dem aktuellen Stand der Technik entsprechen, hilfreiche Erweiterungen darstellen.

Contents

1. Introduction

1.1 Machine Translation

Machine translation (MT) is the task of translating a given text written in one natural language into another natural language in an automatic fashion. In spite of more than 60 years of research devoted to this topic, fully automatic high-quality translation remains an unsolved problem.

However, machine translation has proven beneficial for a number of applications. In many scenarios, the reader only wants to understand the overall meaning of a text, without requiring a perfect translation. The nowadays pervading social media are a good example for such a scenario. Some of them already offer integrated automatic translation services for messages or posts in foreign languages. Further, within the professional translation industry, machine translation can be of assistance to the human translators, increasing productivity. This can be done either by providing a rough translation suggestion that is then post-edited by the user, or by a more sophisticated human-computer interaction scheme within computer-assisted translation (CAT). A survey comparing post editing to interactive translation is described in [Green & Chuang$^+$ 14].

In order to classify the different approaches to machine translation, we take a look at the pyramid diagram introduced by [Vauquois 68] shown in Figure 1.1. Starting in the lower left corner of the pyramid, we can take different paths to reach the target text in the lower right corner. By analyzing the source text we can attempt to produce a semantic representation encoded in a meta-language, here called *interlingua*. Using this representation, its corresponding target translation can be generated. Although theoretically appealing, this approach complicates the problem of machine translation by splitting it into three hard sub-tasks: definition of a suitably powerful meta-language, analysis and generation. A meta-language capable of describing without ambiguity any semantics expressible with natural language could be viewed as a mapping from all "subsets" of the real world into its unique abstract representation, which is necessarily more complex than any human language. This approach has therefore never been successful for unre-

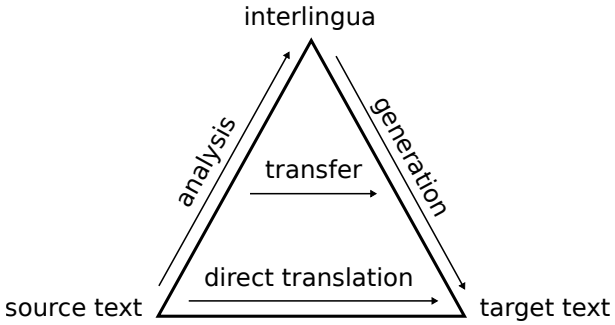


Figure 1.1: Pyramid diagram for classifying different translation approaches.

stricted machine translation in practice. In this work we will focus on the direct path through the base of the pyramid. Here, the syntax and semantics of natural language are ignored and translation is modeled as a direct mapping between word sequences. Between the two extremes, it is possible to follow a compromise and choose any level of integrating semantic or syntactic analysis. Syntax-based machine translation is an example for such a compromise, that has recently experienced a renaissance and has now proven capable of performing similar or better than the direct sequence mapping approach [Bojar & Buck$^+$ 14].

Orthogonal to the above classification, we can also distinguish between rule-based and data-driven approaches to machine translation. In rule-based systems, human experts devise a set of fixed rules which are used to create the target translation from a given source text. The number of rules necessary to fully capture the complex dependencies in human language can be very large. Creating this set of rules is an expensive process, as it involves extensive human effort. Data-driven approaches, on the other hand, leverage bilingual and monolingual document collections in order to automatically learn correspondences between two languages. Translation is treated as a statistical decision problem, where the translation engine searches for the most likely translation of a source language sentence. The underlying probability distributions can be modeled in different ways, but are generally learned and tuned on the available data. This approach is referred to as statistical machine translation. It will be described in more detail in Section 3.2, as it is the approach we follow in this work.

1.2 Generative and Discriminative Training

One main focus of this work is on introducing several novel training algorithms for statistical machine translation and evaluating them against the state of the art. We make use of two fundamentally different approaches: generative and discriminative training. Generative training is closely connected with the original source-channel approach to statistical machine translation [Weaver 55], where the knowledge sources are divided into a language model and a translation model. The principal idea is a generative story, where the source text that needs to be translated is generated by encoding the target text and transmitting it through a noisy channel that accounts for statistical variation. Generative models are typically estimated by maximizing likelihood of a training corpus. Discriminative training, on the other hand does not rely on a generative story, but directly optimizes the models with respect to a quality or error measure of choice. In this work we use the expected Bleu score for this purpose. This has the advantage of being more closely related with the way we evaluate quality on the task that is being performed. It further allows the definition of a large number, usually millions, of very fine-grained features that can capture arbitrary aspects of a translation. Most commonly, these features are trained with gradient-based hill-climbing algorithms that apply regularization to avoid overfitting.

This work introduces two novel training algorithms: A generative, length-incremental phrase training scheme based on force-aligning the training data, and a discriminative procedure that optimizes an expected Bleu objective function with the resilient backpropagation algorithm (RPROP). Both of these methods rely on a leaving-one-out technique to make the best use of the data.

1.3 About this Thesis

The technical core of this thesis is subsumed in Chapters 4, 5 and 6. In Chapters 4 and 5 we aim at giving a detailed, complete and consistent description of the current state of the art of models and search in the phrase-based translation paradigm. Here, the description of previous work is interleaved with novel contributions of this work. In the respective final Sections of each

Chapter, namely Sections 4.3, 4.4, 5.5 and 5.6, we explicitly highlight which parts are the novel contributions and which parts are based on previous publications by other authors. Chapter 6 is structured similarly in order to consistently communicate the current state of the art. Here, Sections 6.1, 6.2.1 and 6.2.2 are reserved for previous work, while the remainder of the chapter introduces novel contributions. Again, the final two Sections 6.4 and 6.5 explicitly put the Chapter into perspective with respect to previous work and highlight our contributions.

The remainder of the thesis is structured as follows. Chapter 2 states the scientific goals of this work, and in Chapter 3 some basic concepts of statistical machine translation, which are required to understand our contributions, are reviewed. Chapters 4 and 5 give a comprehensive account of modeling and search in phrase-based statistical machine translation, including the novel class based smoothing models and language model look-ahead pruning. The generative and discriminative training methods applied in this work are described in Chapter 6. In Chapter 7, we perform an extensive analysis and evaluation of all of our novel methods with other popular techniques and show results on several large-scale shared tasks. Finally, Chapter 8 summarizes how we achieved our scientific goals.

1.4 Previously Published

The following scientific publications were published by the author of this thesis at peer-reviewed conferences during the course of this thesis:

- Models
  - [Huck & Wuebker$^+$ 13] A Phrase Orientation Model for Hierarchical Machine Translation
  - [Wuebker & Peitz$^+$ 13b] Improving Statistical Machine Translation with Word Class Models
  - [Sundermeyer & Alkhouli$^+$ 14] Translation Modeling with Bidirectional Recurrent Neural Networks
  - [Guta & Wuebker$^+$ 15] Extended Translation Models in Phrase-based Decoding
  - [Guta & Alkhouli$^+$ 15] A Comparison between Count and Neural Network Models Based on Joint Translation and Reordering Sequences
- Search
  - [Wuebker & Ney$^+$ 12b] Fast and Scalable Decoding with Language Model Look-Ahead for Phrase-based Statistical Machine Translation
- Training
  - [Wuebker & Mauser$^+$ 10] Training Phrase Translation Models with Leaving-One-Out
  - [Heger & Wuebker$^+$ 10b] A Combination of Hierarchical Systems with Forced Alignments from Phrase-Based Systems
  - [Wuebker & Ney 12a] Phrase Model Training for Statistical Machine Translation with Word Lattices of Preprocessing Alternatives
  - [Wuebker & Hwang$^+$ 12] Leave-One-Out Phrase Model Training for Large-Scale Deployment
  - [Peitz & Mauser$^+$ 12] Forced Derivations for Hierarchical Machine Translation
  - [Wuebker & Ney 13] Length-incremental Phrase Training for SMT

  - [Lehnen & Peter+ 13] (Hidden) Conditional Random Fields Using Intermediate Classes for Statistical Machine Translation
  - [Wuebker & Muehr+ 15] A Comparison of Update Strategies for Large-Scale Maximum Expected BLEU Training

- Software
  - [Wuebker & Huck+ 12] Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation

- International Evaluation Campaigns
  - [Heger & Wuebker+ 10a] The RWTH Aachen Machine Translation System for WMT 2010
  - [Mansour & Peitz+ 10] The RWTH Aachen Machine Translation system for IWSLT 2010
  - [Huck & Wuebker+ 11] The RWTH Aachen Machine Translation System for WMT 2011
  - [Freitag & Leusch+ 11] Joint WMT Submission of the QUAERO Project
  - [Feng & Schmidt+ 11] The RWTH Aachen System for NTCIR-9 PatentMT
  - [Wuebker & Huck+ 11] The RWTH Aachen Machine Translation System for IWSLT 2011
  - [Peitz & Mansour+ 12] The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012
  - [Feng & Schmidt+ 13] The RWTH Aachen System for NTCIR-10 PatentMT
  - [Peitz & Mansour+ 13] The RWTH Aachen Machine Translation System for WMT 2013
  - [Wuebker & Peitz+ 13a] The RWTH Aachen Machine Translation Systems for IWSLT 2013
  - [Freitag & Peitz+ 13] EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project
  - [Peitz & Wuebker+ 14] The RWTH Aachen German-English Machine Translation System for WMT 2014
  - [Freitag & Peitz+ 14] EU-BRIDGE MT: Combined Machine Translation
  - [Wuebker & Peitz+ 14] The RWTH Aachen Machine Translation Systems for IWSLT 2014
  - [Freitag & Wuebker+ 14] Combined Spoken Language Translation
  - [Peter & Toutounchi+ 15b] The RWTH Aachen German-English Machine Translation System for WMT 2015

- Other contributions
  - [Mansour & Wuebker+ 11] Combining Translation and Language Model Scoring for Domain-Specific Data Filtering
  - [Boudahmane & Buschbeck+ 11] Advances on Spoken Language Translation in the Quaero Program

- [Wuebker & Ney$^+$ 14] Comparison of Data Selection Techniques for the Translation of Video Lectures
- [Wuebker & Green$^+$ 15] Hierarchical Incremental Adaptation for Statistical Machine Translation

2. Scientific Goals

In this thesis, the following scientific goals are pursued:

- For many real-world applications of machine translation, e.g. interactive translation, speed is an important criterion. Within standard phrase-based beam search, the tradeoff between quality and speed can be adjusted through pruning parameters. In state-of-the-art decoders, language model computations tend to be among the most expensive and frequent operations during search. This work aims at increasing translation speed in general and improving the tradeoff factor between quality and speed by investigating techniques to specifically minimize the number of language model computations. We plan to achieve this by incorporating the language model score into the pruning process as early as possible.

- Data sparsity is one of the main problems of state-of-the-art translation models. Due to the large vocabulary size, many of the phrase pairs are observed very few times or only once in the training data, which results in unreliable probability estimates. To alleviate the data sparsity problem, previous work has leveraged word classes, e.g. for alignment templates [Och & Tillmann$^+$ 99, Och & Ney 04] or in the IBM-4 and IBM-5 word-based translation models [Brown & Della Pietra$^+$ 93]. These word classes can be trained in an unsupervised manner independent of the language, which makes them an appealing alternative to linguistic annotations like part-of-speech tags. In this thesis we plan to investigate, whether the standard models in a phrase-based decoder can be re-parameterized using word classes to obtain a smoother distribution with better generalization capabilities. This could be of particular interest for the language model, where the smaller vocabulary size allows modeling a longer n-gram context.

- The standard state-of-the-art phrase training procedure extracts the phrase pairs from a Viterbi word alignment. It has the following drawbacks. (i) It is based on heuristics that have no foundation in statistics. (ii) The word alignment is trained with models that are different from the models used in search. (iii) All extracted models are trained independently without taking their interactions into account. In [Wuebker & Mauser$^+$ 10] we addressed these drawbacks by proposing a generative phrase training algorithm based on force-aligning the training data. In this work we plan to further extend and develop this method and improve its performance. One main goal of this work is to entirely get rid of the dependency on a Viterbi word alignment for initialization to obtain an end-to-end generative approach to phrase model training that is consistent with the translation procedure and unbiased by a possibly erroneous word alignment.

- As an alternative to generative training approaches, this work aims at examining discriminative training techniques. The advantage of discriminative learning is that it allows the definition of very fine-grained features that can model arbitrary aspects of translation quality, which are then optimized with respect to a quality measure of choice. We plan to explore

several different update strategies for maximum expected BLEU training [He & Deng 12] and compare them in terms of efficiency and effectiveness.

- All developed techniques shall be evaluated on publicly available large-scale tasks, within research projects and in public evaluations. By comparison with internal results and external research groups it is possible to assess their effectiveness on state-of-the-art translation systems.

3. Preliminaries

In this chapter we will briefly introduce terminology, as well as several concepts that are used throughout this work but are not part of our contributions.

3.1 Terminology and Notation

As will be discussed in Section 3.2 the translation process is cast as a source-channel decoding problem [Weaver 55]. The program performing the search for the best translation given our underlying models is therefore named a *decoder*. Given a source sentence, the gold standard translation, which is usually either produced by professional translators or crawled from the web, will be called a *reference*. The translation output of our decoder is denoted as *hypothesis translation* or *hypothesis* for brevity. A collection of documents will be referred to as a corpus. A *bilingual corpus* or *bitext* is a corpus containing documents in two languages along with a bijective mapping from each source sentence to its corresponding gold standard translation on the target side.

For experimental evaluation, the collection of the available bitexts is split into a large portion used to train the models, called the *training data* (`train`), and one or several smaller *development sets* (`dev`) and *test sets* (`test`), used for parameter tuning and final evaluation, respectively. In order for the results to be meaningful and generalizable, it is important that `train`, `dev` and `test` are disjoint. Typically, `train` is crawled from the web or multilingual government sources (e.g. European Parliament speeches) and then sentence-aligned in an automatic fashion. `dev` and `test`, on the other hand, are often human-translated for this particular purpose.

Throughout this work, we will denote a source sentence of length J (i.e. containing J words) as $F = f_1^J = f_1 f_2 \dots f_J$, and a target sentence of length I as $E = e_1^I = e_1 e_2 \dots e_I$. Historically, f stands for *French* or *foreign* and e for *English*. Their length can be expressed as $|F| = |f_1^J| = J$ and $|E| = |e_1^I| = I$. A sequence of n words will be referred to as an *n-gram*. For $n = \{1, 2, 3\}$, n-grams are named *unigrams*, *bigrams* and *trigrams*.

We will also make extensive use of probability theory. In our notation, we will use $Pr(\cdot)$ to refer to the actual correct distributions, which are unknown, as opposed to the model distributions denoted as $p(\cdot)$.

3.2 Statistical Machine Translation

Following the idea by [Weaver 55], statistical machine translation (SMT) interprets the translation process as a source-channel problem as in communication theory. The source text is considered to be an encrypted version of the target text, which needs to be decrypted to obtain the translation. Applying Bayes' decision rule to this idea, we can formalize the translation procedure in the following way. Given a source sentence $f_1^J = f_1 f_2 \dots f_J$, the aim is to find the corresponding target sentence $e_1^I = e_1 e_2 \dots e_I$ which maximizes the posterior probability $Pr(e_1^I | f_1^J)$. With Bayes'

theorem, this can be decomposed into two separate knowledge sources, the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J|e_1^I)$ [Brown & Cocke$^+$ 90]:

$$f_1^J \rightarrow \hat{e}_1^{\hat{I}}(f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I|f_1^J) \right\} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \frac{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)}{Pr(f_1^J)} \right\} \tag{3.1}$$

$$= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} \tag{3.2}$$

Given this decision rule, the task of statistical machine translation can be split into three subproblems as described in [Ney 01], which will be addressed separately in this thesis:

- the **modeling problem**, i.e. how to structure the dependencies of source and target language sentences;
- the **search problem**, i.e. how to find the best translation candidate among all possible target language sentences;
- the **training problem**, i.e. how to estimate the free parameters of the models from the training data.

3.3 Log-linear Modeling

In most state-of-the-art machine translation systems, the source-channel approach described in the previous section is generalized to a log-linear model. In addition to the language and translation model, this allows us to incorporate an arbitrary number M of models $h_m(\cdot,\cdot), m = 1 \ldots M$, to directly express the posterior probability $Pr(e_1^I|f_1^J)$ [Papineni & Roukos$^+$ 98, Och & Ney 02]:

$$Pr(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right)}{\sum_{I',e'^{I'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J) \right)} \tag{3.3}$$

The architecture of a statistical machine translation system following this approach is illustrated in Figure 3.1. The different models $h_m(\cdot,\cdot)$ can capture separate aspects of the translation. The scaling factors λ_m are tuned discriminatively in order to differently weight the corresponding models according to their importance for the final decision. The most popular algorithm for optimizing the λ_m is minimum error rate training (MERT) [Och 03].

Similar to Equation 3.2, during search we can omit the denominator, which is constant with respect to the hypothesis translation, and the exponential function, resulting in a linear combination of models:

$$f_1^J \rightarrow \hat{e}_1^{\hat{I}}(f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I|f_1^J) \right\} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{3.4}$$

In typical state-of-the-art translation systems, several of the models $h_m(\cdot,\cdot)$ are estimated from a Viterbi word alignment, which will be discussed in the next section.

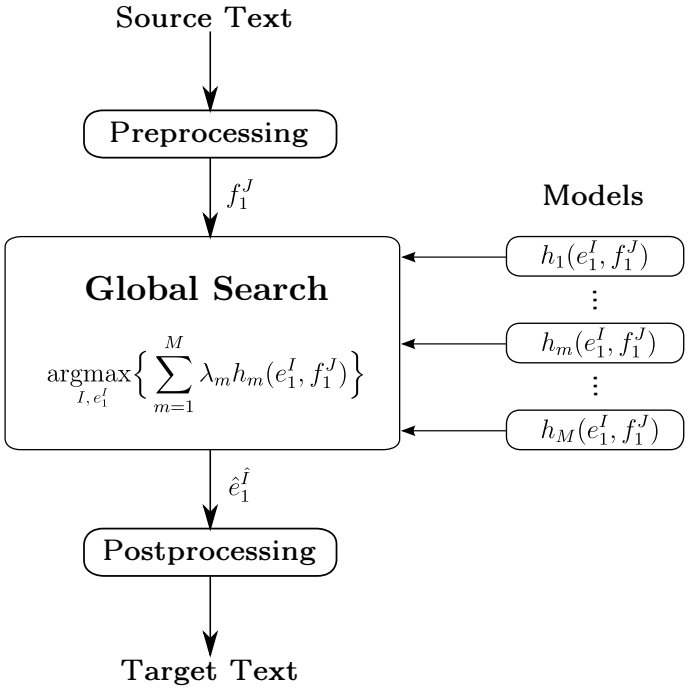


Figure 3.1: Illustration of a statistical machine translation system architecture following the log-linear modeling approach.

3.4 Word Alignment

The first step in training most current state-of-the-art statistical machine translation systems is word-aligning the bilingual training data. Given a source sentence and its translation in the target language, the word alignment describes the word-level correspondences between the two. Formally, a word alignment A is defined as a relation over source indices j and target indices i:

$$A \subseteq I \times J \tag{3.5}$$

For a sentence pair (e_1^I, f_1^J), e_i is aligned to f_j iff $(i, j) \in A$. An example word alignment is given in Figure 3.2.

A common tool for computing word alignments is GIZA++ [Och & Ney 03]. It makes use of the word-based translation models IBM-1 through IBM-5 proposed in [Brown & Della Pietra$^+$ 93] and the hidden Markov alignment model (HMM) [Vogel & Ney$^+$ 96]. We will now briefly sketch these models.

In the following we require a more constrained definition of word alignment. Here, each source word f_j is mapped to exactly one target word e_i, where we include the empty word e_0 on the target side to account for unaligned source words. The alignment is denoted as $a_1^J = a_1 \dots a_J$, where $a_j = i$ is the index of the target word corresponding to f_j.

The posterior translation probability of a source sentence f_1^J given a target sentence e_1^I under the alignment models IBM-1, IBM-2 and HMM is defined as follows.

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \tag{3.6}$$

$$= p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} p(a_j|a_1^{j-1}, J, I) \cdot p(f_j|e_{a_j}) \tag{3.7}$$

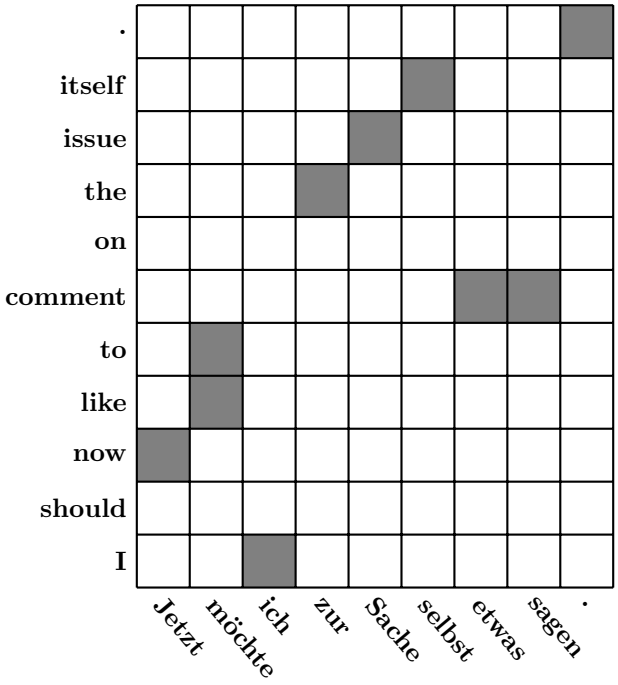


Figure 3.2: Word alignment example.

The posterior is factorized into a length model $p(J|I)$, an alignment model $p(a_j|a_1^{j-1}, J, I)$ and a lexicon model $p(f_j|e_{a_j})$. The models differ in their dependency assumptions of the aligment probability.

- **IBM-1** assumes a uniform distribution $p(a_j|a_1^{j-1}, J, I) = \frac{1}{I+1}$.
- **IBM-2** is also a zero-order model, which assumes a dependency on the source position: $p(a_j|a_1^{j-1}, J, I) = p(a_j|j, J, I)$. The result is a table with probabilities for each combination of (a_j, j, J, I).
- The **HMM** model makes a first-order hidden Markov assumption, which is also stored as a table: $p(a_j|a_1^{j-1}, J, I) = p(a_j|a_{j-1}, J, I)$.

IBM-3 through IBM-5 introduce the concept of *fertility* $\varphi(e_i)$, which is the number of source words aligned to the given target word e_i. Note that the source length J directly depends on the fertilities: $J = \sum_i \varphi(e_i)$.

- **IBM-3** is a re-parameterization of IBM-2 using fertilities, resulting in an inverted zero-order alignment model $p(j|a_j, J, I)$.
- **IBM-4** applied an inverted first-order alignment model $p(j|j', J, I)$, where j' is the source position aligned to the previous target word. This model is deficient as some probability mass is assigned to impossible alignments.
- **IBM-5** is the non-deficient version of IBM-4, which is modified to satisfy the normalization constraints.

The models are trained with the EM-algorithm [Dempster & Laird$^+$ 77]. In order to train our machine translation system, for each sentence pair (e_1^I, f_1^J) we extract a *Viterbi* alignment $\hat{a}_1^J$

defined by

$$(e_1^I, f_1^J) \rightarrow \hat{a}_1^J(e_1^I, f_1^J) = \underset{a_1^J}{\operatorname{argmax}}\{Pr(a_1^J|e_1^I, f_1^J)\} \tag{3.8}$$

$$= \underset{a_1^J}{\operatorname{argmax}}\{Pr(f_1^J, a_1^J|e_1^I)\} \tag{3.9}$$

In practice, the models are trained in sequence, initializing the translation table with the previous model. A typical setup is IBM-1→HMM→IBM-4, each being trained for five EM iterations. As the models are directional, this is done in both translation directions and the Viterbi alignments are afterwards combined using heuristics, resulting in a general undirected alignment as defined in Equation 3.5.

It should be noted that in most state-of-the-art translation systems the IBM models are discarded after the alignment step. The Viterbi alignment is needed as a resource for estimating the models that are actually used in search.

3.5 Evaluation Measures

Automatic evaluation of different MT systems is a difficult task. Given a set of hypothesis translations, even for human experts it can be hard to order them by quality. A meta-evaluation of a shared translation task was performed in [Callison-Burch & Fordyce$^+$ 08]. The authors show that human annotators disagreed on the ranking of two translation hypotheses in 42% of the cases. Additionally, human evaluation is expensive and intractable for a meaningful amount of data. In this work, we will evaluate our hypothesis translations using the Bleu and Ter measures, which are the most widely used in statistical machine translation research. They assess the quality of translation in an automatic fashion by providing a similarity score or an error rate with respect to one or several reference translations.

3.5.1 BLEU

The **Bi**lingual **E**valuation **U**nderstudy (Bleu) [Papineni & Roukos$^+$ 02] is an accuracy measure. It is computed as a combination of the geometric mean of n-gram precisions and a brevity penalty that penalizes short hypotheses. Given a reference translation $\hat{e}_1^{\hat{I}}$, the Bleu score for a hypothesis e_1^I can be expressed as follows:

$$\text{Bleu}(e_1^I, \hat{e}_1^{\hat{I}}) := BP(I, \hat{I}) \cdot \prod_{n=1}^{4} Prec_n(e_1^I, \hat{e}_1^{\hat{I}})^{\frac{1}{4}} \tag{3.10}$$

with

$$BP(I, \hat{I}) := \begin{cases} 1 & \text{if } I \geq \hat{I} \\ e^{(1-\frac{\hat{I}}{I})} & \text{if } I < \hat{I} \end{cases} \tag{3.11}$$

$$Prec_n(e_1^I, \hat{e}_1^{\hat{I}}) := \frac{\sum_{w_1^n} \min\left\{C(w_1^n|e_1^I), C(w_1^n|\hat{e}_1^{\hat{I}})\right\}}{\sum_{w_1^n} C(w_1^n|e_1^I)} \tag{3.12}$$

Here, $C(w_1^n|e_1^I)$ denotes the count, i.e. the number of occurrences, of an n-gram w_1^n in a sentence e_1^I. The denominator of Equation 3.12 evaluates to the total number of n-grams within the

hypothesis, i.e. $I - n + 1$. The BLEU score can be extended to take more than one reference translation into account. We apply document level BLEU, the n-gram counts being collected over the whole data set rather than considering single sentences.

3.5.2 TER

The **T**ranslation **E**dit **R**ate (TER) [Snover & Dorr$^+$ 06] is an error metric based on the Levenshtein distance [Levenshtein 66]. It counts the number of edits required to change a hypothesis into one of the reference translations. In addition to substitutions, deletions and insertions of single words as in the well-known word error rate (WER), it allows shifts, i.e. movements of contiguous word sequences within the hypothesis. All types of edits are assigned equal cost. With one or more reference translations, the number of edit operations is divided by the average number of reference words per sentence.

$$\text{TER}(e_1^I, \hat{e}_1^{\hat{I}}) = \frac{\#\text{ of edits}}{\text{avg. }\#\text{ of reference words}} \tag{3.13}$$

Unless stated otherwise, we report document level TER, where the edit counts are collected over the whole data set.

4. Phrase-Based Translation

This chapter is dedicated to the *modeling* problem (cf. Section 3.2). I aim at giving a complete account of the phrase-based statistical machine translation paradigm, focusing on the different model components. Large parts of this description were published previously [Zens 08]. Our own contributions are integrated into this chapter side by side with previous work, in order to provide a clear and consistent overall picture. The final two Sections 4.3 and 4.4 will disambiguate the novel contributions of this work from previous publications.

4.1 General Approach

With the introduction of the alignment template approach [Och & Tillmann$^+$ 99, Och & Ney 04], statistical machine translation experienced a leap in performance. It served as inspiration for the phrase-based translation paradigm [Koehn & Och$^+$ 03], which was quickly adopted as the new state of the art in statistical machine translation research, replacing the word-based approach. It is still the most widely used design, particularly for commercial systems that require computationally efficient solutions.

The idea of phrase-based translation is to segment the given source sentence into contiguous sequences of words, so-called phrases. The individual source phrases are transferred into the target language by a lookup in the phrase translation table, which in addition to a number of candidate target phrases contains associated model probabilities or scores. The target phrases can be rearranged to produce fluent target language output. This process is illustrated in Figure 4.1 and the resulting system architecture in Figure 4.2. In practice, these three steps are not performed in sequence, but are interleaved. The details will be described in Section 5. By using entire phrases rather than single words as the basic translation units, local context is taken into account directly in the phrase lexicon model, resulting in more meaningful translations.

For a sentence pair (f_1^J, e_1^I), we formally define a segmentation s_1^K into K phrase pairs as follows.

$$k \rightarrow s_k := (i_k; b_k, j_k), \text{ for } k = 1, \ldots, K \tag{4.1}$$

$$i_0 := 0 \tag{4.2}$$

$$j_0 := 0 \tag{4.3}$$

$$i_{K+1} := I + 1 \tag{4.4}$$

$$b_{K+1} := J + 1 \tag{4.5}$$

Here, the last position of the kth target phrase is denoted as i_k. The start and end positions of the source phrase that is aligned to the kth target phrase are denoted as b_k and j_k, respectively. We require that there are no gaps and no overlap, i.e. all source and all target words have to be covered by exactly one phrase. Under this segmentation we define the bilingual phrase pairs

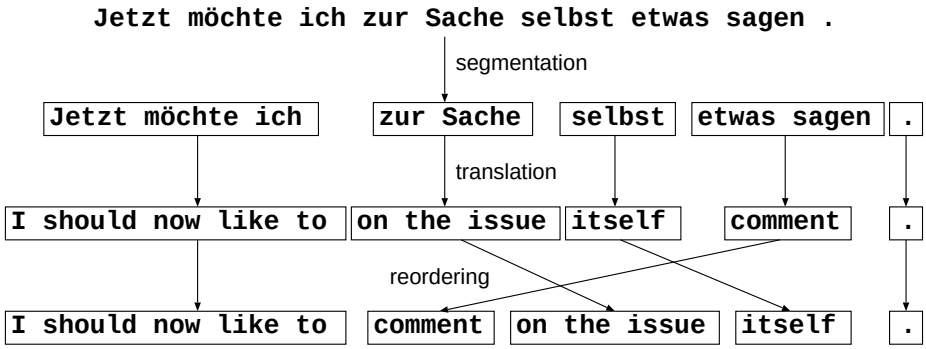


Figure 4.1: Illustration of the phrase-based translation approach.

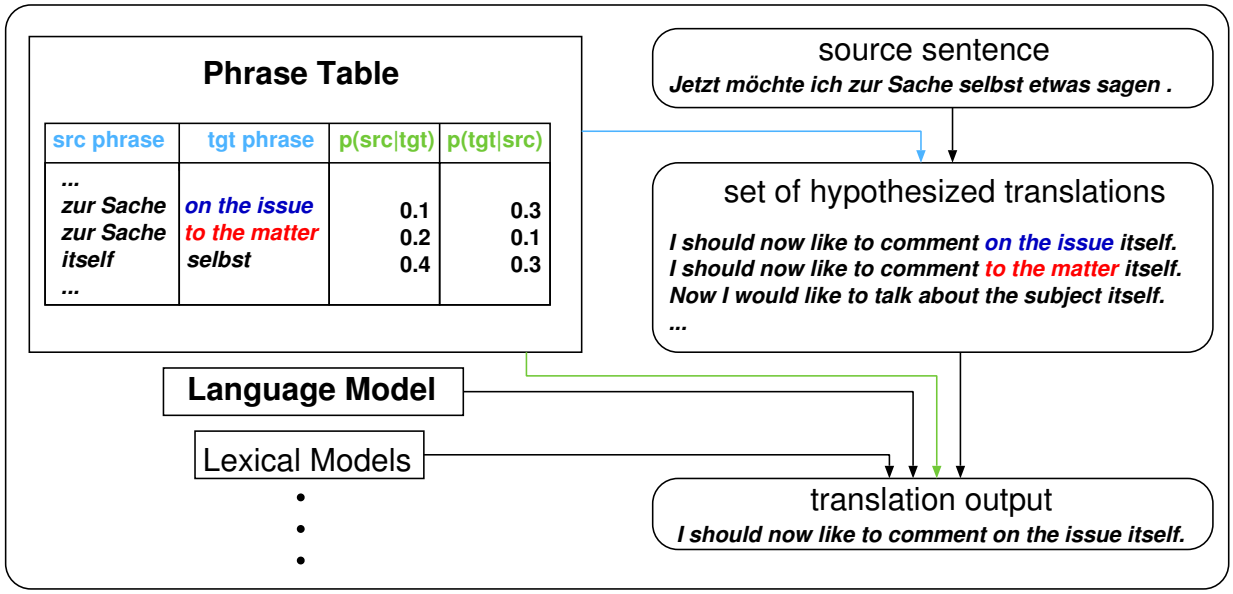


Figure 4.2: Architecture of a phrase-based translation system.

$(\tilde{e}_k, \tilde{f}_k)$ by

$$\tilde{e}_k := e_{i_{k-1}+1} \dots e_{i_k} \tag{4.6}$$

$$\tilde{f}_k := f_{b_k} \dots f_{j_k} \tag{4.7}$$

This definition explicitly contains the phrase-level reordering information. The different variables of this phrase segmentation are illustrated in Figure 4.3.

The phrase segmentation s_1^K is introduced as a hidden variable into the log-linear model from

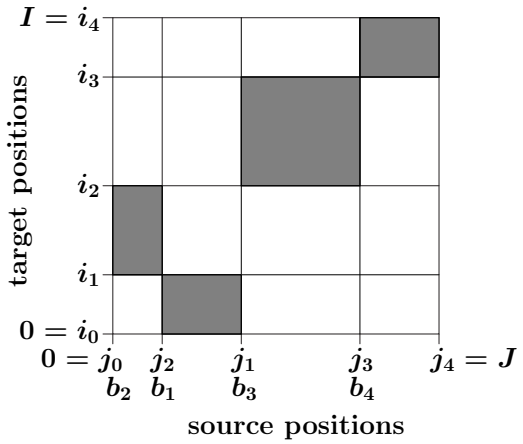


Figure 4.3: Illustration of the phrase segmentation. Taken from [Zens 08].

Equation 3.3:

$$Pr(e_1^I|f_1^J) = \sum_{K,s_1^K} Pr(e_1^I, s_1^K|f_1^J) \tag{4.8}$$

$$= \sum_{K,s_1^K} \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J)\right)}{\sum_{I',e'^{I'}_1,K',s'^{K'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, s'^{K'}_1, f_1^J)\right)} \tag{4.9}$$

$$\approx \max_{K,s_1^K} \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J)\right)}{\sum_{I',e'^{I'}_1,K',s'^{K'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, s'^{K'}_1, f_1^J)\right)} \tag{4.10}$$

Rather than carrying out the full sum over all segmentations s_1^K, in practice we apply the maximum approximation for increased efficiency.

4.2 Model Definition

In this Section we will describe the models $h_m(e_1^I, s_1^K, f_1^J)$ (cf. Equation 4.10) that are being used in this work. This addresses the *modeling* problem, as stated in Section 3.2.

4.2.1 Phrase Translation Model

The standard phrase-based translation model, in the literature also often referred to as the phrasal channel model, assigns probability estimates $p(\tilde{f}|\tilde{e})$ and $p(\tilde{e}|\tilde{f})$ to each phrase pair. They are usually computed as relative frequencies from heuristically extracted counts (the extraction process

will be described in Section 6.1):

$$p(\tilde{f}|\tilde{e}) = \frac{N(\tilde{f},\tilde{e})}{N(\tilde{e})}, \tag{4.11}$$

where $N(\tilde{f},\tilde{e})$ is the joint count of the phrase pair $(\tilde{f},\tilde{e})$ and $N(\tilde{e})$ the marginal count of the target phrase $\tilde{e}$. The inverse model $p(\tilde{e}|\tilde{f})$ is computed analogously. The feature functions $h_{Phr}(\cdot,\cdot,\cdot)$ and $h_{iPhr}(\cdot,\cdot,\cdot)$ are composed of the direct or inverse translation probabilities of the individual phrase pairs under segmentation s_1^K:

$$h_{Phr}(e_1^I, s_1^K, f_1^J) = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) = \sum_{k=1}^{K} \log p(\tilde{f}_k|\tilde{e}_k) \tag{4.12}$$

$$h_{iPhr}(e_1^I, s_1^K, f_1^J) = \log \prod_{k=1}^{K} p(\tilde{e}_k|\tilde{f}_k) = \sum_{k=1}^{K} \log p(\tilde{e}_k|\tilde{f}_k) \tag{4.13}$$

Note that the model scores are additive with respect to the individual phrase pairs and do not require any context outside of the phrases. This is why we call this a *local* model, which will become relevant in the search algorithm (cf. Section 5).

4.2.2 Word Lexicon Model

The word lexicon model is an effective smoothing model inspired by IBM-1, but operates on the phrase level rather than the sentence level. It makes use of lexical translation tables $p(f|e)$ and $p(e|f)$, which are estimated as relative frequencies computed from alignment counts in the Viterbi alignment. All combinations of one source and one target word are taken into account, including the empty words e_0 and f_0, which are used to model unaligned words on the opposite side. This model is used in both translation directions.

$$h_{Lex}(e_1^I, s_1^K, f_1^J) = \log \prod_{k=1}^{K} \prod_{j=b_k}^{j_k} \left(p(f_j|e_0) + \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \right) \tag{4.14}$$

$$= \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \log \left(p(f_j|e_0) + \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \right) \tag{4.15}$$

$$h_{iLex}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{i=i_{k-1}+1}^{i_k} \log \left(p(e_i|f_0) + \sum_{j=b_k}^{j_k} p(e_j|f_i) \right) \tag{4.16}$$

4.2.3 Heuristic Length Models

Two simple heuristics are introduced to control for average phrase and sentence length, namely the word penalty (WP) and the phrase penalty (PP).

$$h_{WP}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} |\tilde{e}_k| = I \tag{4.17}$$

$$h_{PP}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} 1 = K \tag{4.18}$$

4.2.4 Count-Based Smoothing Models

Rare phrase pairs are often over-estimated by simple count models. In order to penalize them, we apply indicator features according to a threshold τ.

$$h_{C,\tau}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} h_\tau(\tilde{f}_k, \tilde{e}_k) \tag{4.19}$$

$$h_\tau(\tilde{f}, \tilde{e}) := \begin{cases} 1 & N(\tilde{f}, \tilde{e}) \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

Typically, we use three models for $\tau = \{1, 2, 3\}$.

In more recent experiments, these threshold indicators are replaced by a single *enhanced low frequency* feature [Chen & Kuhn$^+$ 11], which we found to be equally effective and more stable under multiple MERT runs. It is defined as

$$h_{elf}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \frac{1}{N(\tilde{f}_k, \tilde{e}_k)}. \tag{4.21}$$

4.2.5 n-gram Language Model

The language model can be derived from the original source-channel approach to statistical machine translation (cf. Equation 3.2). Its purpose is to provide a distribution $Pr(e_1^I)$ over the target language. The standard n-gram language models applied in this work model this distribution as a Markov chain with order $n-1$, which results in the formulation

$$h_{LM}(e_1^I, s_1^K, f_1^J) = \log \prod_{i=1}^{I+1} p(e_i | e_{i-n+1}^{i-1}) = \sum_{i=1}^{I+1} \log p(e_i | e_{i-n+1}^{i-1}). \tag{4.22}$$

To add a sentence end probability, we define e_{I+1} as the sentence boundary marker. We apply interpolated modified Kneser-Ney discounting [Kneser & Ney 95, Chen & Goodman 98] for smoothing and to retain probability mass for unseen events. They are estimated using either the SRILM toolkit [Stolcke 02] or `lmplz` [Heafield & Pouzyrevsky$^+$ 13]. Note that this is a non-local model, i.e. it uses context information that goes beyond the phrase boundaries.

4.2.6 Recurrent Neural Network Language Model

On some experiments we additionally employ a neural network based language model. It is used in a rescoring step on n-best lists. We apply a recurrent long-short term memory (LSTM) architecture as described in [Sundermeyer & Ney$^+$ 15] using the *rwthlm* toolkit presented in [Sundermeyer & Schlüter$^+$ 14]. The recurrent formulation has the advantage of a theoretically unbounded history, while the LSTM nodes are designed to counteract the vanishing gradient effect. The network architecture is illustrated in Figure 4.4.

In the input layer, the words of the vocabulary are represented in 1-of-n encoding. The vocabulary typically consists of all non-singleton words from the training data plus an unknown token. The output layer makes use of a softmax activation function for normalization and is class-factored for increased computational efficiency. The word classes are trained in an unsupervised manner using the `mkcls` tool [Och 99]. Formally, we can define the model as

$$h_{rnnLM}(e_1^I, s_1^K, f_1^J) = \sum_{i=1}^{I+1} \log p_{rnn}(e_i | e_1^{i-1}). \tag{4.23}$$

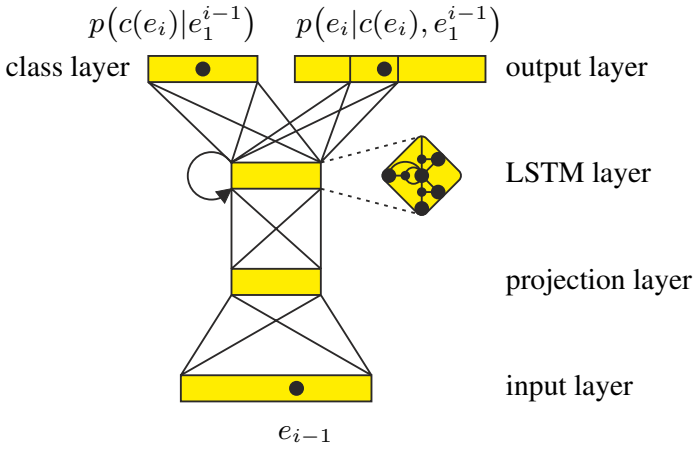


Figure 4.4: Illustration of the recurrent neural network (RNN) language model applied in this work. $c(e)$ is defined as the class of word e. Taken from [Sundermeyer & Schlüter$^+$ 14].

4.2.7 Distortion Model

The phrase-based translation model implicitly captures local reordering phenomena. When it comes to global reordering of the phrase pairs, it is usually a good idea to place a prior over the reorderings that prefers monotonous translation. A simple heuristic distance penalty is widely used in state-of-the-art machine translation systems.

$$h_{Dist}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K+1} q_{Dist}(b_k, j_{k-1}) \tag{4.24}$$

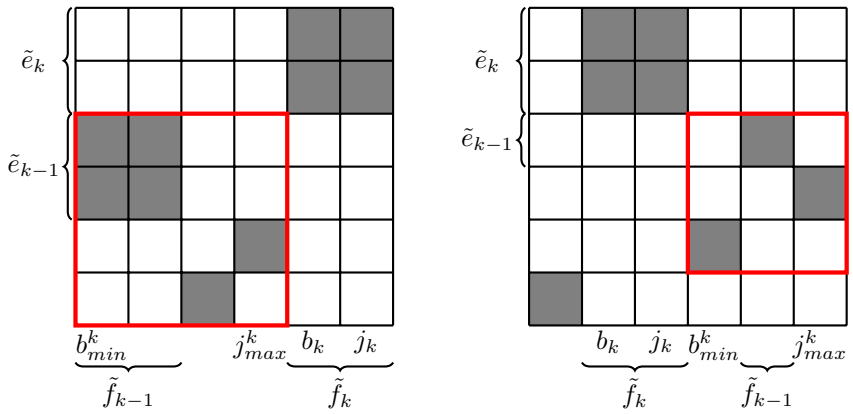
with

$$q_{Dist}(j, j') := \begin{cases} |j - j' + 1| & \text{if } |j - j' + 1| < D \\ |j - j' + 1|^2 & \text{otherwise} \end{cases} \tag{4.25}$$

In this work, we apply a soft upper limit D for the jump width. This is also a non-local model.

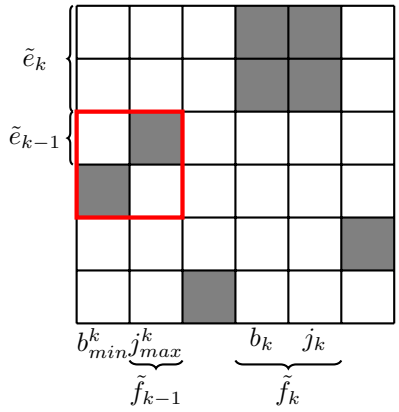
4.2.8 Hierarchical Reordering Model

The hierarchical reordering model (HRM) which we apply in this work was originally proposed by [Galley & Manning 08] and refined by [Cherry & Moore$^+$ 12]. Similar to models introduced in previous work [Tillmann 04, Koehn & Hoang$^+$ 07b], the orientation of a phrase pair is classified into three orientation classes: monotone (M), swap (S) and discontinuous (D). However, instead of determining the orientation with respect to the previous phrase only, the entire previous translation is taken into account. Phrase pairs can be merged into a *block*, if it does not violate the alignment, i.e. every phrase pair is either completely inside or outside the block. The hierarchical reordering model determines the orientation with respect to the largest block containing the previous phrase, but not the current phrase. The three orientation classes are illustrated in Figure 4.5. When translating the k-th phrase $f_k = f_{b_k} \dots f_{j_k}$ with $s_k = (i_k; b_k, j_k)$, we define b_{min}^k and j_{max}^k as the start and end positions on the source side of the largest block under the previous



(a) monotone

(b) swap



(c) discontinuous

Figure 4.5: The three orientations M, S and D for the hierarchical reordering model. The thick red rectangle indicates the largest previous block when translating the k-th phrase $\tilde{f}_k$.

segmentation decisions s_0^{k-1}. The orientation can be formalized as

$$O_{HRM}(s_k|s_0^{k-1}) = \begin{cases} M & \text{if } b_k = j^k_{max} + 1 \\ S & \text{if } j_k = b^k_{min} - 1 \\ D & \text{otherwise} \end{cases} \tag{4.26}$$

In search, the blocks can be computed using a shift-reduce parser. The orientation can also be defined in inverse direction, i.e. by parsing backwards, from the end to the beginning of the target sentence. We denote the inverse orientation as $O_{iHRM}(s_k|s_{k+1}^{K+1})$ and obtain the following models.

$$h_{HRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \log p(O_{HRM}(s_k|s_0^{k-1})|\tilde{f}_k, \tilde{e}_k) \tag{4.27}$$

$$h_{iHRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \log p(O_{iHRM}(s_k|s_{k+1}^{K+1})|\tilde{f}_k, \tilde{e}_k) \tag{4.28}$$

In practice, we apply separate weights λ_m for the three orientations, which has proven slightly more effective. Altogether, this results in six orientation models that are added to the log-linear

combination. For more details on our implementation of this model please refer to [Rietig 13]. The hierarchical reordering model is a non-local model.

4.2.9 Class-Based Smoothing Models

Most of the standard models described in this section suffer from sparsity issues. For example, for most phrase pairs, only few training instances are observed, which results in unreliable probability estimates. A possible way to reduce the sparsity in model estimation is using a smaller vocabulary. By clustering the vocabulary into a fixed number of word classes, we can train models that are less prone to sparsity issues. As proposed in [Wuebker & Peitz$^+$ 13b] we re-parameterize several of the standard models described in this chapter. Instead of word identities, we estimate the models based on word classes trained with the `mkcls` tool [Och 99]. The idea is that this should lead to a smoother distribution, which is more reliable due to less sparsity. We denote the class of a given word e as $c(e)$ and define $c(e_i^{i'}) := c(e_i)c(e_{i+1})\dots c(e_{i'})$. By replacing the word identities by word classes in the phrase translation model (cf. Section 4.2.1), the word lexicon model (cf. Section 4.2.2), the hierarchical reordering model (cf. Section 4.2.8) and the n-gram language model (cf. Section 4.2.5), we obtain the following models.

$$h_{wc\_Phr}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \log p(c(\tilde{f}_k)|c(\tilde{e}_k)) \tag{4.29}$$

$$h_{wc\_Lex}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \log \left(p(c(f_j)|e_0) + \sum_{i=i_{k-1}+1}^{i_k} p(c(f_j)|c(e_i)) \right) \tag{4.30}$$

$$h_{wc\_HRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \log p(O_{HRM}(s_k|s_0^{k-1})|c(\tilde{f}_k), c(\tilde{e}_k)) \tag{4.31}$$

$$h_{wc\_LM}(e_1^I, s_1^K, f_1^J) = \sum_{i=1}^{I+1} \log p(c(e_i)|c(e_{i-n+1}^{i-1})) \tag{4.32}$$

The inverse direction models $h_{wc\_iPhr}(\cdot,\cdot,\cdot)$, $h_{wc\_iLex}(\cdot,\cdot,\cdot)$ and $h_{wc\_iHRM}(\cdot,\cdot,\cdot)$ are obtained analogously. In addition to reduced sparsity, applying this type of smoothing to the language model has the advantage that higher n-gram orders can be modeled efficiently.

Equivalence to Factored Translation.
[Koehn & Hoang 07a] propose *factored translation models*, which integrate different levels of annotation (e.g. morphological analysis) as *factors* into the translation process. The authors analyze the surface form of the source word to produce the factors, which are then translated and in a final step, the surface form of the target word is generated from the target factors. Despite the fact that the translations of the factors operate on the same phrase segmentation, they are assumed to be independent. In practice this is done by *phrase expansion*, where the cross product from the phrase tables of the individual factors is computed to generate a joint phrase table.

In contrast to their approach, in this work each word is mapped to a single class. This means that once a translation option for the surface form is selected, the target side on the word class level is predetermined. As a result, no phrase expansion or generation steps are necessary to incorporate the word class information. We simply extend the phrase with additional scores and keep the set of phrases constant.

Although the implementation is simpler, our approach is mathematically equivalent to a special case of the factored translation framework (see Figure 4.6). The generation step from the

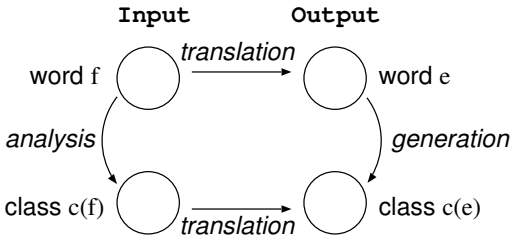


Figure 4.6: The factored translation model equivalent to our approach. The generation step assigns all probability mass to a single event: $p_{gen}(c(e)|e) = 1$.

target word e to its word class $c(e)$ assigns all probability mass to a single event:

$$p_{gen}(c|e) = \begin{cases} 1 \text{ if } c = c(e) \\ 0 \text{ otherwise} \end{cases} \tag{4.33}$$

4.2.10 Discriminative Models

In Section 6.3, we will describe discriminative training techniques that can handle arbitrary features. We introduce a number of sparse feature sets, including several that are discriminative re-parameterizations of models described above: A discriminative phrase table (dPhr) with features $\varphi_{Phr}(\tilde{f}, \tilde{e})$, phrase-internal bilingual word pair features $\varphi_{Lex}(f, e)$ (dLex), and source and target triplet features $\varphi_{sTrip}(f, f', e)$ and $\varphi_{tTrip}(f, e, e')$ (sTrip, tTrip). Triplet features [Hasan & Ganitkevitch$^+$ 08] assign a score to all word triples within a phrase pair that consist of either two source words and one target word (source triplets) or one source and two target words (target triplets). They are invariant with respect to the word order, i.e. $\varphi_{sTrip}(f, f', e) = \varphi_{sTrip}(f', f, e)$ and $\varphi_{tTrip}(f, e, e') = \varphi_{tTrip}(f, e', e)$. We further make use of a discriminative version of the hierarchical orientation model in both directions (dHRM, diHRM) with features $\varphi_{HRM}(O, \tilde{f}, \tilde{e})$ and $\varphi_{iHRM}(O, \tilde{f}, \tilde{e})$ and a re-parameterized version defined on boundary words (dBoundHRM and diBoundHRM) with features $\varphi_{boundHRM}(O, g, t)$ and $\varphi_{iboundHRM}(O, g, t)$ which has proven effective in [Cherry 13]. Here, g denotes a single word from either the source or target vocabulary, and $t \in \mathcal{T} := \{$*first source position, last source position, first target position, last target position*$\}$ its position within the phrase pair $(\tilde{f}, \tilde{e})$. We define the function $T(t, \tilde{f}, \tilde{e}) = g$ to select the word g at position t from $(\tilde{f}, \tilde{e})$.

$$h_{dPhr}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{Phr}(\tilde{f}_k, \tilde{e}_k) \tag{4.34}$$

$$h_{dLex}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} \varphi_{Lex}(f_j, e_i) \tag{4.35}$$

$$h_{sTrip}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k-1} \sum_{j'=j+1}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} \varphi_{sTrip}(f_j, f_{j'}, e_i) \tag{4.36}$$

$$h_{tTrip}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k-1} \sum_{i'=i+1}^{i_k} \varphi_{tTrip}(f_j, e_i, e_{i'}) \tag{4.37}$$

$$h_{dHRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{HRM}(O_{HRM}(s_k|s_0^{k-1}), \tilde{f}_k, \tilde{e}_k) \tag{4.38}$$

$$h_{diHRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{iHRM}(O_{iHRM}(s_k|s_{k+1}^{K+1}), \tilde{f}_k, \tilde{e}_k) \tag{4.39}$$

$$h_{dBoundHRM}(e_1^I, s_1^K, f_1^J) = \sum_{t\in\mathcal{T}} \sum_{k=1}^{K} \varphi_{HRM}(O_{HRM}(s_k|s_0^{k-1}), T(t, \tilde{f}_k, \tilde{e}_k), t) \tag{4.40}$$

$$h_{diBoundHRM}(e_1^I, s_1^K, f_1^J) = \sum_{t\in\mathcal{T}} \sum_{k=1}^{K} \varphi_{iHRM}(O_{iHRM}(s_k|s_{k+1}^{K+1}), T(t, \tilde{f}_k, \tilde{e}_k), t) \tag{4.41}$$

Similar to the standard non-discriminative models, these features can be defined on word classes rather than word identities:

$$h_{wc\_dPhr}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{Phr}(c(\tilde{f}_k), c(\tilde{e}_k)) \tag{4.42}$$

$$h_{wc\_dLex}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} \varphi_{Lex}(c(f_j), c(e_i)) \tag{4.43}$$

$$h_{wc\_sTrip}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k-1} \sum_{j'=j+1}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} \varphi_{sTrip}(c(f_j), c(f_{j'}), c(e_i)) \tag{4.44}$$

$$h_{wc\_tTrip}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k-1} \sum_{i'=i+1}^{i_k} \varphi_{tTrip}(c(f_j), c(e_i), c(e_{i'})) \tag{4.45}$$

$$h_{wc\_dHRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{HRM}(O_{HRM}(s_k|s_0^{k-1}), c(\tilde{f}_k), c(\tilde{e}_k)) \tag{4.46}$$

$$h_{wc\_diHRM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \varphi_{iHRM}(O_{iHRM}(s_k|s_{k+1}^{K+1}), c(\tilde{f}_k), c(\tilde{e}_k)) \tag{4.47}$$

$$h_{wc\_dBoundHRM}(e_1^I, s_1^K, f_1^J) = \sum_{t\in\mathcal{T}} \sum_{k=1}^{K} \varphi_{HRM}(O_{HRM}(s_k|s_0^{k-1}), c(T(t, \tilde{f}_k, \tilde{e}_k)), t) \tag{4.48}$$

$$h_{wc\_diBoundHRM}(e_1^I, s_1^K, f_1^J) = \sum_{t\in\mathcal{T}} \sum_{k=1}^{K} \varphi_{iHRM}(O_{iHRM}(s_k|s_{k+1}^{K+1}), c(T(t, \tilde{f}_k, \tilde{e}_k)), t) \tag{4.49}$$

After discriminative training, all features of the same type are condensed into a single model within the log-linear feature combination [Auli & Galley+ 14, Wuebker & Muehr+ 15].

4.3 Contributions

The implementation of the hierarchical reordering model (cf. Section 4.2.8) was performed under the author's supervision during the course of a Bachelor thesis [Rietig 13]. In terms of novel scientific contributions, the author is responsible for the idea and implementation of the class-based smoothing models (cf. Section 4.2.9) [Wuebker & Peitz+ 13b]. The author also performed the experimentation on the phrase-based systems, while the experiments on hierarchical phrase-based translation are due to Stephan Peitz. The implementation was later refined for better usability

and efficiency by Yunsu Kim as part of his Master Thesis [Kim 15]. The discriminative models given in Section 4.2.10 were also developed by the author of this thesis [Wuebker & Muehr$^+$ 15], with the exception of the phrasal (Equation 4.34), lexical (Equation 4.35) and triplet (Equations 4.36 and 4.37) models [Wuebker & Muehr$^+$ 15], which were part of Sebastian Mühr's Master thesis [Muehr 15] under the author's supervision.

4.4 Related Work

The word class models are equivalent to a special case of the factored translation paradigm [Koehn & Hoang 07a], as is described in Section 4.2.9. Training an additional language model for translation based on word classes has been previously proposed in [Wuebker & Huck$^+$ 12, Mediani & Zhang$^+$ 12, Koehn & Hoang 07a] Also related to our work, [Cherry 13] proposes to parameterize a hierarchical reordering model with sparse features that are conditioned on word classes trained with `mkcls`. However, the features are trained with MIRA rather than estimated by relative frequencies.

Finally, the hierarchical reordering model by [Galley & Manning 08] (cf. Section 4.2.8) can be extended to be applied in a hierarchical phrase-based decoder [Chiang 07] as described in [Huck & Wuebker$^+$ 13].

5. Search

The goal of this chapter is to tackle the *search* problem (cf. Section 3.2) by giving a detailed description of the current state of the art. As in the previous chapter, the novel contributions are intertwined with previously published work [Zens 08] to obtain a complete specification of the algorithm. Again, the final two Sections 5.5 and 5.6 will clarify which parts are our own contributions.

5.1 The Search Graph

When a phrase-based translation system is presented with a source sentence that is to be translated, the search procedure is invoked to produce the output in the target language. Its goal is to find the best candidate translation under the Bayes decision rule. We extend Equation 3.4 with the hidden phrase alignment variable s_1^K introduced in Equation 4.1 and apply the maximum approximation:

$$\begin{aligned} f_1^J \rightarrow \hat{e}_1^{\hat{I}}(f_1^J) &= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I|f_1^J) \right\} \\ &= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \sum_{K,s_1^K} Pr(e_1^I, s_1^K|f_1^J) \right\} \\ &\approx \underset{I,e_1^I,K,s_1^K}{\operatorname{argmax}} \left\{ Pr(e_1^I, s_1^K|f_1^J) \right\} \\ &= \underset{I,e_1^I,K,s_1^K}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \end{aligned} \tag{5.1}$$

As enumerating all possible sentences in the target language is infeasible, we have to resort to further approximations. To solve this, we apply the beam search algorithm [Jelinek 97], that is well suited for this problem. It extends dynamic programming [Bellman 57] with pruning, resulting in polynomial rather than exponential time and space efficiency, at the expense of possibly non-optimal solutions. We implement a specialized version of beam search, the *source cardinality synchronous search* algorithm described in [Zens & Ney 08] with the extensions presented by [Wuebker & Ney$^+$ 12b].

As described in [Zens 08], during search we have to take decisions on:

- the number K of phrases
- the segmentation of the source sentence into K phrases
- the permutation (or reordering) of the phrases

- the phrasal translation $\tilde{e}$ for each source phrase $\tilde{f}$

Note that both the segmentation of the source sentence and the permutation of the phrases is encoded in the variable s_1^K defined in Equation 4.1. We interpret the search procedure for translating the source sentence f_1^J as a sequence of decisions $(\tilde{e}_k, b_k, j_k)$ for $k = 1, \ldots, K$. At each step we decide on a source phrase $\tilde{f}_k = f_{b_k} \ldots f_{j_k}$ and the corresponding translation $\tilde{e}_k$. To avoid gaps and overlap, we keep track of the set of source positions that are already translated, which we denote as the *coverage set* $C \subseteq \{1, ..., J\}$. Following the dynamic programming principle, the search space is represented as a graph where the arcs are labeled with the decisions $(\tilde{e}_k, b_k, j_k)$ and the nodes are labeled with the coverage sets C. The nodes can also be interpreted as partial hypothesis translations. The initial node is labeled with the empty set and the goal node with the full coverage $C_{full} = \{1, \ldots, J\}$. Each path through this graph represents a possible translation of f_1^J, which can be recovered by concatenating the target phrases $\tilde{e}_k$ of the individual decisions. The target sentence is therefore generated in a monotonous left-to-right fashion. The same translation can be produced by several paths representing different phrase segmentations.

Some of the models $h_m(e_1^I, s_1^K, f_1^J)$ depend on previous decisions, which is why we refer to those as *non-local* models. In this work, these include one or multiple n-gram language models, the distortion model, the hierarchical reordering model, as well as their class-based and discriminative re-parameterizations. An n-gram language model depends on the previous $(n-1)$ words, called the language model history, and the distortion model on the end position of the previous source phrase. The hierarchical reordering model additionally requires the top of the stack of the shift-reduce-parser and the start position of the previous phrase. Therefore, the nodes are additionally labeled with this information, which we subsume in a single model state variable χ for simplicity. Labeling the nodes with states allows us to perform *recombination*. When several partial hypotheses have outgoing arcs ending in the same node, they can be recombined. Even though their target translations may be different, they are indistinguishable with respect to all future decisions and their scores. We only need to retain backpointers to the best scoring incoming arc in order to be able to reconstruct the translation in the end via backtracking. The initial state is denoted as χ_0. Nodes are identified by the tuple (C, χ), where C is the coverage set and χ the model state. Each node (C, χ) is assigned a score $Q(C, \chi)$. We compute the scores of the nodes in the order of their source cardinality $|C|$ by hypothesis expansion. Expanding (C', χ') with the decision $(\tilde{e}, b, j)$ yields the successor state

$$(C, \chi) = (C', \chi') \oplus (\tilde{e}, b, j) := (C' \cup \{b, \ldots, j\}, \chi' \oplus (\tilde{e}, b, j)). \tag{5.2}$$

Here, $\chi' \oplus (\tilde{e}, b, j)$ denotes the resulting new model state when expanding χ' with $(\tilde{e}, b, j)$, which depends on the particular models being used. To avoid overlap we have to ensure that $C \cap \{b, \ldots, j\} = \emptyset$. The score of the new state evaluates to

$$Q(C, \chi) = \max_{\substack{(C', \chi'), (\tilde{e}, b, j): \\ C' \cap \{b, \ldots, j\} = \emptyset, \\ (C', \chi') \oplus (\tilde{e}, b, j) = (C, \chi)}} \Big\{ Q(C', \chi') + q_{\mathrm{TM}}(\tilde{e}, b, j) + q_{\mathrm{LM}}(\tilde{e}|\chi') + q_{\mathrm{Dist}}(b|\chi') + q_{\mathrm{HRM}}(\tilde{e}, b, j|\chi') \Big\}. \tag{5.3}$$

Here, we subsume the scores of all local models that do not depend on the state (C', χ') in the overall *translation model score* $q_{\mathrm{TM}}(\tilde{e}, b, j)$. This includes the phrase translation models, the word lexicon models, the heuristic length models, the count-based smoothing models as well as their class-based and discriminative re-parameterizations. $q_{\mathrm{LM}}(\tilde{e}|\chi')$ denotes the language model score, $q_{\mathrm{Dist}}(b|\chi')$ the distortion model score and $q_{\mathrm{HRM}}(\tilde{e}, b, j|\chi')$ the hierarchical orientation model score. Note that with the exception of the language model, all models described in Section 4.2 are defined as a sum over the individual phrase pairs, or decisions $(\tilde{e}, b, j)$. The model scores

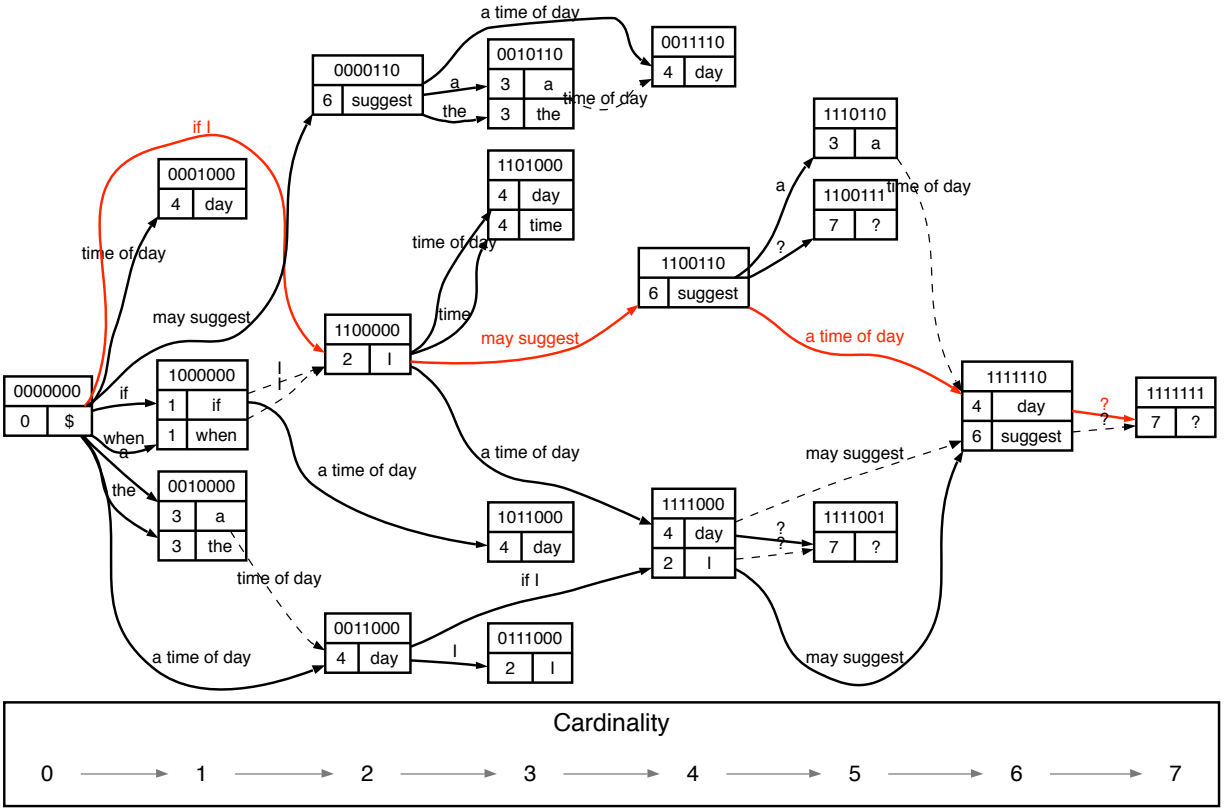


Figure 5.1: Illustration of the search graph. German input sentence: "Wenn ich eine Uhrzeit vorschlagen darf?". English translation: "If I may suggest a time of day?" The nodes are labeled with the coverage C (as a bit vector) and the model state χ, here containing the end position of the last phrase and the bigram language model history. Partial hypotheses are organized by their cardinality. Dashed edges are recombined and the best path is marked red. Scores are omitted. Taken from [Zens 08].

$q_{\mathrm{TM}}(\tilde{e}, b, j)$, $q_{\mathrm{Dist}}(b|\chi')$ and $q_{\mathrm{HRM}}(\tilde{e}, b, j|\chi')$ are simply the values of the individual summands corresponding to the current decision $(\tilde{e}, b, j)$, multiplied with their respective model weight λ_m. As the translation is generated in target order, the language model score (cf. Equation 4.22) can easily be reformulated in the same manner to obtain $q_{\mathrm{LM}}(\tilde{e}|\chi')$. The maximum in Equation 5.3 is computed over all incoming arcs and is thus a direct result of recombination. The search graph is illustrated in Figure 5.1. For simplicity, in this example the information in state χ is limited to the 2-gram language model history and the end position of the last translated phrase.

5.2 Pruning

The search problem as described above is NP-hard [Knight 99]. To achieve feasibility we apply beam search, i.e. we introduce pruning into the dynamic programming algorithm. The idea is that in each step we expand only the most promising hypotheses, and discard (i.e. prune) the

rest. We perform pruning on several levels, for which we distinguish two types of hypotheses:

- **Lexical hypothesis** (C, χ). A lexical hypothesis is identified by its coverage vector C and the model state χ.
- **Reordering hypothesis** C. We will refer to the set of all lexical hypotheses with the same coverage C as a single reordering hypothesis.

We apply histogram pruning [Steinbiss & Tran$^+$ 94] on several levels, i.e. we retain only a fixed number of the most promising partial hypotheses on each pruning level. All other hypotheses are discarded, so that they will not be considered for expansion when generating partial hypotheses for higher cardinalities.

- **Lexical pruning per coverage.** For each reordering hypothesis C we keep a stack of the best scoring N_l lexical hypotheses under the scoring function $\bar{Q}(\cdot, \cdot)$ in memory.

  $$\bar{Q}(C, \chi) := Q(C, \chi) + R(C, \chi) \tag{5.4}$$

  $R(C, \chi)$ is a suitable rest score estimate which will be discussed later in this Section. N_l is called the *lexical histogram size*.

- **Reordering pruning per cardinality.** Per cardinality, the maximum number of reordering hypotheses is N_r. We discard the entire stack of lexical hypotheses for each coverage C that is pruned. Pruning is based on the scoring function

  $$\bar{Q}(C) := \max_{\chi} \bar{Q}(C, \chi). \tag{5.5}$$

  We will refer to N_r as the *reordering histogram size*.

An illustration of these pruning types is given in Figure 5.2. The rest score estimation function $R(C, \chi)$ is necessary in order to avoid the decoder translating the higher scoring "easy" parts of the source sentence first. It is based on heuristics that take an educated guess at the best possible language model, translation model and distortion model scores. In this work, $R(C, \chi) = R(C, j)$ only depends on the coverage C and the last translated source position j. For a description of these rest score heuristics, please refer to [Zens 08], pp. 57-59.

In addition to the above mentioned histogram pruning techniques, we apply constraints on the amount of reordering that is permitted.

- **Phrase-based IBM reordering constraints.** Extending the word-based IBM reordering constraints [Berger & Brown$^+$ 96], we directly discard all coverages C that leave more than $N_{maxGaps}$ contiguous sequences of words uncovered between the first and the last covered positions. The following condition needs to be verified during search:

  $$\left|\{j > 1 | j \in C \wedge j - 1 \notin C\}\right| \leq N_{maxGaps} \tag{5.6}$$

  $N_{maxGaps}$ is called the *maximum number of gaps* and is typically set to 1.

In addition to the source sentence f_1^J, the beam search algorithm takes a matrix $E(\cdot, \cdot)$ as input, which contains the phrase translation options. For each contiguous phrase $\tilde{f} = f_b \ldots f_j$ within the source sentence, $E(b, j)$ is a list of all candidate translations for $\tilde{f}$. We call the procedure of creating this matrix *phrase matching*. For each sentence, the phrase matching is performed directly before the search. As a final pruning technique, we limit the number of target candidates for each source phrase.

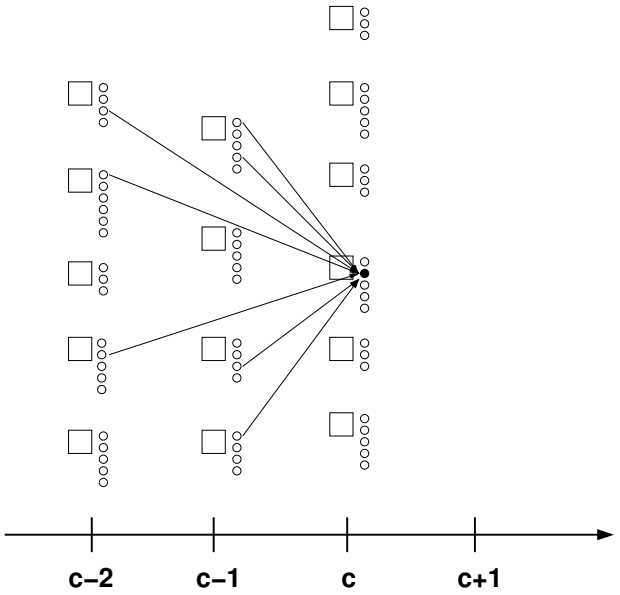


Figure 5.2: Illustration of the lexical and reordering pruning strategies. For each cardinality, we have a list of reordering hypotheses (boxes), identified by a coverage C. For each reordering hypothesis, there is a list of lexical hypotheses (circles). A hypothesis with cardinality c can be generated by expanding a hypothesis of cardinality $c - k$ with a k-word phrase. Taken from [Zens 08].

- **Observation histogram pruning.** For each source phrase $\tilde{f} = f_b \dots f_j$, the best N_o translation candidates according to the scoring function $\bar{Q}(\tilde{f}, \tilde{e})$ are selected in $E(b, j)$ to be considered during search. The scoring is defined by

$$\bar{Q}(\tilde{f}, \tilde{e}) := q_{\mathrm{TM}}(\tilde{e}, b, j) + q_{\mathrm{LME}}(\tilde{e}). \tag{5.7}$$

  Here, $q_{\mathrm{LME}}(\tilde{e})$ is the stand-alone weighted language model score of $\tilde{e}$, without using sentence start and end markers.

During phrase matching, observation histogram pruning is performed by first sorting the phrase pairs by $\bar{Q}(\tilde{f}, \tilde{e})$ and then selecting the top N_o candidates. This has the additional advantage, that the resulting matrix $E(b, j)$ is passed to the decoder with entries sorted according to their model score, which was observed to speed up translation by [Delaney & Shen$^+$ 06]. The pre-sorting defines the order in which the hypothesis expansions take place. As higher scoring phrases are considered first, it is less likely that already created partial hypotheses will have to be replaced, thus effectively reducing the expected number of hypothesis expansions.

5.3 Language Model Look-Ahead

Language model score computations tend to be among the most expensive in decoding. The authors of [Delaney & Shen$^+$ 06] report significant improvements in translation runtime by removing unnecessary language model lookups with an early pruning technique they call *on-the-fly beam pruning*. It is equivalent to lines 10 and 13 in Figure 5.3, which will be described in more detail in Section 5.4. In this Section, we describe a language model look-ahead technique, which is aimed at further reducing the number of language model computations. It is applied in line 14

of the algorithm in Figure 5.3, where the language model look-ahead score $q_{\text{LMLA}}(\tilde{e}|\chi)$ is used to perform early pruning. We propose three different ways for its computation.

Note: For consistency we will continue using the term score rather than cost, which is used interchangeably in the literature. The difference is that costs are generally assumed to be positive for all decisions $(\tilde{e}, b, j)$ under all models, as they were originally defined as the negative logarithm of a probability. This assumption does not always hold in current systems due to non-probabilistic models and possibly negative model weights λ_m. Nevertheless, we will conversely assume scores, i.e. negative costs, to always be negative. As a result we assume that the total score of a translation is monotonically decreasing with respect to the contribution of every model at every elementary decision $(\tilde{e}, b, j)$. This allows us to perform several early pruning techniques that are guaranteed, given the above assumption, not to result in search errors.

- **First-word language model look-ahead pruning** defines the look-ahead score as

$$q_{\text{LMLA}}(\tilde{e}|\chi) = q_{\text{LM}}(\tilde{e}_1|\chi). \tag{5.8}$$

  It is the language model score of the first word of target phrase $\tilde{e}$ given history state χ. Within the log-linear framework, the language model score for the full phrase $\tilde{e}$ is the sum of the scores for each of its words. Therefore, $q_{\text{LM}}(\tilde{e}_1|\chi)$ is an upper bound for the full language model score, given $q_{\text{LM}}(e|\chi) \leq 0 \ \forall e$. This condition is met as $q_{\text{LM}}(e|\chi)$ is the logarithm of a probability. As a result, the technique does not introduce additional search errors, as hypotheses pruned here would otherwise be pruned after the full language model computation in line 16 of the algorithm. $q_{\text{LM}}(\tilde{e}_1|\chi)$ can be reused if the language model score of the full phrase $\tilde{e}$ needs to be computed afterwards. We can exploit the structure of the search to speed up the language model lookups for the first word. The language model probabilities are stored in a *trie*, where each node corresponds to a specific language model history. Usually, each language model lookup consists of first traversing the trie to find the node corresponding to the current language model history and then retrieving the probability for the next word. If the n-gram is not present, we have to repeat this procedure with the next lower-order history, until a probability is found. However, the language model history for the first words of all phrases within the innermost loop of the search algorithm is identical. Just before the loop we can therefore traverse the trie once for the current history and each of its lower order n-grams and store the pointers to the resulting nodes. To retrieve the language model look-ahead scores, we can then directly access the nodes without the need to traverse the trie again. This implementational detail was confirmed to increase translation speed by roughly 20% in a short experiment.

- **Phrase-only language model look-ahead pruning** defines the look-ahead score as

$$q_{\text{LMLA}}(\tilde{e}|\chi) = q_{\text{LME}}(\tilde{e}), \tag{5.9}$$

  which is the stand-alone language model score of the phrase $\tilde{e}$. It was already used for sorting the phrases (cf. Section 5.2). It is therefore pre-computed and does not require additional language model lookups. As it is not a lower bound for the real language model score, this pruning technique can introduce additional search errors. However, our results show that it radically reduces the number of language model lookups and can therefore considerably speed up translation.

- **Hybrid language model look-ahead pruning** combines the two described techniques:

$$q_{\text{LMLA}}(\tilde{e}|\chi) = q_{\text{LM}}(\tilde{e}_1|\chi) + (q_{\text{LME}}(\tilde{e}) - q_{\text{LME}}(\tilde{e}_1)) \tag{5.10}$$

The term $(q_{\mathrm{LME}}(\tilde{e}) - q_{\mathrm{LME}}(\tilde{e}_1))$ is the stand-alone language model score of the phrase $\tilde{e}$ without the estimate for the first word. This is appealing, as the score for the first word is estimated without history, i.e. as a unigram, and is thus the least reliable part of $q_{\mathrm{LME}}(\tilde{e})$. Additionally, as described above, $q_{\mathrm{LM}}(\tilde{e}_1|\chi)$ can be computed efficiently by avoiding multiple trie traversals. Hybrid language model look-ahead can also introduce additional search errors.

5.4 The Full Algorithm

The entire search algorithm is shown in Figure 5.3. FOR, IF, CONTINUE and BREAK have the usual C/C++ semantics. Backpointers are omitted. The search is carried out in a source cardinality synchronous way, meaning that the hypothesis expansions are ordered by the cardinality of the coverage set of the resulting partial hypotheses. For each cardinality c (line 1) we loop over all source phrase lengths l (line 2). The next loop is over all predecessor coverages C' with cardinality $c-l$ in order of their score $\bar{Q}(C') = max_{\chi}\bar{Q}(C', \chi)$ (line 3). Now we loop over all possible start positions j producing no overlap, so effectively the source phrase $\tilde{f} = f_b \ldots f_j, j := b+l-1$ is chosen (line 4), enforce the reordering constraints (line 6) and apply early pruning by checking whether the best scoring predecessor state with the best scoring translation option yield a score within the lexical hypothesis stack of coverage C (line 7). After this we loop over all existing predecessor states (χ'), in order of their score $\bar{Q}(C', \chi')$ (line 8), and again check if we can perform early pruning (line 10). Finally we loop through all phrase translation options $\tilde{e}$ (line 11), for which another early pruning criterion (line 13) and language model look-ahead (line 14, cf. Section 5.3) are evaluated. Note that the BREAK statement in line 13 in combination with pre-sorting of the phrase candidates by a language model score estimate can lead to search errors, as it assumes that the phrases are sorted according to their translation model score only. However, we found this to be of negligible consequence for the translation results. In line 15, the final score including the full language model score is computed. If this score is sufficiently high, the hypothesis is generated, possibly recombined with nodes that are indistinguishable by their label and inserted into the lexical stack (line 18). Finally, if the lexical stack of reordering hypothesis C has too many elements, it is pruned back to size N_l (line 19). After the loop over all previous cardinalities C' is finished, we perform reordering pruning by keeping only the best N_r lexical stacks, where the score of a stack is defined as the score of its best element (line 20).

5.5 Contributions

The entire search algorithm described in this chapter was implemented into RWTH Aachen University's open source toolkit *Jane* [Vilar & Stein$^+$ 10, Wuebker & Huck$^+$ 12] by the author of this dissertation. The language model look-ahead techniques (cf. Section 5.3) are a novel scientific contribution by the author of this work. With the exception of hybrid look-ahead, they were published in [Wuebker & Ney$^+$ 12b]. Here, the original idea of first-word language model look-ahead is due to Richard Zens. The author of this work developed the remaining look-ahead techniques and performed the entire implementation and experimentation.

5.6 Related Work

The search algorithm described here was proposed by [Zens & Ney 08] and is based on the algorithm for single-word models proposed in [Tillmann & Ney 03]. Moses [Koehn & Hoang$^+$ 07b] is the de facto standard open source toolkit for state-of-the-art phrase-based machine translation. Different from the approach described here, it performs dynamic programming beam search

$E(b,j)$	sorted list of all candidate translations for $f_b \dots f_j$
$Q(C,\chi)$	score of hypothesis (C,χ) with coverage C and model state χ
$Q(C)$	$:= max_\chi Q(C,\chi)$
$R(C,j)$	rest score estimate for coverage C and last translated position j
$\bar{Q}(C,\chi)$	$:= Q(C,\chi) + R(C,\chi)$
$q_{\mathrm{TM}}(\tilde{e},b,j)$	weighted sum of local model scores for phrase pair $(f_b^j,\tilde{e})$
$q_{\mathrm{TM}}(b,j)$	$:= max_{\tilde{e}} q_{\mathrm{TM}}(\tilde{e},b,j)$
$q_{\mathrm{LMLA}}(\tilde{e}\|\chi)$	weighted language model look-ahead score of phrase $\tilde{e}$ given state χ
$q_{\mathrm{LM}}(\tilde{e}\|\chi)$	weighted language model score of phrase $\tilde{e}$ given state χ
$q_{\mathrm{Dist}}(b\|\chi)$	weighted distortion score for a jump to source position b given state χ
$q_{\mathrm{HRM}}(\tilde{e},b,j\|\chi)$	weighted HRM score for phrase pair $(f_b^j,\tilde{e})$ under state χ
C `violatesIBM`	true, iff coverage C violates extended IBM reordering constraints
$S(C)$	set of all χ in the current stack of lexical hypotheses for coverage C
`lexPruning` for C	limit the number of elements in stack for coverage set C to N_l
$\bar{Q}_{min}(C)$	lexical pruning threshold $:= \begin{cases} -\infty & \lvert S(C)\rvert < N_l \\ min_{\chi\in S(C)} \bar{Q}(C,\chi) & \text{otherwise} \end{cases}$
`ROPruning` for c	limit the number of stacks for cardinality c to N_r

INPUT: source sentence f_1^J, sorted translation options $E(b,j)$ for $1 \le b \le j \le J$,
models q_{TM}, q_{LM}, q_{Dist}, q_{HRM}

0 $Q(\emptyset,\chi_0) = 0$; all other $Q(\cdot,\cdot)$ entries are initialized to $-\infty$
1 FOR cardinality $c = 1$ TO J DO
2 FOR source phrase length $l = 1$ TO c DO
3 FOR ALL coverages $C' \subset \{1,...,J\} : |C'| = c - l$ DO
4 FOR ALL positions $b \in \{1,...,J\}, j := b + l - 1 : C' \cap \{b,...,j\} = \emptyset$ DO
5 coverage $C = C' \cup \{b,...,j\}$
6 IF C `violatesIBM` THEN CONTINUE
7 IF $Q(C') + R(C,j) + q_{\mathrm{TM}}(b,j) \le \bar{Q}_{min}(C)$ THEN CONTINUE
8 FOR ALL states $\chi' \in S(C')$ DO
9 partial score $q = Q(C',\chi') + q_{\mathrm{Dist}}(b|\chi')$
10 IF $q + R(C,j) + q_{\mathrm{TM}}(b,j) \le \bar{Q}_{min}(C)$ THEN CONTINUE
11 FOR ALL phrase translations $\tilde{e} \in E(b,j)$ DO
12 partial score $q' = q + q_{\mathrm{TM}}(\tilde{e},b,j)$
13 IF $q' + R(C,j) \le \bar{Q}_{min}(C)$ THEN BREAK
14 IF $q' + q_{\mathrm{LMLA}}(\tilde{e}|\chi') + R(C,j) \le \bar{Q}_{min}(C)$ THEN CONTINUE
15 final score $q'' = q' + q_{\mathrm{LM}}(\tilde{e}|\chi') + q_{\mathrm{HRM}}(\tilde{e},b,j|\chi')$
16 IF $q'' + R(C,j) \le \bar{Q}_{min}(C)$ THEN CONTINUE
17 state $\chi = \chi' \oplus (\tilde{e},b,j)$
18 IF $q'' > Q(C,\chi)$ THEN $Q(C,\chi) = q''$, insert χ into S(C)
19 `lexPruning` for C
20 `ROPruning` for c

Figure 5.3: The dynamic programming beam search algorithm.

with a single hypothesis stack for each cardinality, rather than separating lexical from reordering hypotheses. To increase the efficiency of the search for the best translation, research efforts have been made to explore several directions, ranging from generalizing the stack decoding algorithm [Ortiz & Garcia-Varea$^+$ 06] to additional early pruning techniques [Delaney & Shen$^+$ 06], [Moore & Quirk 07b] and more efficient language model querying [Heafield 11]. Language model look-ahead is a standard technique in automatic speech recognition where it has been used for many years [Steinbiss & Tran$^+$ 94, Ortmanns & Ney$^+$ 98, Nolden & Ney$^+$ 11]. At each state

within search, the speech recognizer uses the best language model costs of all words reachable from the current state as a lower bound for pruning. This work presented an adaptation of this approach to machine translation with words as the smallest units.

6. Training

This chapter addresses the *training* problem as defined in Section 3.2. We will mainly focus on the phrasal translation models described in Section 4.2.1 and the discriminative models given in Section 4.2.10. With both generative and discriminative training methods, two fundamentally different approaches are being investigated (cf. Section 1.2). Similar to Chapters 4 and 5, we strive to provide a complete and consistent picture, which is why we start by reviewing the previous state of the art up to Section 6.2.2 and afterwards introduce our novel contributions. Section 6.4 is dedicated to highlighting the novel contributions and their originators. The whole chapter is being put into perspective with respect to related work in Section 6.5.

6.1 Phrase Extraction

Most state-of-the-art translation systems apply a heuristic phrase extraction procedure to obtain the phrase table [Och & Tillmann$^+$ 99]. It takes a word-aligned bitext as input. The word alignment is generally computed by symmetrization of the directional source-to-target and target-to-source Viterbi alignments based on the IBM models (cf. Section 3.4). The output is a set of phrase pairs with accompanying scores for the models described in Sections 4.2.1 through 4.2.4 and 4.2.8. Given a sentence pair (e_1^I, f_1^J) and its alignment matrix $A \subseteq I \times J$ we extract the following set of bilingual phrases.

$$\begin{aligned}\mathcal{BP}(e_1^I, f_1^J, A) = \Big\{ (e_{i_1}^{i_2}, f_{j_1}^{j_2}) : \forall (i,j) \in A : i_1 \leq i \leq i_2 \leftrightarrow j_1 \leq j \leq j_2 \\ \wedge \exists (i,j) \in A : i_1 \leq i \leq i_2 \wedge j_1 \leq j \leq j_2 \Big\}\end{aligned} \tag{6.1}$$

Informally, a phrase pair is extracted if its source and target side are both contiguous, and if none of the words inside the phrase are aligned to a word outside of the phrase. This is done for every sentence pair in the bitext and the occurrences of each phrase pair are counted in order to estimate relative frequency models. To constrain the phrase table size, we restrict the maximum length of the phrases. We define the length $|\tilde{f}|$ of a phrase $\tilde{f}$ as the number of its words and only extract phrase pairs $(\tilde{e}, \tilde{f})$ with $|\tilde{e}| \leq e_{max}$ and $|\tilde{f}| \leq f_{max}$. In this work, we typically set $e_{max} = f_{max} = 6$.

The phrase translation models are computed from the joint counts $N(\tilde{f}, \tilde{e})$ and the marginal counts $N(\tilde{f})$ and $N(\tilde{e})$ as described in Equation 4.11. These counts can be computed in different ways. In the popular Moses toolkit [Koehn & Hoang$^+$ 07b], the joint count $N(\tilde{f}, \tilde{e})$ is increased by 1 for every time a phrase pair $(\tilde{f}, \tilde{e})$ is extracted. This means that the joint counts are symmetric, i.e. $N_{Moses}(\tilde{f}, \tilde{e}) = N_{Moses}(\tilde{e}, \tilde{f})$. The marginals are computed as $N_{Moses}(\tilde{e}) = \sum_{\tilde{f}} N_{Moses}(\tilde{f}, \tilde{e})$.

In our work we follow a different approach. If a single occurrence of target phrase $\tilde{e}$ is extracted with $N > 1$ source phrases $\tilde{f}$, each of them contributes to the joint count $N(\tilde{f}, \tilde{e})$ with $\frac{1}{N}$ in order to penalize ambiguous phrase pairs. This happens if the source phrase has unaligned boundary words, which can be added or removed from the phrase without violating the consistency condition

in Equation 6.1. As marginal counts $N(\tilde{e})$ we simply use the number of occurrences of $\tilde{e}$ in the target side of the bitext, independent from the word alignment. Note that $N(\tilde{e}) \geq \sum_{\tilde{f}} N(\tilde{f}, \tilde{e})$, which means that the resulting probability estimate $p(\tilde{f}|\tilde{e})$ does not necessarily satisfy the sum-to-one constraint, i.e. $\sum_{\tilde{f}} p(\tilde{f}|\tilde{e}) \leq 1$. Effectively we thus assume that $p(\tilde{f}|\tilde{e})$ includes an implicit segmentation prior $\frac{\sum_{\tilde{f}} N(\tilde{f},\tilde{e})}{N(\tilde{e})}$, which can be interpreted as the likelihood that the target phrase $\tilde{e}$ is consistent with the word alignment, given that it appears in the target sentence.

The word lexicon model described in Section 4.2.2 is also typically estimated from heuristic counts extracted from the word alignment as $p(f|e) = \frac{N(f,e)}{N(e)}$. The corresponding class-based models h_{wc_Phr} and h_{wc_Lex} (cf. Section 4.2.9) can easily be estimated in the same manner as their word-based counterparts by replacing all words by their class in the bitext while keeping the alignment matrix unchanged.

The hierarchical reordering model introduced in Section 4.2.8 is also computed from the word-aligned bitext. The orientation counts are obtained by extracting *corners* along with the phrase pairs with the *corner propagation algorithm* presented by [Cherry & Moore$^+$ 12]. The orientation model is then estimated by relative frequencies with a smoothing factor $\sigma > 0$ as follows.

$$p(O|\tilde{f}, \tilde{e}) = \frac{\sigma \cdot p(O) + N(O, \tilde{f}, \tilde{e})}{\sigma + \sum_{O' \in \{M,S,D\}} N(O', \tilde{f}, \tilde{e})} \tag{6.2}$$

$$p(O) = \frac{N(O)}{\sum_{O' \in \{M,S,D\}} N(O')} \tag{6.3}$$

$$N(O) = \sum_{(\tilde{f},\tilde{e})} N(O, \tilde{f}, \tilde{e}) \tag{6.4}$$

We usually set $\sigma = 1$. For more details on hierarchical reordering model training please refer to [Rietig 13].

6.2 Generative Training

6.2.1 Introduction

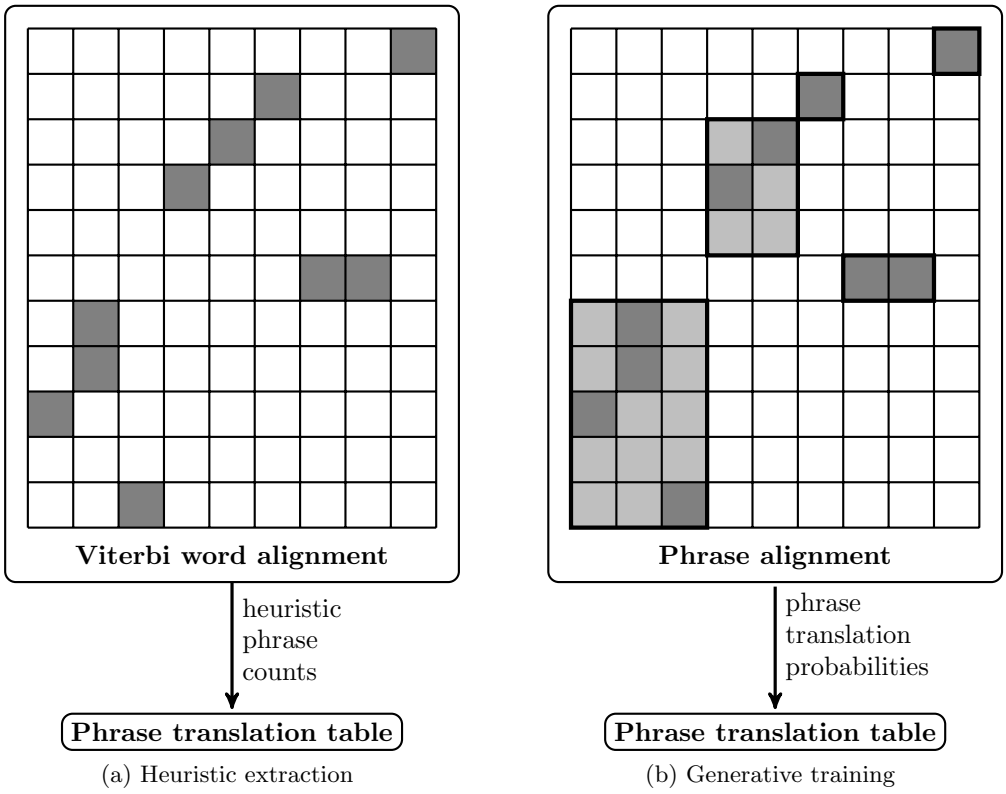
The training procedure presented in the previous section has several drawbacks.

(i) The phrase extraction is based on heuristics that have no foundation in statistics.

(ii) The models used to generate word alignments are inconsistent with the models used in decoding.

(iii) All models are trained independently, without taking their interaction and interdependencies into account.

In this work we aim at going beyond simple counting of heuristically extracted phrases. The generative training procedure we apply is inspired by the expectation-maximization (EM) algorithm [Dempster & Laird$^+$ 77] and makes use of a modified version of the translation decoder which is constrained to produce the reference translation to obtain a phrase alignment. We call this process *forced alignment* or *force-aligning* the training data. As a result, the models in training and decoding are identical. The **E-step** of the algorithm corresponds to force-aligning the training data with a constrained translation decoder, which yields a distribution over possible phrasal segmentations and their alignment. Different from original EM, we make use of not only the two translation channel models that are being learned, but the full log-linear combination



(a) Heuristic extraction (b) Generative training

Figure 6.1: Illustration of the difference between the state-of-the-art heuristic phrase extraction and generative phrase training.

of models as in translation decoding. In the **M-step**, we re-estimate the phrase table from the phrase alignments.

An illustration of the basic idea can be seen in Figure 6.1. In the literature this method by itself has been shown to be problematic because it suffers from over-fitting [DeNero & Gillick$^+$ 06], [Liang & Buchard-Côté$^+$ 06]. Since the initial phrases were extracted from the same training data that we are force-decoding, very long phrases can be found for segmentation. As these long phrases tend to occur in few training sentences, the EM algorithm generally overestimates their probability. Shorter phrases, which generalize better to unseen data and thus are more useful for translation, are neglected. In order to counteract these effects, our training procedure applies leave-one-out and cross-validation heuristics. Our results show that this improves translation quality.

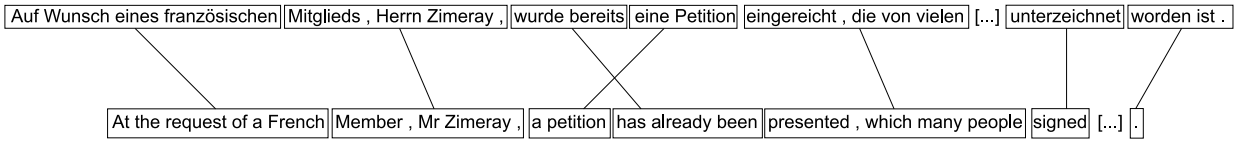
Ideally, we would produce all possible segmentations and alignments during training. However, this has been shown to be infeasible for real-world data [DeNero & Klein 08]. As training uses a modified version of the translation decoder, it is straightforward to apply pruning in a manner identical to regular decoding. We consider three ways of approximating the full search space:

(i) the single-best Viterbi alignment,

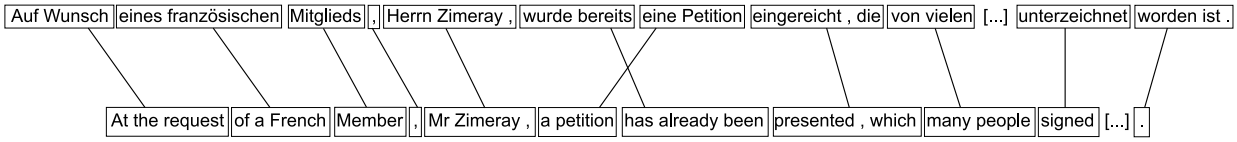
(ii) the n-best alignments,

(iii) all alignments remaining in the search space after pruning.

In order to be able to force-align the training data, we need an initial phrase table. We



(a) Without leave-one-out



(b) With leave-one-out

Figure 6.2: Effect of leave-one-out on a phrase segmentation in forced alignment.

experiment with two types of initialization. As a first step, we can initialize training with the table extracted from word alignments as described in Section 6.1. Our experiments show that this way we are capable of improving translation quality while at the same time substantially reducing the number of phrase pairs in the table, resulting in more efficient translation systems. However, the ultimate goal is to do away with the dependence on inconsistent word alignment procedures and extraction heuristics. We therefore introduce extensions that allow us to initialize with an empty phrase table. New phrase pairs are generated on the fly during training. This is done in a length-incremental fashion, starting with short phrases and increasing the maximum length in each iteration.

6.2.2 Forced Alignment

The main idea of forced alignment is to produce a phrase segmentation and alignment for each sentence pair of the training data using the same models that are used in unconstrained search. The search process described in Section 5 is extended by constraining partial translations to the corresponding reference target language sentences. Given a source sentence f_1^J and target sentence e_1^I, we modify Equation 5.1 to define the best phrase segmentation $\hat{s}_1^{\hat{K}}(e_1^I, f_1^J)$ that covers both sentences under our set of models.

$$(e_1^I, f_1^J) \rightarrow \hat{s}_1^{\hat{K}}(e_1^I, f_1^J) = \underset{K, s_1^K}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \tag{6.5}$$

The models are identical to the ones used in unconstrained translation. The language model h_{LM}, however, can in practice be ignored, as its score is constant in constrained search. In addition to the target constraint, we extend the phrase matching (cf. Section 5.2) to the bilingual case, discarding all candidates that are no sub-sequences of the target language sentence. Sentences for which the decoder can not find an alignment are discarded for phrase model training. As *coverage* we define the percentage of sentence pairs that the decoder can successfully align. In order to obtain a good coverage of the training data, in practice the constraint is loosened to allow alignment of the source sentence to the incomplete prefixes of the target sentence. A fixed penalty is added for each uncovered target word.

Leave-One-Out

As was mentioned in Section 6.2.1, previous approaches found over-fitting to be a problem in phrase model training. The leave-one-out method we will describe in this section tackles this

by penalizing rare phrase pairs and thus generating phrase alignments that are more likely to generalize to unseen data.

The same training data consisting of N parallel sentence pairs (F_n, E_n) is used both for initialization of the translation model $p(\tilde{f}|\tilde{e})$ and phrase model training. While in this way we can make full use of the available data and avoid out-of-vocabulary words during training, it has the disadvantage that it can lead to over-fitting. All phrases that were extracted from a specific sentence pair (F_n, E_n) can be used in constrained search for the alignment of this sentence pair. This especially includes long phrases that only appear in few training sentences. As a result, the training procedure over-fits those long phrases to a single or few sentence pairs and heavily overestimates their translation probabilities. In the extreme case, whole sentences will be learned as phrasal translations. The average length of the phrase pairs used for alignment is an indicator of this kind of over-fitting, as the number of matching training sentences decreases with increasing phrase length. We show an example in Figure 6.2. Without application of the leave-one-out heuristic the sentence is segmented into a few long phrases, which are unlikely to generalize to real-world data. Phrase boundaries are unintuitive. With the leave-one-out method the phrase pairs are shorter and therefore better suited for generalization to unseen data.

Previous attempts have tried to deal with the over-fitting problem by limiting the maximum phrase length [DeNero & Gillick$^+$ 06, Marcu & Wong 02] and by smoothing phrasal probabilities with lexical models [Ferrer & Juan 09]. However, [DeNero & Gillick$^+$ 06] experienced similar over-fitting effects with short phrases due to the fact that the same word sequence can be segmented in different ways, leading to specific segmentations being learned for specific training sentence pairs. Our results confirm these findings. In this work we therefore apply a leave-one-out strategy, which has proven successful in language modeling [Kneser & Ney 95, Chen & Goodman 98].

The basic idea of the leave-one-out (more precisely: leave-one-sentence-out) method is to remove the sentence that is currently being aligned by the decoder from the statistics of the models that are being used. For each sentence pair, we thus obtain a model that was trained on all data except for this one sentence. We modify the phrase translation probabilities for each sentence pair individually as follows. Given a training example $(F_n, E_n) = (f_1^J, e_1^I)$ and the corresponding alignment matrix A, we re-extract all phrase pairs from this sentence following Equation 6.1 and obtain *local counts*. As local counts $N_n(\tilde{f}, \tilde{e})$, $N_n(\tilde{e})$ and $N_n(\tilde{f})$ we define the joint and marginal counts that can be extracted from the n-th sentence pair in the training data. These are subtracted from the global counts $N(\tilde{f}, \tilde{e})$, $N(\tilde{e})$ and $N(\tilde{f})$ in order to re-compute the translation probabilities:

$$p_{l1o,n}(\tilde{f}|\tilde{e}) = \begin{cases} \frac{N(\tilde{f},\tilde{e}) - N_n(\tilde{f},\tilde{e})}{N(\tilde{e}) - N_n(\tilde{e})} & N(\tilde{f},\tilde{e}) - N_n(\tilde{f},\tilde{e}) > 0 \\ \gamma(\tilde{f},\tilde{e}) & \text{otherwise} \end{cases} \tag{6.6}$$

$$\gamma(\tilde{f},\tilde{e}) = \begin{cases} \alpha & \text{standard leave-one-out} \\ \beta^{(|\tilde{f}|+|\tilde{e}|)} & \text{length-based leave-one-out} \end{cases} \tag{6.7}$$

The inverse direction model $p_{l1o,n}(\tilde{e}|\tilde{f})$ is computed analogously. The resulting model is identical to the one we were to obtain by removing the current sentence pair from the training data and applying Equation 4.11. For efficient computation, the source and target phrase marginal counts are stored in the phrase table along with the translation probabilities. On-the-fly computation of the global joint counts is straightforward. Singleton phrases, that would be assigned a probability of zero according to the above definition, require special treatment. We refer to singleton phrases as phrase pairs that can be only extracted from a single sentence of the training data. In many cases, the decoder requires these singleton phrases to produce an alignment on the sentence pair they were extracted from. Therefore, we keep these phrases available for the decoder by assigning

Table 6.1: Average source phrase lengths in forced alignment without the leave-one-out method and with the standard and length-based leave-one-out strategies on the WMT 2008 German→English Europarl task.

leave-one-out	type	average phrase length
no	-	2.5
yes	standard	1.9
	length-based	1.6

them a penalty, corresponding to a positive probability $\gamma(\tilde{f}, \tilde{e})$ close to zero. We evaluated two different strategies for this, which we call *standard* and *length-based* leave-one-out. The standard leave-one-out method assigns the same fixed probability $\gamma(\tilde{f}, \tilde{e}) = \alpha$ to singleton phrase pairs. This way the decoder will prefer using more frequent phrases for the alignment, but is able to resort to singletons if necessary. However, we found that this results in longer singleton phrases being preferred over shorter ones, because fewer of them are needed to produce the target sentence. In order to better generalize to unseen data, we would like to give the preference to shorter phrases. This is achieved by the length-based leave-one-out method, where singleton phrases are assigned the probability $p_{l1o,n}(\tilde{f}, \tilde{e}) = \gamma(\tilde{f}, \tilde{e}) = \beta^{(|\tilde{f}|+|\tilde{e}|)}$, where $|\tilde{f}|$ and $|\tilde{e}|$ are the source and target phrase lengths and $\beta < 1$ is fixed. In our experiments we set $\alpha = e^{-20}$ and $\beta = e^{-5}$. Table 6.1 shows the decrease in average source phrase length by application of the leave-one-out strategy on the German→English Europarl data set.

Cross-Validation

The leave-one-out method described in the previous section can only be implemented efficiently for the first training iteration and if we initialize with the phrase table extracted from word alignment. In all other cases, we would have to store local phrase counts for all training sentence pairs on disk and access them at training time. To avoid the infeasible overhead in terms of disk storage and i/o, we propose a generalization to which we refer as cross-validation. The idea is identical to leave-one-out, but instead of re-computing the translation model for every single sentence, we work with batches of training examples. In our experiments, the size of these batches is set to 10k sentence pairs.

Parallelization

To cope with the runtime and memory requirements of phrase model training that was pointed out by previous work [Marcu & Wong 02, Birch & Callison-Burch$^+$ 06], we parallelize the forced alignment procedure by splitting the training corpus into blocks of 10k sentence pairs. From the initial phrase table, each of these blocks only loads the phrases that are required for alignment. Alignment and phrase counting are performed separately for each block and then accumulated to build the entire trained phrase table.

Local Language Models

On the target side of each parallelization block we train a unigram language model, which we call the *local language model*. It is applied to compute the phrase-internal language model estimate $q_{\mathrm{LME}}(\tilde{e})$ for phrase candidate pre-sorting and observation histogram pruning (cf. Equation 5.7). One effect of this is that the order in which phrase candidates are considered is adjusted to the local part of the data, which has a positive effect on decoding speed. Secondly, the phrase

candidates selected by observation histogram pruning better suit the current data block. As a result, the number of phrases remaining after bilingual phrase matching is increased compared to the same setup without a local language model.

Backoff Phrases

Backoff phrases are phrase pairs that are generated on-the-fly by the decoder at training time. When aligning a sentence pair, the decoder inserts the cross product of all source m_s-grams and target m_t-grams into the translation options for given maximum lengths m_s and m_t. Formally, for the sentence pair (f_1^J, e_1^I) and maximum lengths m_s and m_t, we generate all phrase pairs $(\tilde{f}, \tilde{e})$ where the following condition holds.

$$\begin{aligned} \exists j, i: \quad & 1 \leq j \leq J - m_s + 1 \\ \wedge \; & 1 \leq i \leq I - m_t + 1 \\ \wedge \; & \tilde{f} = f_j^{(j+m_s-1)} \\ \wedge \; & \tilde{e} = e_i^{(i+m_t-1)} \end{aligned} \tag{6.8}$$

These generated phrase pairs are given a fixed penalty π_p per phrase, π_s per source word and π_t per target word, which are summed up and substituted for the two channel models. The lexical smoothing scores are computed in the usual way based on an IBM-1 table. Note that this table is not heuristically extracted from a word alignment, but contains the real translation probabilities estimated with the IBM-1 model by GIZA++.

We use backoff phrases in different contexts. Firstly, they can be applied as a means to bootstrap a new phrase table without initialization, i.e. it is possible to entirely drop the dependency on the heuristically extracted set of phrases. Secondly, they are helpful in order to avoid alignment failures and thus can increase the coverage of the training data. In this case, we also make use of a modified version, where new phrase pairs $(\tilde{f}, \tilde{e})$ are only generated if there are no translation candidates available for $\tilde{f}$ after the bilingual phrase matching. If we initialize with the baseline phrase table or already have generated a sufficient amount of phrase pairs, backoff phrases are usually only used if a first decoding run fails and we have to resort to *fallback runs*, which are described in the next paragraph.

Fallback Decoding Runs

Whenever constrained decoding fails, we allow for *fallback decoding runs* with slightly altered parameterization, in order to maximize the number of successfully aligned sentences. In this work, we only change the parameterization of the backoff phrases. If a first decoding run fails, we perform a second run where we allow to generate backoff phrases for all source phrases that have no target candidates after the bilingual phrase matching. Finally, if the second run also fails, the cross-product of source and target n-grams is generated in the third run. In these cases, the maximum backoff phrase length is typically fixed to $m_s = m_t = 1$. We denote the number of fallback runs with $n_{fb} = 2$.

6.2.3 Phrase Model Training

We can draw the statistics used to train our generative models either from a single best alignment, from the n-best alignments or from an alignment graph representing all alignments explored by the decoder. We have developed two different models for phrase translation probabilities that make use of the force-aligned training data. Additionally we consider smoothing by phrase table interpolation of the generative model with the state-of-the-art heuristics.

Viterbi

The simpler one of our generative phrase models estimates phrase translation probabilities by their relative frequencies in the Viterbi (n-best) alignments of the data. This is similar to the heuristic model with counts drawn from the phrase-aligned data produced in training rather than computed on the basis of a word alignment. The translation probability of a phrase pair $(\tilde{f}, \tilde{e})$ is estimated as

$$p_{FA}(\tilde{f}|\tilde{e}) = \frac{N_{FA}(\tilde{f}, \tilde{e})}{\sum_{\tilde{f}'} N_{FA}(\tilde{f}', \tilde{e})} \tag{6.9}$$

where $N_{FA}(\tilde{f}, \tilde{e})$ is the count of the phrase pair $(\tilde{f}, \tilde{e})$ in the phrase-aligned training data. This can be applied to either the single Viterbi phrase alignment or an n-best list. In the simplest model, each hypothesis in the n-best list is weighted equally. We will refer to this model as the *count* model. We also experimented with weighting the counts by the likelihood estimate of the corresponding entry in the the n-best list. The likelihoods of all n-best list entries are normalized to 1. We will refer to this model as the *weighted count* model.

Forward-Backward

Ideally, the training procedure would consider all possible alignment and segmentation hypotheses with alternatives being weighted by their posterior probability. As discussed earlier in Chapter 5, exhaustive search over all possible alignments is infeasible for typical tasks. However, we can approximate the full hypothesis space with the decoder's search space. It can be represented as a graph of partial hypotheses [Ueffing & Och$^+$ 02] on which we can compute expectations using the forward-backward algorithm. We will refer to the model trained in this manner as the *full* model. In contrast to the Viterbi method described above, phrases are weighted by their posterior probability in the word graph. As suggested in work on minimum Bayes-risk decoding for statistical machine translation [Tromble & Kumar$^+$ 08, Ehling & Zens$^+$ 07], we use a global factor to scale the posterior probabilities.

Phrase Table Interpolation

[DeNero & Gillick$^+$ 06] have reported improvements in translation quality by interpolation of phrase tables produced by the generative and the heuristic model. We adopt this method and report results using linear and log-linear interpolation of our generative model with the baseline phrase table.

The interpolated probabilities $p_{linInt}(\tilde{f}|\tilde{e})$ and $p_{logInt}(\tilde{f}|\tilde{e})$ of the phrase translation probabilities are computed as

$$p_{linInt}(\tilde{f}|\tilde{e}) = (1-\omega) \cdot p_H(\tilde{f}|\tilde{e}) + \omega \cdot p_{gen}(\tilde{f}|\tilde{e}) \tag{6.10}$$

$$p_{logInt}(\tilde{f}|\tilde{e}) = \Big(p_H(\tilde{f}|\tilde{e})\Big)^{1-\omega} \cdot \Big(p_{gen}(\tilde{f}|\tilde{e})\Big)^{\omega} \tag{6.11}$$

where ω is the interpolation weight, p_H the heuristically estimated baseline phrase model and p_{gen} the generative model. The interpolation weight ω is adjusted on the development corpus. Log-linear interpolation generally retains the intersection and linear interpolation the union of the phrase pairs present in the two tables. Note that log-linear interpolation results in an unnormalized probability distribution.

In addition to a hand-adjusted fixed interpolation weight, we also experimented with adding both the heuristic and the generative phrase models as separate feature functions $h_m(\cdot)$ in the

```
1  Initialize with empty phrase table
2  Set backoff phrase penalties π_p = 15, π_s = π_t = 3
3  Set maximum backoff phrase length m_s = m_t = 0
4  Set number of fallback runs n_fb = 0
5  WHILE m_s, m_t ≤ m_max:
6     Set m_s = m_s + 1, m_t = m_t + 1
7     Set λ_Phr = λ_Phr + δ
8     Set λ_iPhr = λ_iPhr + δ
9     Force-align training data and re-estimate phrase table
10 Set m_s = m_t = 1, n_fb = 2
11 WHILE iteration < iteration_max:
12    Force-align training data and re-estimate phrase table
```

Figure 6.3: The complete length-incremental training algorithm.

log-linear framework. This way we allow different interpolation weights for the two translation directions and can optimize them automatically along with the other feature weights. We will refer to this method as using *separate features*. In this case we also retain the intersection of the two phrase tables. Theoretically, separate features have more degrees of freedom and should perform at least as well as fixed interpolation. However, our results show a slightly lower performance. This can be explained by the fact that minimum error rate training of the log-linear parameters is less reliable with a higher number of models.

6.2.4 Length-Incremental Training

As we mentioned in Section 6.2.1, our ultimate goal is to entirely drop the dependence on word alignments and heuristic extraction. In this section we describe the length-incremental training algorithm, which in the first iteration is initialized with an empty phrase table, generating phrases on the fly via backoff phrases. The maximum backoff phrase length is first set to $m_s = m_t = 1$. Afterwards, we iterate the forced alignment training, increasing m_s and m_t by 1 in each iteration up to a maximum of $m_{max} = 6$. To avoid over-fitting, we apply cross-validation. The lexical probabilities for backoff phrases are computed using an IBM-1 table [Brown & Della Pietra$^+$ 93] rather than count-based relative frequencies.

After $m_{max} = 6$ iterations, we have created a sufficient number of phrase pairs and continue iterating the training procedure with new parameters. Now, we introduce two fallback decoding runs. In the first run, no backoff phrases are generated. If the first run fails, we allow backoff phrase generation for source phrases without translation candidates. If this one also fails, all possible backoff phrases are generated in the final decoding run. The maximum backoff phrase length here is $m_s = m_t = 1$.

The log-linear feature weights λ_m used for training are mostly standard values. Only λ_{wp} for the word penalty is hand-adjusted, which will be described in more detail in Section 7.3.2, and λ_{Phr}, λ_{iPhr} for the two phrasal channel models are incremented with each iteration. The idea is that we want to put more trust into the estimated models with each additional training iteration. We start off with $\lambda_{Phr} = \lambda_{iPhr} = 0$ and increment the weights by $\delta = 0.02$ in each iteration, until the standard value $\lambda_{Phr} = \lambda_{iPhr} = 0.1$ is reached in iteration 6. Afterwards the values are kept fixed. Minimum error rate training is not part of the phrase training procedure, but only used later for evaluation. The algorithm is illustrated in Figure 6.3.

We also experiment with two variants of this algorithm, where the length-increment is only applied to either the source or the target side, while the other is fixed to m_{max} from the beginning.

We will refer to these as *source-length-incremental training* and *target-length-incremental training*. In source-length-incremental training, we fix the target backoff phrase length to $m_t = m_{max}$ throughout the entire training procedure. The source backoff phrase length is set to $m_s = 1$ in the first iteration, and incremented by 1 in each of the subsequent iterations. Target-length-incremental training is defined analogously. This has the advantage that the decoder can better handle differences in the source and target sentence lengths already in the first iteration.

6.3 Discriminative Training

6.3.1 Introduction

Discriminative training aims at tackling the same drawbacks of the standard heuristic training pipeline that we outlined in Section 6.2.1. This section is mainly based on [Wuebker & Muehr$^+$ 15] and closely follows its structure.

The main advantage of learning parameters in a discriminative fashion is that it is possible to directly optimize towards the quality or error measure on the task that we want to perform. In statistical machine translation, it has become the state of the art to extend the generative noisy-channel formulation [Brown & Della Pietra$^+$ 93] as a discriminative, log-linear combination of multiple models [Och 03], as was described in Section 3.3. However, these component models are usually still estimated by heuristics, as outlined in Section 6.1.

In this work, we will present a flexible and efficient discriminative training procedure that can be applied to any number and any type of features. The RPROP algorithm is used to optimize a maximum expected Bleu objective. To approximate the infeasibly large space of translation hypotheses we use n-best lists. Leave-one-out can be applied to the n-best list generation to make them more representative with respect to unseen data.

We make the following scientific contributions:

1. We propose to apply the RPROP algorithm for maximum expected Bleu training and perform an experimental comparison with growth transformation (GT) [He & Deng 12, Setiawan & Zhou 13], stochastic gradient descent [Auli & Galley$^+$ 14] and the AdaGrad algorithm [Green & Wang$^+$ 13]. RPROP yields superior performance, reaching a total improvement of 1.2 Bleu points over our IWSLT German→English baseline using 5.22M features.

2. Our experiments show that in terms of time and memory efficiency, RPROP clearly outperforms GT. The latter needs to update a much larger number of features due to its re-normalization component. On the IWSLT data, RPROP is 6.4 times faster than GT and requires a third of the memory.

3. On the WMT German→English task, we perform discriminative training on 4M sentence pairs, which, to the best of our knowledge, is 2.4 times the size of the largest training set reported in previous work (1.66M sentences in [Simianer & Riezler$^+$ 12]). This proves the scalability of our approach.

4. On two large scale tasks our experiments show good improvements over strong baselines which include recurrent language modeling components. On the Chinese→English DARPA BOLT task, we achieve nearly twice the improvement reported in [Setiawan & Zhou 13] on the same test sets which results in a superior final system. Finally, the best single system reported on matrix.statmt.org is outperformed by 0.8 Bleu points on the WMT German→English *newstest2013* set.

5. We show that leave-one-out has a positive impact on translation quality.

6.3.2 Update Strategies

Previously Proposed Algorithms

The **Growth Transformation (GT)** or Extended Baum-Welch Algorithm was proposed by [He & Deng 12] for maximum expected BLEU training of the standard phrase translation and word lexicon models. It is an algorithm to iteratively optimize polynomials of random variables that are subject to sum-to-one constraints. Therefore, it is suitable for training probability distributions. The disadvantage is that each parameter update requires a re-normalization step, which artificially blows up the number of features that are being updated and significantly compromises time and memory efficiency. The update formulas are derived in [He & Deng 12].

Stochastic Gradient Descent (SGD) is a well-known and frequently applied training algorithm, which is used by [Auli & Galley$^+$ 14] for maximum expected BLEU training of reordering models. It performs the following update:

$$\vartheta^{(t+1)} = \vartheta^{(t)} + \eta \cdot \nabla_{\vartheta}^{(t)} \tag{6.12}$$

Here, $\nabla_{\vartheta}^{(t)}$ is the gradient of the objective function with respect to parameter ϑ in iteration t. Its disadvantage is the high sensitivity to the fixed learning rate η. However, as it does not subject the feature values to sum-to-one-constraints, it is considerably more time and memory efficient than growth transformation.

As an improvement to stochastic gradient descent, **AdaGrad** [Duchi & Hazan$^+$ 11] is designed for large sets of sparse features and makes use of an adaptive learning rate. It was proposed for training machine translation models by [Green & Wang$^+$ 13]. Although its main area of application are online algorithms, it can also be applied in our offline setting and is more robust than stochastic gradient descent due to the adaptive learning rate. Following [Green & Wang$^+$ 13], we apply the approximation with a diagonal outer product matrix with non-zero entries G_θ, which is computationally cheap. This results in the update equations

$$\vartheta^{(t+1)} = \vartheta^{(t)} + \eta \cdot (G_\theta^{(t)})^{-\frac{1}{2}} \cdot \nabla_{\vartheta}^{(t)} \tag{6.13}$$

$$G_\theta^{(t)} = G_\theta^{(t-1)} + (\nabla_{\vartheta}^{(t)})^2 \tag{6.14}$$

RPROP

The resilient backpropagation algorithm (RPROP) proposed by [Riedmiller & Braun 93] is a gradient-based optimization algorithm that empirically learns the step size without taking the slope into account, which makes it highly robust and avoids the need for a learning rate. Every time the gradient switches algebraic sign compared to the previous iteration, the last step is reverted and the step size reduced. If the sign remains unchanged compared to the previous iteration, the step size is increased. Formally, given a set of parameters Θ and an objective function $O(\Theta)$, in iteration t each parameter $\vartheta \in \Theta$ is updated according to

$$\vartheta^{(t+1)} = \begin{cases} \vartheta^{(t-1)} & \text{, if} \quad \nabla_{\vartheta}^{(t-1)} \cdot \nabla_{\vartheta}^{(t)} < 0 \\ \vartheta^{(t)} + \Delta\vartheta^{(t)} & \text{, else if} \quad \nabla_{\vartheta}^{(t)} > 0 \\ \vartheta^{(t)} - \Delta\vartheta^{(t)} & \text{, else if} \quad \nabla_{\vartheta}^{(t)} < 0 \\ \vartheta^{(t)} & \text{, otherwise} \end{cases} \tag{6.15}$$

where $\nabla_{\vartheta}^{(t)} := \frac{\partial O(\Theta^{(t)})}{\partial \vartheta}$ denotes the derivative of the objective function. The step size $\Delta\vartheta^{(t)} > 0$ is increased or decreased depending on the sign of the gradient:

$$\Delta\vartheta^{(t)} = \begin{cases} \eta^{+} \cdot \Delta\vartheta^{(t-1)} & \text{, if } \quad \nabla_{\vartheta}^{(t-1)} \cdot \nabla_{\vartheta}^{(t)} > 0 \\ \eta^{-} \cdot \Delta\vartheta^{(t-1)} & \text{, if } \quad \nabla_{\vartheta}^{(t-1)} \cdot \nabla_{\vartheta}^{(t)} < 0 \\ \Delta\vartheta^{(t-1)} & \text{, otherwise} \end{cases} \tag{6.16}$$

The strength parameters $0 < \eta^{-} < 1 \leq \eta^{+}$ usually have little impact and are fixed to $\eta^{-} = 0.5$ and $\eta^{+} = 1.2$ throughout this work. The RPROP algorithm is simple, easy to implement and has proven effective for a number of tasks, e.g. in [Wiesler & Richard^{+} 13, Heigold & Hahn^{+} 11, Lavergne & Allauzen^{+} 11, Hahn & Dinarelli^{+} 11]. Different from growth transformation, it does not assume the parameters to form a probability distribution and performs its updates without a sum-to-one constraint.

Compared to stochastic gradient descent and AdaGrad, RPROP's practical advantage is the absence of a learning rate that needs to be tuned by hand. Further, we see its theoretical advantage in the empirically learned step size. In the first iterations, RPROP's updates are considerably smaller than with the other strategies, resulting in a more careful exploration of the search space. In higher iterations, the update step size keeps growing for good features and we observe an exponential increase of the objective function. In contrast, growth transformation, stochastic gradient descent, and AdaGrad determine the size of their feature updates based on the slope of the gradient, which we believe to be misleading given the complex topology of the feature space in machine translation.

6.3.3 Maximum Expected BLEU

Following [He & Deng 12], we want to optimize a maximum expected Bleu objective. We require a sentence-level metric and we use the unclipped sentence-level Bleu-4 score with smoothed 3-gram and 4-gram precisions as in [He & Deng 12], which we denote as $\beta(E, \hat{E})$ with respect to the reference translation $\hat{E}$. Given a corpus $\{(E_1, F_1), \ldots, (E_n, F_n), \ldots, (E_N, F_N)\}$, the normalized expected Bleu score under parameter set Θ with respect to the posterior probability distribution $p_{\Theta}(E|F)$ is defined as

$$\langle \beta \rangle_{\Theta} = \frac{1}{N} \sum_{n=1}^{N} \sum_{E \in \mathbb{E}_{\Theta}(F_n)} p_{\Theta}(E|F_n) \beta(E, E_n) \tag{6.17}$$

Here, we compute the sum over all target sentences E in an n-best list $\mathbb{E}_{\Theta}(F)$ generated by the decoder, which is a subset of the most likely hypotheses with respect to the parameterized probability $p_{\Theta}(E|F)$. We write $\langle \cdot \rangle$ to denote the expectation. The parameter set Θ corresponds to the discriminative models introduced in Section 4.2.10. Each of the feature types defines a large number of sparse features $\vartheta \in \Theta$.

The normalized posterior translation probability $p_{\Theta}(E|F)$ from source sentence F to target sentence E was defined in Equation 3.3. We approximate the denominator with an n-best list $\mathbb{E}_{\Theta}(F)$ and define the total score under parameter set Θ as $f_{\Theta}(E, F)$. Further, we introduce a scaling factor $\omega \geq 0$, which accounts for the fact that the model weights are optimized for ranking hypotheses and thus do not necessarily represent a reasonable probability distribution. $\omega \ll 1$ results in a smooth distribution, where in the extreme case $\omega = 0$ the n-best hypotheses are equally distributed. With $\omega \gg 1$, on the other hand, we obtain a peaked distribution where most of the probability mass is assigned to few hypotheses at the top of the n-best list. Unless

otherwise specified, we set $\omega = 1$, which corresponds to Equation 3.3. We obtain

$$p_\Theta(E|F) = \frac{\exp(\omega \cdot f_\Theta(E,F))}{\sum_{E' \in \mathbb{E}_\Theta(F)} \exp(\omega \cdot f_\Theta(E',F))} \tag{6.18}$$

$$f_\Theta(E,F) = \sum_{m \in M} \lambda_m h_{m,\Theta}(E,F) \tag{6.19}$$

Note that the models $h_{m,\Theta}(\cdot,\cdot)$ depend on the parameterization, which is why we add Θ as an index. Further, we apply the maximum approximation as in Equation 4.10 and for ease of notation we assume the hidden segmentation and alignment s_1^K to be implicitly included in the target sentence variable E.

Maximum entropy models tend to generalize poorly, so we need to apply regularization. [He & Deng 12] use Kullback-Leibler regularization, raising the need of having normalized models $h_{m,\Theta}(E,F)$. We employ the more general L_2-regularization and the objective function $O(\Theta)$ is defined as

$$O(\Theta) = log\langle\beta\rangle_\Theta - \tau \cdot \sum_{\vartheta \in \Theta} \vartheta^2 \tag{6.20}$$

including the hyper-parameter τ that controls the degree of regularization. The derivative of the objective function, which is is required for optimization with gradient-based training methods, directly follows:

$$\frac{\partial O(\Theta)}{\partial \vartheta} = -\tau \cdot 2\vartheta + \frac{1}{\langle\beta\rangle_\Theta} \cdot \frac{\partial \langle\beta\rangle_\Theta}{\partial \vartheta} \tag{6.21}$$

We denote the number of times that feature ϑ fires in the derivation for translation hypothesis E given source sentence F as $\#_\vartheta(E,F)$. For each set of trained features, we introduce a new model $h_{m_\vartheta,\Theta}(\cdot,\cdot)$ into the log-linear combination (Equation 6.19). Without loss of generality we set $\lambda_{m_\vartheta} = 1$. With

$$\frac{\partial f_\Theta(E,F)}{\partial \vartheta} = \lambda_{m_\vartheta} \#_\vartheta(E,F) = \#_\vartheta(E,F), \tag{6.22}$$

the derivative of $p_\Theta(E|F)$ is defined as

$$\begin{aligned}
&\frac{\partial p_\Theta(E|F)}{\partial \vartheta} = \\
&\frac{e^{\omega \cdot f_\Theta(E,F)} \cdot \omega \cdot \frac{\partial f_\Theta(E,F)}{\partial \vartheta} \cdot \sum_{E' \in \mathbb{E}_\Theta(F)} e^{\omega \cdot f_\Theta(E',F)} - e^{\omega \cdot f_\Theta(E,F)} \cdot \sum_{E' \in \mathbb{E}_\Theta(F)} \left(e^{\omega \cdot f_\Theta(E',F)} \cdot \omega \cdot \frac{\partial f_\Theta(E',F)}{\partial \vartheta} \right)}{\left(\sum_{E' \in \mathbb{E}_\Theta(F)} e^{\omega \cdot f_\Theta(E',F)} \right)^2} \\
&= \omega \cdot \frac{e^{\omega \cdot f_\Theta(E,F)}}{\sum_{E' \in \mathbb{E}_\Theta(F)} e^{\omega \cdot f_\Theta(E',F)}} \cdot \left(\#_\vartheta(E,F) - \sum_{E' \in \mathbb{E}_\Theta(F)} \left(\frac{e^{\omega \cdot f_\Theta(E',F)}}{\sum_{E'' \in \mathbb{E}_\Theta(F)} e^{\omega \cdot f_\Theta(E'',F)}} \cdot \#_\vartheta(E',F) \right) \right) \\
&= \omega \cdot p_\Theta(E|F) \cdot \left(\#_\vartheta(E,F) - \sum_{E' \in \mathbb{E}_\Theta(F_n)} p_\Theta(E'|F) \#_\vartheta(E',F) \right)
\end{aligned} \tag{6.23}$$

Finally, the derivative of the expected BLEU can be computed as

$$\begin{aligned}\frac{\partial\langle\beta\rangle_\Theta}{\partial\vartheta} &= \frac{1}{N}\sum_{n=1}^{N}\sum_{E\in\mathbb{E}_\Theta(F_n)}\beta(E,\hat{E}(F_n))\frac{\partial p_\Theta(E|F_n)}{\partial\vartheta} \\ &= \frac{\omega}{N}\sum_{n=1}^{N}\Bigg(\sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\beta(E,\hat{E}(F_n))\#_\vartheta(E,F_n) \; - \\ &\quad \Big(\sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\beta(E,\hat{E}(F_n))\Big)\cdot\Big(\sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\#_\vartheta(E,F_n)\Big)\Bigg)\end{aligned} \tag{6.24}$$

By using local expectations $\langle\cdot\rangle_n$ of the BLEU score and the feature count $\#_\vartheta$ we obtain a more compact notation:

$$\frac{\partial\langle\beta\rangle_\Theta}{\partial\vartheta} = \frac{\omega}{N}\sum_{n=1}^{N}(\langle\beta\#_\vartheta\rangle_n - \langle\beta\rangle_n\langle\#_\vartheta\rangle_n) \tag{6.25}$$

with

$$\langle\beta\rangle_n = \sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\beta(E,\hat{E}(F_n)) \tag{6.26}$$

$$\langle\#_\vartheta\rangle_n = \sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\#_\vartheta(E,F_n) \tag{6.27}$$

$$\langle\beta\#_\vartheta\rangle_n = \sum_{E\in\mathbb{E}_\Theta(F_n)}p_\Theta(E|F_n)\beta(E)\#_\vartheta(E,F_n) \tag{6.28}$$

To obtain common factors that can be used by all parameter updates, we move $\#_\vartheta$ to the front of the equation in the implementation:

$$\frac{\partial\langle\beta\rangle_\Theta}{\partial\vartheta} = \frac{\omega}{N}\sum_{n=1}^{N}\sum_{E\in\mathbb{E}_\Theta(F_n)}\#_\vartheta(E,F_n)\cdot p_\Theta(E|F_n)\Big(\beta(E,\hat{E}(F_n)) - \langle\beta\rangle_n\Big) \tag{6.29}$$

Note: Different from the definition in this work, in many implementations the models $h_{m,\Theta}(\cdot,\cdot)$ are defined as *negative* log probabilities. As a result, the posterior probability $p_\Theta(\cdot,\cdot)$ (cf. Equation 6.18) is computed with negative exponents. This carries over to the derivative, which subsequently switches sign.

6.3.4 Efficient Implementation

The expected BLEU $\langle\beta\rangle_\Theta$ can be computed efficiently in a single pass over the full n-best list. As can be seen from Equation 6.29, the derivative $\frac{\partial\langle\beta\rangle_\Theta}{\partial\vartheta}$ is additive with respect to each firing instance of feature ϑ in the n-best list. The additive factor solely depends on the sentence pair that is currently being processed. Therefore, for each sentence of the training data we need to iterate through its n-best list twice. In the first pass, we compute the expectation of the sentence-level BLEU score $\langle\beta\rangle_n$. In the second pass we update the derivative for each time the feature fires. The only thing we need to keep in memory is a list of the current derivatives for each parameter ϑ.

1 Create the baseline system and run MERT to obtain the model weights λ_m
2 Generate n-best lists on the training corpus
3 Compute sentence-level BLEU $\beta(E_n)$ for each hypothesis E_n in the lists
4 Initialize parameters with $\vartheta = 0, \forall \vartheta \in \Theta$, $t = 0$
5 WHILE $t < t_{max}$:
6 $t = t + 1$
7 Compute the derivatives $\frac{\partial O(\Theta)}{\partial \vartheta}$
8 Perform parameter update and output $\Theta^{(t)}$
9 Run MERT on `dev` with each table $\Theta^{(t)}$
10 Select the best $\Theta^{(t)}$ on `dev`
11 Evaluate on the test sets

Figure 6.4: The complete maximum expected BLEU training algorithm.

6.3.5 Complete Training Algorithm

The complete training and evaluation procedure is summarized in Figure 6.4. We start by building a baseline translation system with MERT-optimized model weights λ_m. With the baseline system we generate n-best lists on the training data. Now, for each translation hypothesis E_n of the n-best list, we compute the sentence-level BLEU score $\beta(E_n, \hat{E}(F_n))$ and initialize the parameter set to $\vartheta = 0, \forall \vartheta \in \Theta$. Next, we run the training algorithm for a fixed maximum number of iterations and output the updated feature values $\Theta^{(t)}$ after each iteration t. Note that different from [He & Deng 12] we keep the λ_m weights fixed throughout all iterations of maximum expected BLEU training. Finally, we run MERT with each $\Theta^{(t)}$, select the best table on `dev` and evaluate on our test sets. To facilitate this, all sparse features of one type (each type corresponds to one of the Equations 4.34 through 4.49) are condensed into a single dense model $h_m(\cdot, \cdot)$ within the log-linear model combination.

6.4 Contributions

The generative forced alignment training methods described in Section 6.2.1 are based on initial work during the course of the author's diploma thesis [Wuebker 09] and were previously published in [Wuebker & Mauser$^+$ 10]. Here, the main idea is due to the author's diploma thesis supervisor Arne Mauser while the implementation and experiments were performed by the author of this thesis, with the exception of the forward-backward model (cf. Section 6.2.3), which was implemented by Arne Mauser. Its refinements and the length-incremental training algorithm were developed by the author of this dissertation and published in [Wuebker & Ney 12a, Wuebker & Hwang$^+$ 12, Wuebker & Ney 13]. The author is also responsible for the entire implementation into RWTH Aachen University's open source toolkit *Jane*.

The discriminative training procedure using the RPROP algorithm detailed in Section 6.3 was joint work of the author of this thesis with Sebastian Mühr and Patrick Lehnen. The author contributed the original idea, while Patrick Lehnen suggested using the RPROP algorithm, L_2-regularization and condensation into separate log-linear models. The initial implementation is due to Sebastian Mühr during his Master Thesis under the author's supervision. The author refined and extended the implementation for increased memory and time efficiency, as well as improved usability, in order to allow large training sets and make the software easier to maintain and extend. The author further added the SGD and AdaGrad update algorithms for comparative experiments.

6.5 Related Work

6.5.1 Generative Training

The first attempt to generative translation modeling was the joint probability model poposed by [Marcu & Wong 02]. It is trained via a hill-climbing technique that uses break, merge, swap and move operations. Due to the computational complexity the authors only consider phrases with at least five occurrences in the training data. This model was shown to slightly underperform standard heuristic extraction in [Koehn & Och$^+$ 03]. [Birch & Callison-Burch$^+$ 06] present a variant which is constrained by a word alignment for higher efficiency. A different training procedure for this model based on a Gibbs sampler is introduced in [DeNero & Buchard-Côté$^+$ 08]. The initialization depends on the word alignment.

The fully generative phrase model presented by [DeNero & Gillick$^+$ 06] is trained with the expectation-maximization (EM) algorithm. The authors show that the model abuses the hidden segmentation variable to over-fit to the data, resulting in an inferior translation quality compared to the heuristic baseline.

In [Ferrer & Juan 09], the authors train a conditional "inverse" phrase model by a semi-hidden Markov model. In addition to the phrases, they model the segmentation sequence that is used to produce a phrase alignment. To counteract over-fitting, the phrase model is interpolated with IBM Model 1 probabilities. Our work also includes the lexical smoothing scores in the forced alignment training procedure. [Ferrer & Juan 09] show that Viterbi training produces almost the same results as full Baum-Welch training, reporting improvements over a simplified phrase-based model using only an inverse phrase model and a language model. Similar to [Ferrer & Juan 09] we also compare Viterbi training with the forward-backward algorithm. Different from [Ferrer & Juan 09], however, we apply a full competitive translation system including all standard models.

[Moore & Quirk 07a] also describe an iteratively-trained phrase model. It is segmentation-free and thus can close the performance gap to phrase tables induced from surface heuristics, which confirms the findings by [DeNero & Gillick$^+$ 06], who claimed that the hidden segmentation variable was the main problem. The phrase model introduced by [Mylonakis & Sima'an 08] uses prior probabilities based on inversion transduction grammar (ITG) and smoothing as a learning objective to prevent over-fitting. Both methods rely on the word alignment for phrase pair selection. [Saers & Wu 11] present an EM algorithm for principled induction of bilexica based on linear inversion transduction grammar (LITG). The model by itself underperforms the baseline, but similar to our results, the authors report moderate improvements by combining it with the baseline phrase table. A different probabilistic ITG model is proposed in [Neubig & Watanabe$^+$ 11]. It performs direct phrase table extraction from unaligned data with results similar to the heuristic baseline on several tasks.

[Blunsom & Cohn$^+$ 09] use a Gibbs sampler to perform inference over latent synchronous derivation trees with a non-parametric Bayesian model. The training procedure is initialized by extracting rules from a word alignment, but the authors allow the sampler to diverge from the initial values by a burn-in phrase of 1000 passes over the data. The model is used to generate a hierarchical alignment, on which the usual extraction heuristics are applied to infer a distribution over rule tables.

A number of different models that are trained from forced derivation trees are presented by [Duan & Li$^+$ 12]. They include a re-estimated translation model, two reordering models and a rule sequence model. For inference the parameters are optimized towards alignment F-score. The standard heuristic extraction scheme is used for initialization.

[Feng & Cohn 13] propose another generative, word-based Markov chain translation model, where the authors exploit a hierarchical Pitman-Yor process for smoothing. However, in their

experiments, it is only applied to induce word alignments which in turn were used to extract phrase pairs. Additionally, their experiments were performed on top of weak baselines.

6.5.2 Discriminative Training

Discriminative training is one of the most active research areas in statistical machine translation and it can be integrated into the pipeline at various stages.

Minimum error rate training (MERT) [Och 03] is a discriminative method to optimize the different feature weights in the log-linear model combination on a small development data set. While only capable of optimizing a handful of features, it has proven hard to beat and is still used in many state-of-the-art translation systems. The margin infused relaxed algorithm (MIRA) [Watanabe & Suzuki$^+$ 07, Chiang & Marton$^+$ 08] and the pairwise ranking optimization (PRO) [Hopkins & May 11] were proposed more recently and can replace MERT in the tuning pipeline and scale to thousands of parameters.

Following a different line of work, [Liang & Buchard-Côté$^+$ 06] describe an end-to-end discriminatively trained SMT system. They use a perceptron-style update algorithm to tune more than a million features on the bilingual training data. The Direct Translation Model 2 introduced by [Ittycheriah & Roukos 07] also trains millions of features on the training bitext. However, the underlying translation paradigm is different from the standard phrase-based model and the weights are estimated with a maximum entropy model. In [Gao & He 13], Markov random field models for phrase translation are trained with gradient ascent. These models are interpreted as undirected phrase compatibility scores rather than translation probabilities. As a result they are not subject to a sum-to-one constraint, similar to this work. [Simianer & Riezler$^+$ 12] propose a distributed algorithm for large-scale joint discriminative training and feature selection. The authors divide the training corpus into several shards, on which features are updated via perceptron-style gradient descent. The trained weight vector is then averaged and the most informative features across all shards are identified by an l_1/l_2 regularization term. The results show that training on large data sets improves translation quality over using a small development corpus. Another approach that scales to a large number of sparse features and is based on the AdaGrad algorithm was proposed in [Green & Wang$^+$ 13, Green & Cer$^+$ 14]. Different from this work, the authors use either the tuning set or a small subsample of the training data (15k sentences) for discriminative training.

A notably different idea is pursued by [Yu & Huang$^+$ 13]. The authors present a large-scale training procedure that explicitly minimizes search errors, which is achieved by force-decoding the training data and updating when the correct derivation drops off the beam.

[Blunsom & Cohn$^+$ 08] propose to train conditional random field (CRF) features with the L-BFGS [Nocedal & Wright 99] updating scheme and apply it to a hierarchical phrase-based decoder. In a similar way [Lavergne & Allauzen$^+$ 11] use a CRF to estimate phrasal translation scores for the n-gram based approach to SMT [Casacuberta & Vidal 04, Mariño & Banchs$^+$ 06]. As in our work, model updates are carried out by the RPROP algorithm. Both approaches only improve over constrained baselines.

This work is mainly inspired by [He & Deng 12, Setiawan & Zhou 13]. Here, the standard phrasal and lexical channel models are trained with the growth transformation (GT) algorithm. The authors use n-best lists on the training data to approximate the hypothesis space and optimize a maximum expected Bleu objective. Applying the same objective function to reordering features, [Auli & Galley$^+$ 14] report good results training with stochastic gradient descent (SGD).

Our work differs in several key aspects:

(i) We propose to apply the RPROP algorithm for maximum expected Bleu training. In our experimental comparison provided in Section 7.4, it yields superior results compared with GT, SGD and AdaGrad.

(ii) We apply maximum expected BLEU training on the largest data set reported in the literature, namely four million sentence pairs.

(iii) We apply a leave-one-out heuristic [Wuebker & Mauser$^+$ 10] to make better use of the training data.

(iv) We apply more feature types than previous work: phrasal, lexical, reordering and triplet features.

(v) Finally, we refrain from running MERT after each training iteration, which can be expensive for large translation systems.

7. Experimental Evaluation

All evaluated techniques are carried out with *Jane*, which is RWTH Aachen University's open-source toolkit for statistical machine translation. It is described in detail in [Vilar & Stein$^+$ 10, Stein & Vilar$^+$ 11, Wuebker & Huck$^+$ 12, Freitag & Huck$^+$ 14].

7.1 Language Model Look-Ahead

Here we will describe the experimental evaluation of the language model look-ahead techniques introduced in Section 5.3 as well as phrase pre-sorting (cf. Section 5.2). With the exception of hybrid look-ahead results, this material was published in [Wuebker & Ney$^+$ 12b].

7.1.1 Setup

We perform experiments on the WMT 2011 German→English task. The data is described in Section A.1.2. A word alignment is trained with GIZA++ [Och & Ney 03] on all available bilingual data. The English language model is a 4-gram language model with modified Kneser-Ney smoothing [Kneser & Ney 95] created with the SRILM toolkit [Stolcke 02] on all bilingual data and parts of the provided monolingual shuffled news data. To confirm our results, we run the final set of experiments also on the English→French task of the IWSLT 2011 shared task (cf. Section A.2.1).

We compare our novel pruning techniques implemented in *Jane* with the *Moses* decoder [Koehn & Hoang$^+$ 07b], the most popular open source toolkit for statistical machine translation. For a fair comparison, we use identical phrase tables and scaling factors for Moses and Jane. The phrase table is pruned to a maximum of 400 target candidates per source phrase before decoding. To evaluate the pure algorithm independently of disk access, the phrase table and language model are loaded into memory before translating. For translation speed measurements, loading time is eliminated. We use the standard settings in Moses.

7.1.2 Effects on Search

To observe the effect of the heuristic language model score pre-sorting of the phrase candidates and the two language model look-ahead methods, we first fixed the pruning parameters to $N_r = N_l = 8$ and ran a set of experiments on the test set. In addition to translation quality in Bleu and speed in words per second, we kept track of the number of hypothesis expansions and language model computations. The results are given in Table 7.1.

As was noted in Section 5.2, the language model score pre-sorting affects both the set of phrase candidates due to observation histogram pruning and the order in which they are considered. To separate these effects, experiments were run both with and without histogram pruning ($N_o = 100$). In both cases we observe similar improvements over the baseline, which performs pre-sorting with respect to the translation model scores only. The number of hypothesis expansions is reduced by

Table 7.1: Comparison of the number of hypothesis expansions per source word (# HYP) and language model computations per source word (# LM) with respect to language model pre-sorting, first-word language model look-ahead, phrase-only language model look-ahead and hybrid language model look-ahead on `newstest2009`. Speed is given in words per second. Results are presented with ($N_o = 100$) and without ($N_o = \infty$) observation pruning.

N_o	pre-sort	look-ahead	BLEU [%]	# HYP	# LM	words/sec
∞	no	none	20.1	3.0K	322K	2.2
	yes		20.1	2.5K	183K	3.6
100	no	none	19.9	2.3K	119K	7.1
	yes	none	20.1	1.9K	52K	15.8
		first-word	20.1	1.9K	40K	31.4
		phrase-only	19.8	1.6K	6K	69.2
		hybrid	20.0	1.8K	20K	61.1

∼20%, the number of language model lookups by 43% to 56% and translation speed by a factor of 1.6 to 2.2. When histogram pruning is applied, we additionally see a small increase by 0.2% in BLEU resulting from selecting phrase candidates with better language model score estimates.

Application of first-word language model look-ahead further reduces the number of language model lookups by 23%, resulting in an increase in translation speed by a factor of two. This can partly be attributed to fewer trie node searches as described in Section 5.3. The heuristic phrase-only language model look-ahead method introduces additional search errors, resulting in a BLEU drop by 0.3%, but yields another 85% reduction in language model computations and increases throughput by a factor of 2.2. The hybrid approach is able to regain most of the loss in BLEU with only a small difference in translation speed compared to phrase-only look-ahead. In comparison with first-word look-ahead, the number of language model lookups is reduced by 50%.

7.1.3 Performance

We evaluated our proposed extensions to the original beam search algorithm in terms of scalability and their usefulness for different application constraints. Real world applications might, for example, prioritize either translation quality or translation speed, or prefer balancing the two extreme cases.

We compare Moses and five different setups of our decoder. The plain baseline is first extended by adding language model score pre-sorting. On top of this system, we separately evaluate the three language model look-ahead methods. For all setups we translated the test set with the beam sizes $N_r = N_l = \{1, 2, 4, 8, 16, 24, 32, 48, 64\}$. For Moses we used the beam sizes $2^i, i \in \{1, \ldots, 9\}$. Observation histogram pruning is fixed to $N_o = 100$ for both decoders. Figure 7.1.2 plots translation performance in BLEU against translation speed. Without the proposed extensions, Moses reaches a slightly higher BLEU score than our decoder. However, the latter already scales better for higher speed, which confirms observations made in [Zens & Ney 08]. By including language model score pre-sorting in Jane, we perform similar to Moses while further accelerating translation. With the same BLEU score as Moses with 1.8 words/sec, Jane's translation speed is 16 words/sec. Adding first-word language model look-ahead shifts the graph further to the right, now reaching the same BLEU performance at 31 words/sec. Fixing translation speed at roughly 70 words/sec, our approach yields 20.0% BLEU compared to 17.2% BLEU with Moses. For phrase-only language

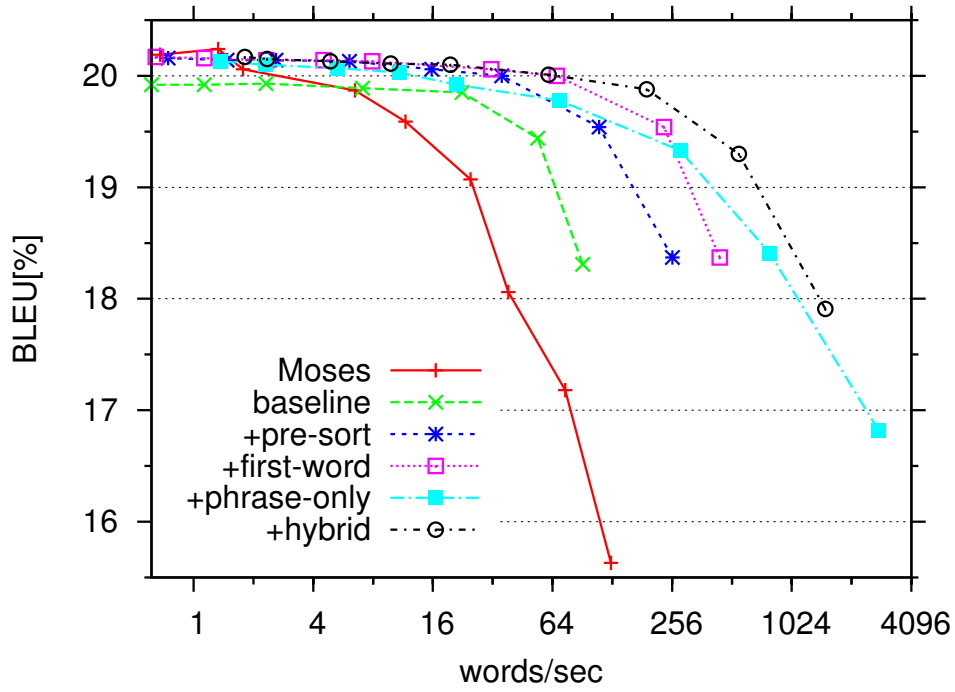


Figure 7.1: Translation performance in BLEU [%] vs. speed on the German→English `newstest2009` set on a logarithmic scale. We compare Moses with our approach without language model look-ahead and language model score pre-sorting (baseline), with added language model pre-sorting (+pre-sort) and with either first-word or phrase-only language model look-ahead on top of using pre-sorting. Observation histogram size is fixed to $N_o = 100$ for both decoders.

model look-ahead the graph is somewhat flatter. It yields a slightly lower top performance while showing a better trade-off between translation quality and speed. Finally, hybrid language model look-ahead outperforms all previous approaches in terms of the efficiency/quality trade-off.

We performed a final set of experiments on both the WMT and the IWSLT task. Our decoder with the three language model look-ahead methods is directly compared with Moses in four scenarios: The best possible translation, the fastest possible translation without performance constraints and the fastest possible translation with no more than 1% and 2% loss in BLEU on the development set compared to the best value. These results are given in Table 7.2. On the WMT data, both decoders have a similar top performance. Here, hybrid look-ahead is already 2.5 times faster than Moses. If we allow for a small degradation in translation quality, our approaches outperform Moses in terms of translation speed by a large margin. With hybrid language model look-ahead, our decoder is faster by a factor of 16 for no more than 1% BLEU loss, a factor of 22 for 2% BLEU loss and in the fastest setting, phrase-only language model look-ahead is faster than Moses by a factor of 22. The results on the IWSLT data are very similar. Here, the speed difference reaches a factor of 6 for the best quality and a factor of 19 in the fastest setting.

Table 7.2: Comparison of Moses with this work. All three language model look-ahead look-ahead techniques are evaluated. We consider both the best and the fastest possible translation, as well as the fastest settings resulting in no more than 1% and 2% BLEU loss on the development set.

setup	**system**	**WMT 2011 German→English**				**IWSLT 2011 English→French**			
		beam size (N_c, N_l)	**speed** words/sec	**BLEU** [%]	**TER** [%]	**beam size** (N_c, N_l)	**speed** words/sec	**BLEU** [%]	**TER** [%]
best	Moses	256	0.7	20.2	63.2	16	10	29.5	52.8
	this work: first-word	(48,48)	1.1	20.2	63.3	(8,8)	23	29.5	52.9
	phrase-only	(64,64)	1.4	20.1	63.2	(16,16)	18	29.5	52.8
	hybrid	(64,64)	1.8	20.2	63.2	(8,8)	58	29.6	52.8
BLEU: $\geq$ **-1%**	Moses	16	12	19.6	63.7	4	40	29.1	53.2
	this work: first-word	(4,4)	67	20.0	63.2	(2,2)	165	29.1	53.1
	phrase-only	(8,8)	69	19.8	63.0	(4,4)	258	29.3	52.9
	hybrid	(4,4)	191	19.9	62.9	(2,2)	498	29.2	52.9
BLEU: $\geq$ **-2%**	Moses	8	25	19.1	64.2	2	66	28.1	54.3
	this work: first-word	(2,2)	233	19.5	63.4	(1,1)	525	28.4	53.9
	phrase-only	(4,4)	280	19.3	63.0	(2,2)	771	28.5	53.2
	hybrid	(2,2)	555	19.3	63.2	(1,1)	1.4K	27.8	54.0
fastest	Moses	1	126	15.6	68.3	1	116	26.7	55.9
	this work: first-word	(1,1)	444	18.4	64.6	(1,1)	525	28.4	53.9
	phrase-only	(1,1)	2.8K	16.8	64.4	(1,1)	2.2K	26.4	54.7
	hybrid	(1,1)	1.5K	17.9	64.1	(1,1)	1.4K	27.8	54.0

7.2 Word Class Models

7.2.1 Setup

To evaluate the class-based smoothing models introduced in Section 4.2.9, we perform experiments on the Quaero 2012 French→German task and the IWSLT 2012 German→English task. Data statistics and descriptions are given in Sections A.3.1 and A.2.3, respectively. For the French→German task, we make use of a standard phrase-based system augmented with Galley's hierarchical reordering model (cf. Section 4.2.8). The language model is a 4-gram LM trained on all German monolingual sources provided for the *NAACL 2012 Seventh Workshop on Statistical Machine Translation*[1]. To train class-based models, we run `mkcls` [Och 99] separately on the source and target side of the bitext and cluster the vocabulary into 100 classes each. This class mapping is used to train class-based versions of the phrase translation models in both directions, the lexical translation models in both directions as well as the hierarchical reordering model. We further build a 4-gram word class language model trained on the same data as the baseline language model. Due to the smaller vocabulary we can also efficiently experiment with a 7-gram context length for the word class language model in additional experiments. Finally, we evaluate with larger numbers of classes, namely 200 and 500.

On the German→English task, the class-based smoothing models are evaluated on both a standard phrase-based and a hierarchical phrase-based baseline. Here, the phrase-based baseline is also augmented with the hierarchical reordering model. We limit the training data to the in-domain portion, the *TED* talks, for rapid experimentation. The number of classes is 100. The 4-gram baseline language model is trained on the *TED*, *Europarl* and *news-commentary* corpora. We directly use a word class language model with 7-gram context size.

Feature weights are optimized with minimum error rate training in all setups. The confidence level computation in our results is based on [Koehn 04].

7.2.2 Results

Table 7.3 reports the perplexities for different German language models that were used in the Quaero task. The baseline 4-gram language model has a perplexity of 180.1. The word class language models are worse by more than a factor of two, with values of 397.4 for the 4-gram and 384.4 for the 7-gram language model. This is consistent with the translation results we obtain by replacing the baseline language model, which are described later in this paragraph (cf. Table 7.4). A linear interpolation with the class-based language models, however, leads to an improvement in perplexity over the baseline by 2.4% with the 4-gram word class language model and by 3.5% with the 7-gram word class language model. This is also reflected in the translation quality, where we observe absolute improvements of 0.7% and 1.0% BLEU. Note that these perplexities computed for linear interpolation do not directly correspond to the translation results, which are generated with log-linear interpolation. Due to the necessary re-normalization overhead, computing perplexities for log-linear interpolation was infeasible.

French→German translation results are shown in Table 7.4. In a first set of experiments we replaced a subset of the standard models by their class-based counterparts. Here, *TM* (translation model) denotes the two phrasal and lexical channel models, *LM* denotes the language model and *HRM* the hierarchical reordering model. *wcTM*, *wcLM* and *wcHRM* denote the corresponding class-based smoothing models. Unsurprisingly, this replacement degrades performance with different levels of severity. The strongest drop in automatic scores can be seen when replacing TM (*-TM +wcTM*), while we observe only a small performance gap with *-HRM +wcHRM*. The performance drop when replacing the language model (*-LM +wcLM*) reflects the large perplexity

[1] `http://statmt.org/wmt12/`

Table 7.3: Perplexities for the different German language models. The +-operator denotes interpolation. Note that perplexity is computed with linear interpolation while the translation results are with log-linear interpolation directly in the decoder. Perplexities for log-linear interpolation were infeasible to compute due to the additional overhead for re-normalization.

language model	**perplexity**	dev		test	
		Bleu [%]	**Ter** [%]	**Bleu** [%]	**Ter** [%]
baseline	180.1	24.6	61.8	27.8	57.6
4-gram wcLM	397.4	22.2	62.9	25.9	58.9
7-gram wcLM	384.4	22.6	62.3	25.8	59.0
baseline + 4-gram wcLM	175.8	24.7	61.2	28.5	56.9
baseline + 7-gram wcLM	173.8	25.0	61.1	28.8	57.0

Table 7.4: Bleu and Ter results on the French→German Quaero task. Results marked with ‡ are statistically significant with 95% confidence, results marked with † are statistically significant with 90% confidence. - *X* + *wcX* denote the systems, where the model *X* in the baseline is replaced by its word class counterpart. $wcModels_k$ denotes all word class models trained on k classes.

	dev		test	
	Bleu [%]	**Ter** [%]	**Bleu** [%]	**Ter** [%]
- TM + wcTM	21.2	64.2	24.7	59.5
- LM + 4-gram wcLM	22.2	62.9	25.9	58.9
- HRM + wcHRM	24.6	61.9	27.5	58.1
phrase-based	24.6	61.8	27.8	57.6
+ wcTM	24.7	61.4	28.1	57.1
+ 4-gram wcLM	24.9	61.2	28.4	57.1
+ wcHRM	25.4‡	60.9‡	28.9‡	56.9‡
+ 7-gram wcLM	25.5‡	60.7‡	29.2‡	56.6‡
+ $wcModels_{200}$	25.5‡	60.8‡	29.3‡	56.4‡
+ $wcModels_{500}$	25.2†	60.8‡	29.0‡	56.6‡

difference reported in Table 7.3. However, when the class-based models are integrated as additional models into the log-linear combination, translation quality is improved. Adding *wcTM* yields 0.3% Bleu and 0.5% TER absolute on `test`. By adding the 4-gram *wcLM*, we see an additional 0.3% Bleu improvement and *wcHRM* yields a further increase of 0.5% Bleu and 0.2% Ter. Finally, an additional boost is achieved by extending the context length of the word class language model to 7-grams, reaching a total gain over the baseline of 1.4% Bleu and 1.0% Ter. This result is statistically significant on the 95% confidence level. Using 200 classes instead of 100 seems to perform slightly better on `test`. With 500 classes, translation quality degrades again.

Table 7.5 shows the results for the German→English task. The improvements are similar in Ter, but less pronounced in Bleu. By adding all word class models, we gain 0.3% Bleu and 1.1% Ter over the phrase-based baseline. With the hierarchical decoder we observe an increase of 0.3% Bleu and 0.8% Ter by adding *wcTM* and the 7-gram word class language model.

Table 7.5: BLEU and TER results on the German→English task with both a phrase-based and a hierarchical phrase-based translation engine. Results marked with ‡ are statistically significant with 95% confidence, results marked with † are statistically significant with 90% confidence.

	dev		test	
	BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
phrase-based	30.2	49.6	28.6	51.6
+ wcTM	30.2	49.2	28.9	51.3
+ 7-gram wcLM	30.5	48.3‡	29.0	50.6†
+ wcHRM	30.8	48.3‡	28.9	50.5‡
hierarchical phrase-based	29.6	50.3	27.9	52.5
+ wcTM	29.8	50.3	28.1	52.3
+ 7-gram wcLM	30.0	49.8	28.2	51.7

Table 7.6: Comparison of leave-one-out with phrase length restriction on the development set using the Viterbi count model after a single training iteration.

	maximum phrase length	**BLEU** [%]	**TER** [%]
baseline	6	25.7	61.1
no leave-one-out	2	25.2	61.3
	3	25.7	61.3
	4	25.5	61.4
	5	25.5	61.4
	6	25.4	61.7
standard leave-one-out	6	26.4	60.9
length-based leave-one-out	6	26.5	60.6

7.3 Generative Training

7.3.1 Forced Alignment Training

The results presented in this section evaluate the training techniques described in Section 6.2.2 and were already published in [Wuebker & Mauser+ 10] and [Mauser 15]. They will only be summarized here for completeness. The experiments are carried out on the German→English portion of the Europarl data provided for the WMT 2008 shared translation task, which is described in Section A.1.1.

The baseline is created by the standard heuristic phrase extraction from a word alignment generated with GIZA++. This phrase table is also used to initialize generative training. Our phrase-based translation engines use eight dense features: Phrasal and lexical channel models in both translation directions, phrase and word penalties, a 4-gram language model and a distortion penalty. Automatic evaluation is performed in case-sensitive fashion.

We first compare our leave-one-out strategies introduced in Section 6.2.2 with a simple phrase length restriction. Using the count model estimated with a 100-best list, we evaluate the performance of standard and length-based leave-one-out as well as different maximum phrase lengths after a single training iteration. Table 7.6 clearly shows that leave-one-out is the more effective

means to overcome over-fitting. The length-based variant performs slightly better.

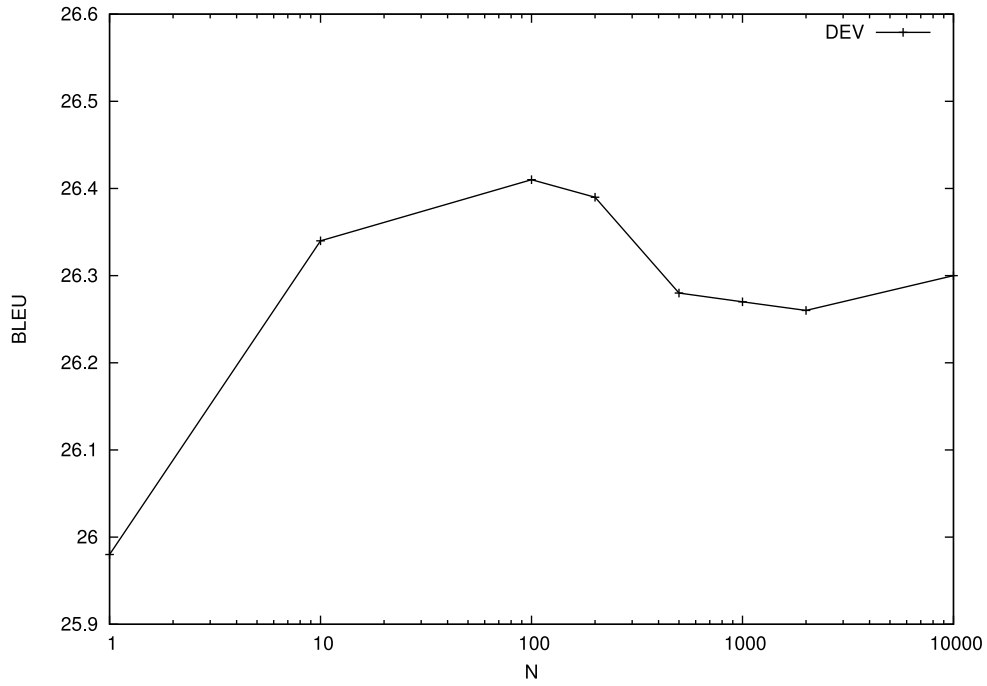


Figure 7.2: Performance of the count model in Bleu [%] after a single training iteration on the development set plotted against size n of the n-best list on a logarithmic scale.

Table 7.7: Phrase table size of the count model with different n-best list sizes, the forward-backward model and the heuristic baseline phrase extraction.

n	**# phrase pairs**	**% of baseline table**
1	4.9M	5.3
10	8.4M	9.1
100	15.9M	17.2
1000	27.1M	29.2
10000	40.1M	43.2
forward-backward	59.6M	64.2
baseline	92.7M	100.0

The size n of the n-best list used to estimate the count model is a hyper-parameter that needs to be selected manually. A smaller n has the advantage of a smaller size of the resulting phrase table. For higher values of n, a larger number of competing alignments is taken into account, resulting in a larger model size but also a smoother distribution. It can be seen from Figure 7.2 that $n = 1$ is insufficient, but the variations in performance between $n = 10$ and $n = 10000$ are smaller than 0.2 Bleu absolute. We observe the maximum at $n = 100$, which we use for subsequent experiments. The sizes of the resulting phrase models are given in Table 7.7. With $n = 100$ we retain 17% of the original phrases. Applying the forward-backward algorithm rather than n-best lists, the model size is 64.2% of the baseline. Our experiments reported in Table 7.8 show that the improved performance of the count model partially derives from the smaller size of

Table 7.8: Final results for generative phrase training on the German→English WMT 2008 Europarl data. *Baseline filtered* denotes the baseline phrase table filtered to contain the same set of phrases as the count model. We also show results with log-linear interpolation, separate features (cf. Section 6.2.3) and a second iteration of the count model using cross-validation, which is initialized with the first iteration using the leave-one-out method. Using separate features adds the learned phrase translation probabilities in both directions to the log-linear model combination, resulting in two additional models.

	dev		test		# log-linear models
	BLEU [%]	TER [%]	BLEU [%]	TER [%]	
baseline	25.7	61.1	26.3	60.9	8
baseline filtered	26.0	61.6	26.9	61.2	
count model	26.5	60.6	27.2	60.5	
count, iteration 2 (cross-validation)	26.4	60.3	27.2	60.0	
forward-backward	26.3	60.0	27.0	60.2	
log-linear interpolation	**27.0**	**59.4**	**27.7**	**59.2**	
separate features	26.8	60.1	27.6	59.9	10

the phrase model. Filtering the baseline phrase table to contain the same phrase pair set as the generatively trained model yields an increase by 0.6% BLEU absolute on the test set. A single iteration of training the count model results in 0.9 BLEU points improvement. Running a second iteration of phrase training results in an additional gain in TER, but no further improvement in BLEU. We used the phrase table trained with leave-one-out in the first iteration to initialize the second iteration, where we applied cross-validation. Additional iterations did not show any further gains. Applying one iteration of the forward-backward algorithm instead of n-best lists is slightly inferior in terms of BLEU, but better in TER. By log-linear feature interpolation we observe our best results of 1.4% BLEU absolute above the baseline. Finally, using separate features fails to beat fixed interpolation weights.

7.3.2 Length-Incremental Training

Several attempts have been made in previous work to do away with heuristics in the statistical machine translation training pipeline and thus close the gap between phrase table generation and translation decoding. Most of these approaches, however, either do not achieve state-of-the-art performance or still rely on the word alignment or extraction heuristics, e.g. as a prior in discriminative training or for initialization of the training procedure. In the experiments described in this chapter, we aim at moving towards a unified framework that induces the phrases based on the same models as in decoding (cf. Section 6.2.4).

The phrase table is trained without using a word alignment or the extraction heuristics. Different from previous work, we are able to generate all possible phrase pairs on-the-fly during the training procedure. A further advantage of our proposed algorithm is that we use a slightly modified version of the beam search applied in standard unconstrained translation. This makes it easy to re-implement by modifying any translation decoder, and ensures that training and translation are consistent. In principle, we apply the forced decoding approach used in the previous section, but initialize the phrase table with IBM-1 lexical probabilities [Brown & Della Pietra$^+$ 93] instead of heuristically extracted relative frequencies. New phrase pairs are generated at training

time as *backoff phrases*. Their size is incremented over the training iterations. The experiments are carried out on the IWSLT 2011 Arabic-English shared task.

Parameterization

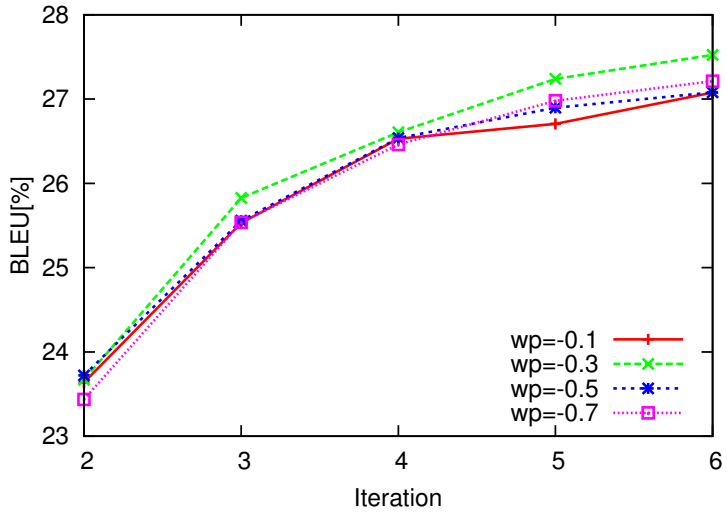
The training procedure has a number of hyper parameters, most of which had little or no impact on the results in our preliminary experiments. We will now describe the parameters that need to be chosen carefully. To successfully align a sentence pair, we require that our decoder fully covers the source sentence. However, we allow for incompletely aligned target sentences in order to achieve a good success rate in terms of number of aligned sentence pairs. The percentage of successfully aligned sentence pairs is denoted as *sentence coverage*. Note that we count a sentence pair as successfully aligned even if the target sentence is not fully covered. The word penalty (wp) feature weight λ_{wp} should be selected with care. A high value leads to a high sentence coverage, but many of the target sides may not be fully aligned. A small word penalty results in a lower sentence coverage, but a higher percentage of the target sentences is aligned. We denote the total percentage of successfully aligned target words as *word coverage*. Note the distinction to the sentence coverage, which is defined above. Figure 7.3 displays BLEU scores and corresponding word coverages for the training iterations 2 through 6 with different values for the word penalty. In the first iteration only one-to-one phrases are allowed, i.e. phrase pairs containing a single word on both source and target side. The number of aligned target words is therefore predetermined and the results are identical. For $\lambda_{wp} = -0.1$, the word coverages are slightly decreasing with each iteration. For $\lambda_{wp} = -0.3$ to $\lambda_{wp} = -0.7$ the word coverage shows a small increase from iteration 2 to 3 and then decreases again. In terms of BLEU score, $\lambda_{wp} = -0.3$ outperforms the other values by a small margin and we continue using this value in all subsequent experiments.

The backoff phrase penalties π_p, π_s and π_t are closely connected with the learning rate of the training procedure. Low penalties result in only few, very good phrases getting an advantage over the ones generated on-the-fly. This corresponds to a slow learning rate. Larger penalties have the effect that a larger percentage of the phrase pairs generated in the previous iterations will be favored over new backoff phrases, resulting in a faster learning rate. We denote phrase pairs that are present in the phrase table, but more expensive than their backoff phrase counterparts as *surplus phrases*. The behavior over the training iterations 2 through 6 with different penalties π_0 in terms of percentage of surplus phrase pairs and BLEU score are presented in Figure 7.4. We set $\pi_s = \pi_t = \pi_0$ and $\pi_p = 5\pi_0$. It can be observed that $\pi_0 = 4$ yields less than 0.1% surplus phrases throughout all iterations, whereas $\pi_0 = 0.5$ starts at 98.2% surplus phrases and decreases to 55.9% in iteration 6. In terms of BLEU, a fast learning rate seems to be preferable. The best results are achieved with $\pi_0 = 3$, where the rate of surplus phrases starts at 6.8% and decreases to 1.7% in iteration 6. In all subsequent experiments, we set the penalty to $\pi_0 = 3$.

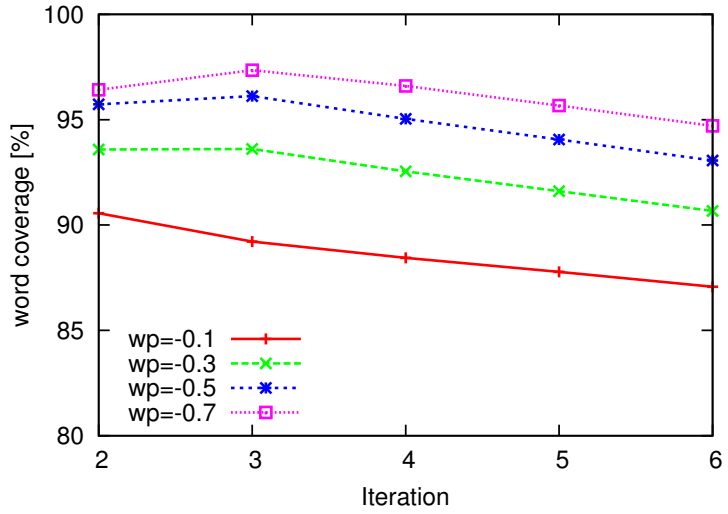
Results

We carry out our experiments on the IWSLT 2011 Arabic→English shared task described in Section A.2.2. The language model is a 4-gram LM trained on all provided in-domain (TED) monolingual data and a selection based on [Moore & Lewis 10] of the out-of-domain corpora. To account for statistical variation, all reported results are average scores over three independent MERT runs.

The baseline phrase table is generated by standard heuristic extraction from a symmetrized word alignment trained with GIZA++ using the IBM-4 model. The maximum phrase length is six words and lexical smoothing scores are computed from IBM-1 probabilities. We use the development set for minimum error rate training and the test set for evaluation. As a second baseline we also perform one iteration of forced alignment training initialized from the baseline,



(a) BLEU

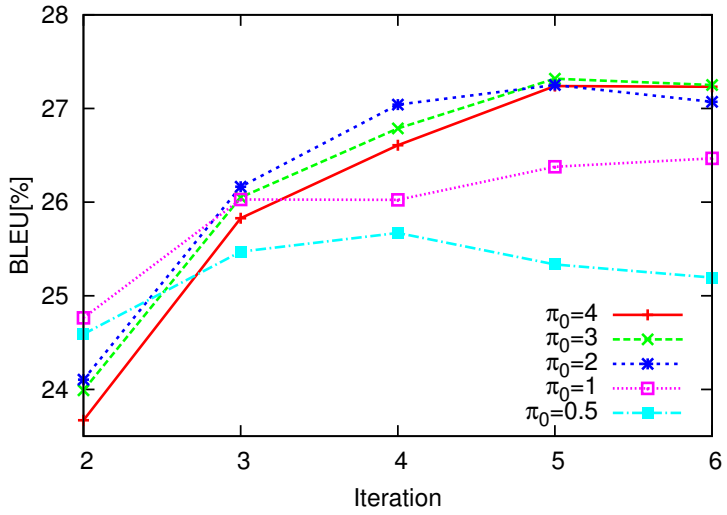


(b) word coverage

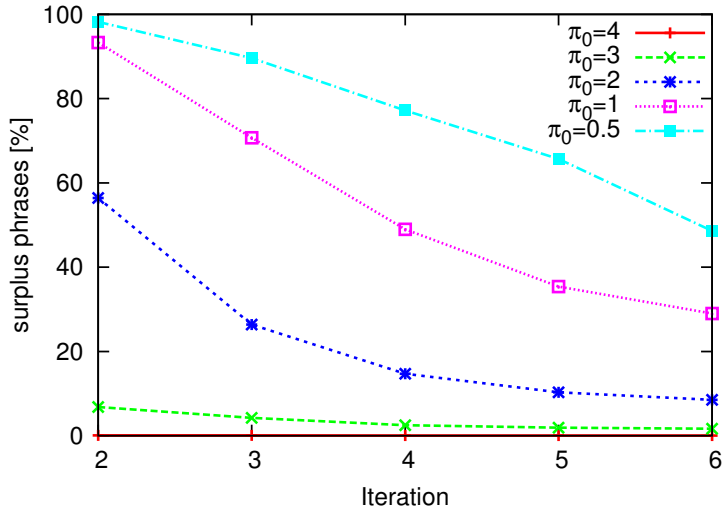
Figure 7.3: BLEU scores and word coverages on `dev` over the first 6 training iterations with different word penalties (wp).

as already applied in Section 7.3.1, denoted as *leave-one-out*. For length-incremental training we apply the algorithm presented in Section 6.2.4. After each iteration, we run minimum error rate training with the resulting phrase table and evaluate on the test set. The set of models used here is identical to the baseline system.

BLEU results are plotted in Figure 7.5. We can see that the performance increases from the first iteration to the fifth iteration, after which only small fluctuations can be observed. Performance on the development set is similar to the baseline. On the test set, the trained phrase tables consistently are slightly better than the baseline. Iteration 12 marks the optimum on the development set. The exact BLEU and TER scores for this iteration as well as for the two baselines are given in Table 7.9. The phrase table trained with leave-one-out performs similar to the heuristic baseline on this task. Length-incremental training is slightly superior to the baseline, yielding an improvement of 0.3% BLEU and 0.4% TER on the test set. With application of the



(a) BLEU



(b) surplus phrases

Figure 7.4: BLEU scores and percentage of surplus phrases on `dev` over the first 6 training iterations with different backoff phrase penalties π_0.

source-length-incremental and target-length-incremental training variants discussed in Section 6.2.4, we observe slightly larger improvements of up to 0.5% BLEU and 0.5% TER over the baseline.

Similar to results observed in [DeNero & Gillick$^+$ 06] and [Wuebker & Mauser$^+$ 10], a linear interpolation of standard length-incremental training with the baseline containing all phrase pairs from either of the two tables, 27.8M in total, yields a moderate improvement of 0.5% BLEU and 0.5% TER on both data sets. By linear interpolation of the baseline table with source-length-incremental training we observe the best results, 0.8% BLEU and 0.9% TER abosolute above the baseline.

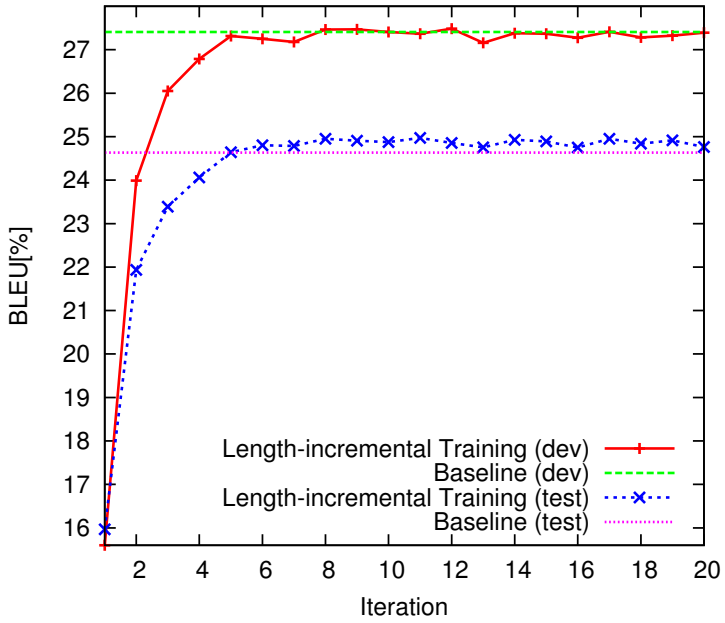


Figure 7.5: BLEU scores for length-incremental training on the development and test corpora over 20 training iterations.

Table 7.9: BLEU and TER scores of the baseline, phrase training with leave-one-out and length-incremental training after 12 iterations (src-len-inc: 10, tgt-len-inc: 5), as well as a linear interpolation of the baseline with length-incremental phrase table.

	dev		test	
	BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
baseline	27.4	54.0	24.6	57.8
leave-one-out	27.3	54.2	24.6	57.7
length-incremental	27.5	53.8	24.9	57.4
+ linear interpolation	27.9	53.5	25.1	57.3
src-length-incremental	27.5	54.0	25.0	57.2
+ linear interpolation	**28.0**	**53.2**	**25.4**	**56.9**
tgt-length-incremental	27.2	54.2	25.1	57.3
+ linear interpolation	27.9	53.5	25.1	57.3

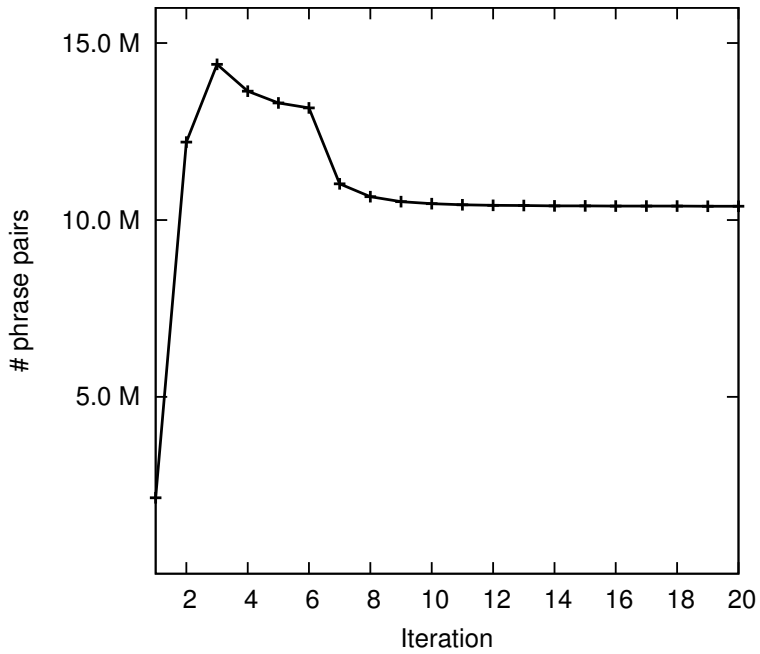


Figure 7.6: Number of generated phrase pairs over 20 training iterations. In the first six iterations, we always generate backoff phrases while incrementing their size. Starting in iteration seven, backoff phrases are only generated in fallback decoding runs, which explains the sudden drop in phrase table size.

Analysis

Figure 7.6 plots the number of phrase pairs present in the phrase tables after each iteration of length-incremental training. In the first six iterations, we keep generating new phrase pairs via backoff phrases. After three iterations we observe the maximum of 14.4M phrase pairs. For comparison, the size of the baseline phrase table is 19.3M phrase pairs. Starting from the seventh iteration, backoff phrases are only generated in fallback decoding runs (cf. algorithm in Figure 6.3). This results in a decrease of the number of phrase pairs that are being used, which levels out at 10.4M phrases.

Taking a look at the phrase length distributions in both the baseline and the trained phrase table shown in Figure 7.7, it can be observed that the latter contains more short phrases, which confirms observations from previous work. In the length-incrementally trained phrase model, phrases of length one and two make up 47% of all phrases. On the other hand, their percentage is only 32% in the baseline table. This trend is even more pronounced in the intersection of the two tables, where 68% of the phrases are of length one and two. This figure resembles the distribution of phrase lengths actually used in decoding as reported in [Guta & Wuebker$^+$ 15], where phrases of length one and two are used to translate 60% of the German→English test set. This could explain the improved performance, but we leave further investigations to future work.

Interestingly, the total overlap between the two phrase tables is rather small. Only about 18.5% of the phrases from the trained table also appear in the heuristically extracted one. This illustrates that generating phrases on-the-fly without restrictions based on a word alignment or a bias from initialization has the effect that our training procedure strongly diverges from the baseline phrase table. We conclude that most previous work in this area, which adhered to the above mentioned restrictions, was only able to explore a fraction of the full potential of real phrase training.

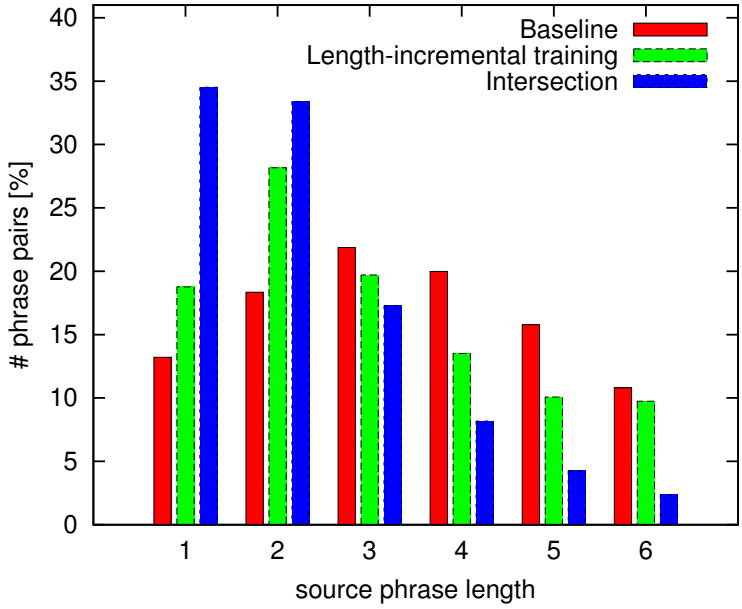


Figure 7.7: Histogram of the phrase lengths present in the phrase tables. The baseline table contains 19.3M phrase pairs, the trained table 10.4M phrase pairs and their intersection 1.9M phrase pairs.

Training Times

As training was not run under controlled conditions we can only estimate roughly how the training times compare between different methods. For a fair comparison, we report the training times on a single machine by summing the times for all parallel and sequential processes.

Heuristic phrase extraction from the word alignment took about 1.7 hours. A single iteration of standard phrase training (leave-one-out) requires about 24 hours. Similarly, the first iteration of length-incremental training and all iterations after the sixth also took roughly 24 hours. The iterations two through six of length-incremental training are considerably more expensive due to the larger size of backoff phrases. Iteration six, with a maximum backoff phrase size of six words on source and target side, was the slowest with around 740 hours.

7.4 Discriminative Training

7.4.1 Experimental Setup

We carry out the main experiments on the IWSLT 2013 German→English task (cf. Section A.2.4). For rapid experimentation and in order to avoid domain adaptation effects, we use the in-domain training data both for translation model training and for our maximum expected BLEU procedure. The language model is a large 4-gram language model with modified Kneser-Ney smoothing [Kneser & Ney 95, Chen & Goodman 98], trained with the SRILM toolkit [Stolcke 02]. The complete News Commentary, Europarl v7 and Common Crawl corpora as well as selected parts of the Shuffled News and LDC English Gigaword corpora are used as additional monolingual data sources for language model training. The data selection is based on cross-entropy difference [Moore & Lewis 10]. In total, the language model is trained on 1.7 billion running words. The baseline further contains a hierarchical reordering model [Galley & Manning 08] and a 7-gram word class language model [Wuebker & Peitz$^+$ 13b]. On this task, all reported results are aver-

Table 7.10: Results for the IWSLT 2013 German→English task. For the comparison between update strategies, we limit the feature set to phrase pair features only (Equation 4.34). The final row additionally leverages the word pair, triplet and discriminative hierarchical reordering model features given in Equations 4.35 through 4.39. Growth transformation (GT), stochastic gradient descent (SGD), AdaGrad and RPROP are trained with leave-one-out, unless otherwise specified. Statistically significant improvements over the baseline on the 99% level are printed in boldface.

	# features	dev		test	
		BLEU [%]	TER [%]	BLEU [%]	TER [%]
baseline	18	32.1	47.2	30.4	49.2
GT [He & Deng 12]	6.08M	**32.7**	**46.8**	**30.9**	**48.9**
SGD [Auli & Galley+ 14]	921K	**32.6**	47.0	**30.8**	49.2
AdaGrad [Green & Wang+ 13]	921K	**32.6**	**46.5**	**31.1**	**48.4**
RPROP	921K	**32.7**	**46.4**	**31.3**	**48.4**
RPROP without leave-one-out	921K	**32.5**	47.0	**30.7**	49.2
RPROP + word pair, triplet and HRM features	5.22M	**33.3**	**45.8**	**31.6**	**47.9**

aged over three independent MERT runs, and statistical significance is evaluated with *MultEval* [Clark & Dyer+ 11].

Additional experiments are run on two large-scale tasks over strong baselines which include n-best reranking with a recurrent neural language modeling component. We use our internal evaluation system as a baseline for the DARPA BOLT Chinese→English task described in Section A.4.1. It is a powerful hierarchical phrase-based statistical machine translation engine with 19 dense features, including a hierarchical reordering model [Huck & Wuebker+ 13] and an LSTM recurrent neural language model [Sundermeyer & Schlüter+ 12] for reranking. The 5-gram backoff language model is trained on 2.9 billion running words in total. To facilitate a direct comparison, we use the same data for tuning and testing as [Setiawan & Zhou 13], namely DEV10-tune (tune) and DEV10-dev[2], which are taken from LDC2010E30 and consist of web data, and the NIST MT06 evaluation set. Further, we test on an additional single-reference test set from the discussion forum domain, which is referred to as P1R6-dev. We perform maximum expected BLEU training on the discussion forum portion of the training data.

On the German→English task of the *9th Workshop on Statistical Machine Translation* both standard translation model training and maximum expected BLEU training are performed on all available bilingual data, more than four million sentence pairs. Our baseline is a phrase-based translation engine with a 4-gram backoff language model trained on 2.5 billion words with `lmplz` [Heafield & Pouzyrevsky+ 13], a recurrent neural language model, a 7-gram word class language model and the hierarchical reordering model.

In the initial experiments we limit the feature set trained for maximum expected BLEU to a discriminative phrase table (cf. Equation 4.34). Later, we also apply lexical features (cf. Equation 4.35), triplets (cf. Equations 4.36 and 4.37) and a discriminative variant of the hierarchical lexicalized reordering model (HRM, cf. Equations 4.38 and 4.39).

[2]named *dev* in [Setiawan & Zhou 13]

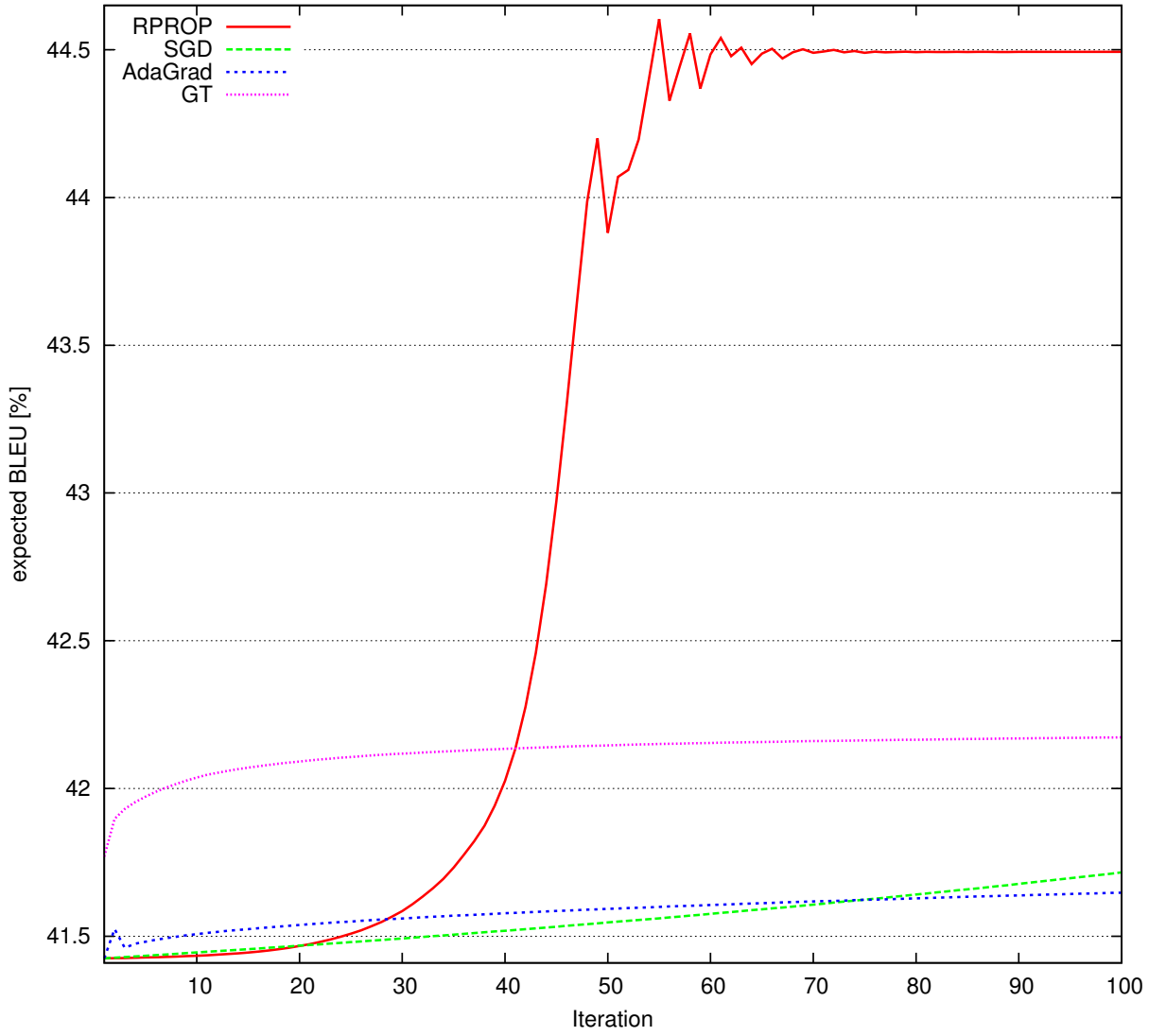


Figure 7.8: Expected BLEU value on IWSLT German→English for the different update strategies. Note that growth transformation (GT) applies a different regularization term and is therefore not directly comparable with the other techniques.

7.4.2 Results

IWSLT

Table 7.10 shows the IWSLT results. First, we compare the performance of the four update algorithms, using only the discriminative phrase table features for simplicity. Unless otherwise stated, the n-best lists of the training data were generated with leave-one-out. In all cases, different values for the regularization parameter τ and in the case of stochastic gradient descent and AdaGrad also for the learning rate η were tested, selecting the best configurations based on the validation set `test2011`. For AdaGrad we also experimented with FOBOS regularization and feature selection [Duchi & Singer 09], but did not observe improved results. Confirming observations in [Gao & He 13], we found that in all cases regularization is not strictly necessary - results are barely affected as long as τ is sufficiently small - and that stochastic gradient descent

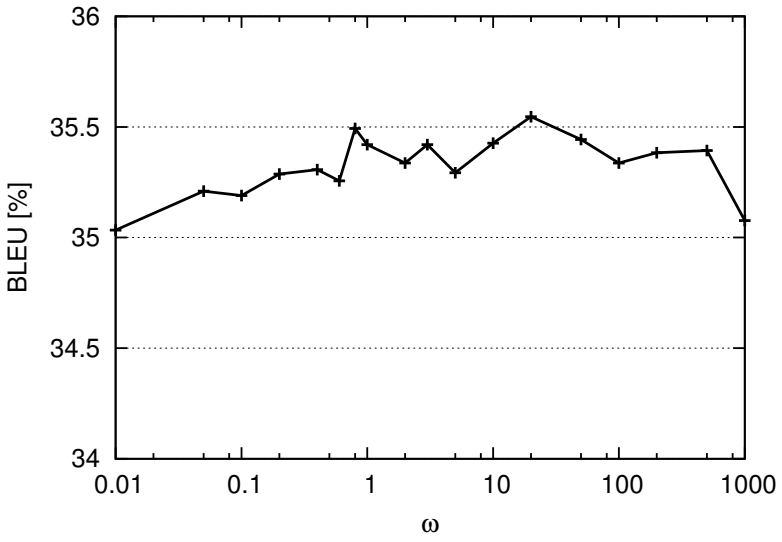


Figure 7.9: BLEU [%] scores on the IWSLT German→English selection set `tst2011` with varying scaling factors ω (cf. Equation 6.18).

is considerably more sensitive to the learning rate η than AdaGrad. Further, stochastic gradient descent and RPROP need around 25 iterations to reach optimal results. For growth transformation and AdaGrad, 5-10 iterations are sufficient. To facilitate a fair comparison, however, we run all algorithms for 40 iterations and select the iteration performing best on a selection set, namely iterations 19 (AdaGrad), 23 (GT), 29 (RPROP) and 35 (SGD). Figure 7.8 displays the expected BLEU function as it evolves in training with different update strategies. Although the value for growth transformation is not directly comparable to the others due to a different regularization term, the respective characteristics are clearly visible. Stochastic gradient descent exhibits a linear growth pattern. Growth transformation resembles a logarithmic and RPROP an exponential function in the first 50 iterations. However, afterwards it starts fluctuating and is saturated around iteration 65. AdaGrad also displays logarithmic characteristics, after initially overshooting and then retracting as the regularization kicks in.

In terms of BLEU, RPROP performs best, followed by AdaGrad, growth transformation and stochastic gradient descent. The RPROP-AdaGrad and AdaGrad-growth transformation differences are small (0.2% BLEU absolute) but statistically significant at the 95% level. Altogether, RPROP improves over the baseline by 0.9 BLEU points and 0.8 points TER, which is statistically significant at the 99% level. In an additional experiment we verified that leave-one-out has a clear impact on the results. The difference between RPROP with and without leave-one-out is 0.6% BLEU and 0.8 % TER absolute. By adding lexical, triplet and reordering features, we get an additional gain and observe a total absolute improvement of 1.2 % BLEU and 1.3% TER over the baseline system.

We also experimented with varying scaling factors ω for the maximum entropy computation of posterior probabilities (Equation 6.18). We plot the BLEU [%] scores with different ω on our selection set `tst2011` in Figure 7.9. Altogether the differences in BLEU are small. Between $\omega = 0.2$ and $\omega = 500$, BLEU values vary between 35.29% and 35.54%, which we can interpret as noise. Only the extreme cases $\omega = 0.01$ and $\omega = 1000$ result in noticeably lower quality. Our conclusion is that no optimization of the scaling factor is necessary and we can safely set $\omega = 1$, which means that it can be dropped from the gradient computations.

We will now shortly discuss the efficiency of growth transformation compared to the other

Table 7.11: Comparison of different feature sets for maximum expected BLEU training on the IWSLT 2013 German→English task. We separately test all feature sets described in Section 4.2.10. For the word class based feature types we trained 100 classes for both source and target vocabulary. Statistically significant improvements over the baseline on the 99% level are printed in boldface.

model type	**word classes**	**# features**	dev		test	
			BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
baseline		18	32.1	47.2	30.4	49.2
discriminative phrase table	no	921K	**32.7**	**46.4**	**31.3**	**48.4**
(Equations 4.34, 4.42)	yes	474K	**32.7**	**46.6**	**31.0**	**48.6**
word pairs	no	722K	**32.6**	**46.6**	**30.8**	**48.7**
(Equations 4.35, 4.43)	yes	9.7K	**32.7**	**46.6**	**31.1**	**48.5**
source triplets	no	1.6M	32.3	**46.9**	**30.9**	**48.8**
(Equations 4.36, 4.44)	yes	232K	**32.4**	**46.6**	**30.9**	**48.7**
target triplets	no	1.7M	**32.5**	**46.6**	**31.1**	**48.5**
(Equations 4.37, 4.45)	yes	235K	**32.6**	**46.4**	**31.1**	**48.3**
discrimnative HRM	no	2.2M	**32.7**	**46.6**	**30.9**	**48.7**
(Equations 4.38, 4.39, 4.46, 4.47)	yes	1.1M	**32.6**	**46.2**	**30.9**	**48.4**
boundary word HRM	no	786K	**32.6**	**46.7**	**30.8**	**48.7**
(Equations 4.40, 4.41, 4.48, 4.49)	yes	2.4K	**32.6**	**46.8**	**30.9**	**48.8**

updating algorithms. In our training data we have 921K active discriminative phrase table features. Using growth transformation on the same data, this results in a total of 6.08M features being updated due to the re-normalization component. Consequently, growth transformation is less time and space efficient than the other algorithms. In our implementation, growth transformation required around 16 hours and 6.7G memory for 40 iterations. RPROP, AdaGrad and stochastic gradient descent, on the other hand, finished in less than 2.5 hours and required 2.1G memory.

In this setup, we separately test all model types described in Section 4.2.10. Each of them can be computed either on the word level or by using word classes. Here, we trained 100 classes on both source and target vocabulary. The results are shown in Table 7.11. Although the word-level discriminative phrase table performs best, all of the model types show significant improvements over the baseline. The performance difference between the word and word class models are small in most cases. The number of parameters that are trained with RPROP, however, can for some models be drastically reduced by parameterizing using word classes. An investigation of how these models are best combined is left to future work.

Further experiments were performed on the large IWSLT setup, where the baseline system is trained on all available bilingual data. These results are presented in Table 7.12. Our observation is that our discriminative training scheme has a stronger impact on quality in this scenario. Including phrasal, lexical and triplet features, we can see as much as 1.7 BLEU points improvement over the baseline on `test`. As maximum expected BLEU training is performed on the in-domain portion of the data only, we can assume that part of that difference is due to domain adaptation effects. Here, we also had a closer look at how translation quality evolves through the training iterations. The results of this experiment are plotted in Figure 7.10. We can see that the setups with and without leave-one-out perform similarly in the first ∼20 iterations. At that point, however, without leave-one-out the performance already seems to converge, while the translation

Table 7.12: Results for the large setup of the IWSLT 2013 German→English task, where maximum expected BLEU training was performed on the TED data only. We compare using phrase pair features both with and without leave-one out, as well as a richer feature set including word pair (Equation 4.35) and triplet (Equations 4.36 and 4.37) features with leave-one-out. Statistically significant improvements over the baseline on the 99% level are printed in boldface.

	dev		test	
	BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
baseline	32.9	47.0	30.3	49.8
RPROP without leave-one-out	**33.4**	46.4	**30.9**	49.0
RPROP	**34.0**	**45.9**	**31.5**	**48.7**
RPROP + word pair and triplet features	**34.6**	**45.1**	**32.0**	**48.1**

Table 7.13: Results for the BOLT Chinese→English task in BLEU [%] on the discussion forum test set (`P1R6-dev`), the mixed web test set (`DEV10-dev`) and NIST MT06. The baseline is our BOLT evaluation system and contains a recurrent neural language model for re-ranking. We compare with [Setiawan & Zhou 13] who applied maximum expected BLEU training with growth transformation (GT). Note that the number of features reported by [Setiawan & Zhou 13] is artificially blown up due to re-normalization.

	# features	P1R6-dev		DEV10-dev		NIST MT06	
		BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
baseline	19	18.0	65.6	34.1	56.8	39.7	54.1
SGD	12.4M	18.0	65.8	34.3	56.5	39.8	53.8
AdaGrad	12.4M	18.3	65.4	34.7	**56.4**	40.1	53.7
RPROP	12.4M	**18.7**	**64.8**	**34.8**	**56.4**	**40.5**	**53.6**
[Setiawan & Zhou 13] (GT)	150M	-	-	32.7	-	40.3	-

quality with leave-one-out continues improving and reaches a higher maximum around iteration 30.

BOLT

On the BOLT task, we directly compare with the system presented by [Setiawan & Zhou 13], which was trained with growth transformation. We use the same tune set for minimum error rate training and report results on the same test sets, which are given in Table 7.13. Using RPROP we observe nearly twice the improvement reported in [Setiawan & Zhou 13] on both `DEV10-dev` and `NIST MT06` using phrasal, lexical and triplet features[3]. The baseline on `DEV10-dev` is already much stronger than the system in [Setiawan & Zhou 13] and RPROP training yields an additional 0.7 BLEU points, as opposed to 0.44 reported by Setiawan. On `MT06` our baseline system is slightly worse, but with the larger gain obtained by maximum expected BLEU training with the RPROP algorithm our final system outperforms the one reported in [Setiawan & Zhou 13] by 0.2% BLEU absolute. We would like to stress that this is not a domain adaptation effect.

[3]Reordering features are not applicable to our hierarchical system.

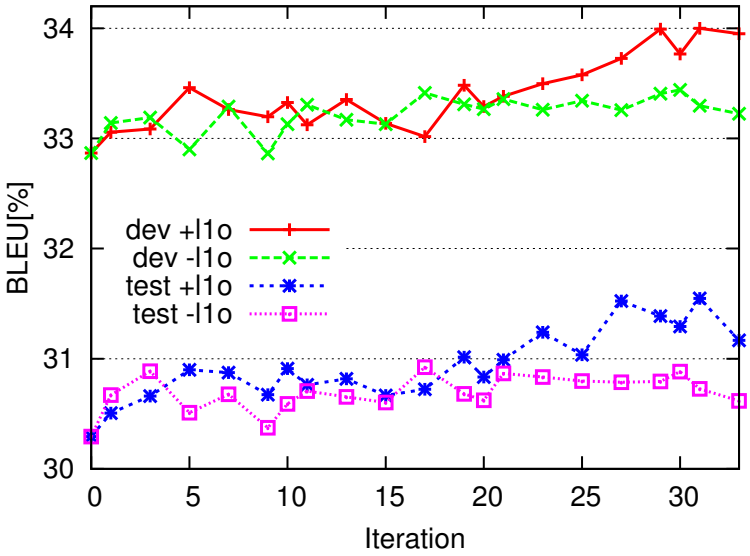


Figure 7.10: BLEU scores on the large IWSLT German→English task for RPROP training evaluated over the training iterations. Results are reported on both `dev` and `test`, with (*+l1o*) and without (*-l1o*) leave-one-out.

Maximum expected BLEU training was performed on discussion forum data, which is different from the web domain in `DEV10-dev` and `MT06`. On the `P1R6-dev` discussion forum test set, on the other hand, RPROP training can be seen as a special type of domain adaptation. The improvement here is 0.7% BLEU absolute with a single reference (`DEV10-dev` and `MT06` both have four references). By training the same feature sets with stochastic gradient descent and AdaGrad we can confirm results observed on the IWSLT task. Here, stochastic gradient descent yields only minor improvements. AdaGrad performs better, but still 0.1 - 0.4 BLEU points worse than RPROP. Running growth transformation has proven infeasible in our hierarchical phrase-based setup, which generally requires a larger number of phrase pairs than a standard phrase-based system.

WMT

Table 7.14 lists our results on the German→English WMT task. This is our largest setting: Maximum expected BLEU training is performed on the full training data with more than 4M sentence pairs. To the best of our knowledge, this number is unsurpassed in the literature on discriminative training in machine translation. Altogether, training took more than one month. About 75% of the time were needed for decoding the training data to generate n-best lists. Here, we applied phrasal, lexical an reordering features, 45M in total. With the re-normalization step required by GT, this number would grow to 309M. On `newstest2013`, our baseline outperforms the best single system reported on `matrix.statmt.org` by 0.2 BLEU points, but is clearly below the performance of Edinburgh's 2014 submission on `newstest2014`, which is a syntax-based system. The discriminatively trained features yield an absolute improvement of 0.6% BLEU and 0.3% TER on `newstest2013` and of 0.5% BLEU and 0.4% TER on `newstest2014`.

Table 7.14: Results for the WMT 2014 German→English task. The baseline contains a recurrent neural LM which is applied in a re-ranking step. We compare with the best single system that is reported on `matrix.statmt.org`, which was submitted by the University of Edinburgh both in 2013 and 2014. The 2013 submission is a standard phrase-based decoder with 14 features, while the 2014 submission was a syntax-based system with 9 features.

	# features	newstest2012		newstest2013		newstest2014	
		BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
baseline	19	25.9	56.5	28.3	53.3	28.1	52.7
RPROP	45.0M	26.4	56.2	28.9	53.0	28.6	52.3
`matrix.statmt.org`	14/9	-	-	28.1	-	29.5	-

Table 7.15: We compare translation quality measured in BLEU and TER and decoding speed in words per second using different language model look-ahead techniques on the WMT 2015 German→English task. The translation engine is our baseline system including a 5-gram standard language model and a 7-gram word class language model using the *trie* representation of the KenLM toolkit.

language model look-ahead	words/sec.	newstest2013		newstest2014		newstest2015	
		BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
none	0.32	28.2	54.0	28.4	52.9	29.1	51.4
first-word	0.66	28.2	54.0	28.4	52.9	29.1	51.4
phrase-only	2.05	27.8	54.5	28.0	53.6	28.9	52.1
hybrid	1.05	28.2	53.9	28.2	52.9	29.2	51.5

7.5 Comprehensive Large-Scale Comparison

In order to put the techniques presented in this work into perspective, we will now compare several of them on a single large-scale task using a state-of-the-art baseline system. We also include internal results with other popular methods as well as external results from other teams.

7.5.1 Experimental Setup

We chose the German→English task of the *EMNLP 2015 Tenth Workshop on Statistical Machine Translation* for our purpose. A data description can be found in Section A.1.4. Our phrase-based translation system uses a 5-gram language model trained on 4.4 billion running words. The baseline is augmented with the hierarchical reordering model. We concatenate the `newstest2011` and `newstest2012` corpora for tuning and results are reported on the `newstest2013`, `newstest2014` and `newstest2015` sets. We first add a 7-gram word class language model to the baseline. On top of this, we compare the word class models (Section 4.2.9), Viterbi forced alignment training (Section 6.2.3) with 2000-best lists and maximum expected BLEU training (Section 6.3). We apply the word class variants of both the translation model and the hierarchical reordering model, trained with 200 classes. Forced alignment training is performed for one iteration with leave-one-out and the count model based on 2000-best lists. Maximum expected BLEU training applies all feature values presented in Section 4.2.10 and uses the RPROP algorithm for optimization. Here,

Table 7.16: Results on the WMT 2015 German→English task. We compare several of the novel models presented in this work: a 7-gram word class language model (wcLM), word class translation and reordering models (wcTM, wcHRM), forced alignment phrase training and maximum expected BLEU training. Additionally we present results with the joint translation and reordering model[4] (JTR) [Guta & Alkhouli+ 15], estimated either as a count model with Kneser-Ney smoothing (KN) or with feed-forward neural networks (FFNN). Another comparison is against a system containing an LSTM recurrent language model and several feed-forward neural network models[5] as described in [Peter & Toutounchi+ 15b]. RWTH Aachen University's final submission to the shared task is printed in boldface and it included maximum expected BLEU training as the only additional component. Finally, the result of the winning system by the Karlsruhe Institute of Technology (KIT) is given for reference.

	newstest2013		newstest2014		newstest2015	
	BLEU [%]	**TER** [%]	**BLEU** [%]	**TER** [%]	**BLEU** [%]	**TER** [%]
baseline	27.8	54.2	27.8	53.2	28.5	51.8
+ 7-gram wcLM[4]	28.2	54.0	28.4	52.9	29.1	51.4
+ wcTM, wcHRM	28.3	54.0	28.4	52.9	29.2	51.4
+ forced alignment training	28.3	54.1	28.2	53.3	29.1	51.9
+ maximum expected BLEU training	28.8	53.4	28.8	52.6	29.7	51.1
+ JTR (KN)[4]	28.9	53.3	28.7	52.6	29.8	50.8
+ JTR (FFNN)[5]	28.8	53.4	28.6	52.5	29.4	51.1
+ neural network models[5]	29.1	53.0	28.3	52.6	29.5	50.9
+ maximum expected BLEU training[5]	**29.4**	**52.7**	**28.9**	**52.0**	**30.3**	**50.3**
WMT 2015 best submission (KIT)	-	-	-	-	30.4	-

we use a subset of the training data for training, namely the `newstest2008`, `newstest2009` and `newstest2010` sets plus 200k sentences selected from the common crawl corpus using the bilingual cross-entropy criterion described in [Axelrod & He+ 11]. We apply `newstest2013` as selection set for the training iteration.

7.5.2 Results

We first re-evaluate the effect of language model look-ahead in this large-scale state-of-the-art setup. The baseline includes one standard 5-gram and one class-based 7-gram language model. Both make use of the *trie* representation of the KenLM toolkit. We compare the three look-ahead variants with the baseline system that does not use look-ahead in Table 7.15. We re-optimized the log-linear model weights for hybrid and phrase-only look-ahead. Similar to our previous results, we observe that the first-word look-ahead doubles translation speed without changing the translation output. Hybrid look-ahead increases speed by an additional factor of 1.6 with very little impact on translation quality. Phrase-only look-ahead leads to a clear drop in translation quality, but is again nearly twice as fast as the hybrid variant. The following experiments are run with first-word look-ahead.

The main results on this task are listed in Table 7.16. We report the best out of three minimum error rate training runs, selected on `newstest2013`. We compare with internal results using the *joint translation and reordering sequence* (JTR) model [Guta & Alkhouli+ 15], using two different methods for its computation: as a count model with Kneser-Ney smoothing (KN)

and with feed-forward neural networks (FFNN). Further, we list a system including four neural networks: one LSTM recurrent neural language model [Sundermeyer & Ney+ 15], a feed-forward 7-gram language model, a feed-forward translation model with a source window of five words and a feed-forward joint model with the same source window plus a target history of four words [Devlin & Zbib+ 14]. This system was further augmented with maximum expected BLEU training for RWTH Aachen University's submission [Peter & Toutounchi+ 15b]. Finally, we include the winning system of the official evaluation, submitted by the Karlsruhe Institute of Technology (KIT) as reported on the `matrix.statmt.org` website.

We can see that the 7-gram word class language model (wcLM) improves quality by up to 0.6% BLEU and 0.7% TER absolute. The word class translation and reordering models (wcTM, wcHRM), on the other hand, result in only minor improvements. Forced alignment phrase training even leads to a small degradation in translation performance, but also reduced the size of the phrase table by nearly two thirds from 327 to 117 million entries. Applying maximum expected BLEU training yields improvements of up to 0.6% BLEU and 0.3% TER absolute. The performance of this system is on par with both the JTR and the neural networks. On top of the neural network models, we can even observe further absolute improvements of up to 0.8% BLEU and 0.6% TER using maximum expected BLEU training. This was the final system submitted to the WMT evaluation by RWTH Aachen University. The winning system (KIT) was better by an additional 0.1% BLEU.

Finally, we take a look at several translation examples in Figure 7.11. One example sentence was selected for each of the four techniques (a) word class language model, (b) word class translation and reordering model, (c) forced alignment training and (d) maximum expected BLEU training, to illustrate their effect on the translation result. In example (a), the 7-gram word class language model has a stronger preference for the correct translation than the standard language model, resulting in an exact match of the reference translation. In example (b), the word order is improved by application of word class models. Using forced alignment training improves the translation in example (c) by fixing a word deletion error and a better word choice. Example (d) illustrates a sentence, where maximum expected BLEU training results in a considerably improved word order and lexical selection.

7.6 BOLT Evaluation

RWTH Aachen University was part of the DELPHI IBM team in the Broad Operational Language Technology (BOLT) Program funded by DARPA. The author of this work was responsible for training and maintaining the Chinese→English translation engines. As part of the final official evaluation, IBM conducted an internal comparison between the team members. We will summarize the results in this section.

7.6.1 Evaluation Systems

All evaluation systems are based on the hierarchical decoder that is part of RWTH Aachen University's open source toolkit *Jane* [Vilar & Stein+ 10]. The training data is described in Section A.4.1 and the systems are heavily tuned towards the specific task. There are four data conditions, which focus on different types of user generated content: *discussion forum*, *SMS/Chat*, *telephony - human transcripts* and *telephony - automatic transcripts*. For each of the conditions we were provided with a small amount of additional in-domain training and tuning data.

[4]Provided by Andreas Guta.
[5]Provided by Jan-Thorsten Peter.

source	Den Ermittlern erzählte der Rentner, seine Mutter sei nach Spanien gereist.
baseline	Investigators told the retirees, his mother has traveled to spain.
+ 7-gram wcLM	The pensioner told investigators that his mother had travelled to spain.
reference	The pensioner told investigators that his mother had travelled to spain.

(a) word class language model

source	Passagiere schimpfen häufig über Gepäck Zuschläge und andere Gebühren, doch Fluggesellschaften greifen gern darauf zurück.
baseline + 7-gram wcLM	Excess baggage charges and other charges, but passengers often gripe about airlines are happy.
+ wcTM, wcHRM	Passengers often grumble about luggage surcharges and other fees, but airlines are happy.
reference	Passengers often grumble about baggage charges and other fees, but airlines love them.

(b) word class translation and reordering model

source	In diesem Jahr sind aufreizende unbelebte Gegenstände der letzte Schrei.
baseline + 7-gram wcLM	This year are provocative forces of rage.
+ forced alignment training	This year, provocative inanimate objects are all the rage.
reference	This year, sexy inanimate objects are all the rage.

(c) forced alignment training

source	Die Zerstörung der Einrichtungen bedeutet, dass Syrien keine neuen Chemiewaffen mehr produzieren kann.
baseline + 7-gram wcLM	The destruction of Syria is not a new chemical weapons facilities that can produce more.
+ maximum expected BLEU training	The destruction of the facilities means that Syria can produce no new chemical weapons.
reference	Destruction of the equipment means that Syria can no longer produce new chemical weapons.

(d) maximum expected BLEU training

Figure 7.11: Translation examples from the WMT 2015 German→English task. We show examples that illustrate the effect of (a) the word class language model, (b) the word class translation and reordering models, (c) the forced alignment training and (d) maximum expected BLEU training.

Table 7.17: Results for the final BOLT evaluation within the IBM team on the *discussion forum* condition. The evaluation measure is $\frac{\text{TER}-\text{BLEU}}{2}$, denoted as (T-B), where smaller numbers are better. All numbers are computed by IBM.

Group	**average (T-B)** [%]	`DEV12-test` **(T-B)** [%]	`P1R6-test` **(T-B)** [%]	`P1-Progress-test` **(T-B)** [%]
IBM Forest Syntax	22.1	23.6	21.2	21.5
RWTH Aachen University	**23.1**	**23.8**	**22.4**	**23.0**
IBM TreeToString	23.3	24.5	22.3	23.0
Stanford University	23.6	24.9	22.5	23.4
University of Le Mans	23.8	24.9	23.2	23.5
IBM Direct Translation Model	24.5	25.4	23.4	24.6
University of Maryland	24.6	25.7	23.4	24.6

All systems contain a 5-gram backoff language model and a 7-gram word class language model, which are mixture models from a large number of data sources, tuned for perplexity on the condition-specific development sets. In total, both language models are trained on ∼2.9 billion running words. The translation systems also contain a hierarchical reordering model [Huck & Wuebker$^+$ 13] and are adapted towards the domain by leveraging the condition-specific training data with maximum expected BLEU training (cf. Section 6.3) and LSTM recurrent neural language and translation models [Sundermeyer & Alkhouli$^+$ 14, Sundermeyer & Schlüter$^+$ 12].[6]

The translation quality is measured in $\frac{\text{TER}-\text{BLEU}}{2}$, denoted as (T-B), where smaller numbers correspond to higher quality. All results were computed centrally at IBM.

7.6.2 Results: Discussion Forum

The *discussion forum* condition focuses on translation of data crawled from web forums. Table 7.17 presents the results computed by IBM on three blind test sets: `DEV12-test`, `P1R6-test` and `P1-Progess-test`. On average, the RWTH Aachen University system performs second best, 1.0% (T-B) behind the IBM Forest syntax system, and 0.2% (T-B) better than the IBM TreeToString engine. The total ranking is quite consistent throughout the different test sets.

7.6.3 Results: SMS/Chat

The data used in the *SMS/Chat* condition consists of informal short messages, taken both from web chats and mobile phone services. The results are given in Table 7.18. Here we consider one blind test set (`P2R1-test`) and one development set (`P2R2-syscomtune`). For this genre, RWTH Aachen University generated the highest quality translations on average, outperforming the University of Le Mans by 0.3% (T-B).

7.6.4 Results: Telephony - Human Transcripts

Telephony - human transcripts refers to language used in spontaneous speech. The data is produced by human transcription of telephone conversations. The results on two blind test sets (`P3R1-test` and `P3R1-validation`) can be seen in Table 7.19. RWTH Aachen University is in fourth position, 1.1% (T-B) behind the best scoring IBM Forest Syntax system.

[6] The recurrent translation models as described in [Sundermeyer & Alkhouli$^+$ 14] were co-developed by the author, but are not part of this thesis.

Table 7.18: Results for the final BOLT evaluation within the IBM team on the *SMS/Chat* condition. The evaluation measure is $\frac{\text{TER}-\text{BLEU}}{2}$, denoted as (T-B), where smaller numbers are better. All numbers are computed by IBM.

Group	**average (T-B)** [%]	`P2R1-test` **(T-B)** [%]	`P2R2-syscomtune` **(T-B)** [%]
RWTH Aachen University	**23.2**	**22.5**	**23.8**
University of Le Mans	23.5	23.2	23.8
IBM TreeToString	23.6	23.4	23.8
IBM Forest Syntax	23.6	23.8	23.5
Stanford University	23.7	23.5	23.8
University of Maryland	24.2	23.9	24.5
IBM Direct Translation Model	24.8	25.0	24.6

Table 7.19: Results for the final BOLT evaluation within the IBM team on the *telephony - human transcripts* condition. The evaluation measure is $\frac{\text{TER}-\text{BLEU}}{2}$, denoted as (T-B), where smaller numbers are better. All numbers are computed by IBM.

Group	**average (T-B)** [%]	`P3R1-test` **(T-B)** [%]	`P3R1-validation` **(T-B)** [%]
IBM Forest Syntax	16.2	15.3	17.1
IBM TreeToString	16.3	15.7	17.0
University of Maryland	17.1	16.4	17.8
RWTH Aachen University	**17.3**	**16.6**	**18.0**
Stanford University	17.4	16.7	18.2
University of Le Mans	17.8	17.0	18.5
IBM Direct Translation Model	18.0	17.1	18.8

7.6.5 Results: Telephony - Automatic Transcripts

The data condition using the same data source as in the previous section, but with error prone transcripts produced by automatic speech recognition, is referred to as *telephony - automatic transcripts*. We only report results on a single blind test set, `P3R1-test`, in Table 7.20. Replacing the correct transcription with an automatically generated one results in a drop in translation quality by ∼13% (T-B). Here, the RWTH Aachen University system was the best within the IBM team, with a small margin of 0.1% (T-B) over IBM Forest Syntax.

7.7 IWSLT 2014 Evaluation

In this Section we discuss the results of the 2014 *International Workshop on Spoken Language Translation*[7] (IWSLT) German→English and English→French shared tasks. A summary of the entire evaluation campaign can be found in [Cettolo & Niehues$^+$ 14]. The goal of this task is the translation of video lectures from TED conferences. RWTH Aachen University participated both in the machine translation (MT) and the spoken language translation (SLT) tracks for the

[7] `http://workshop2014.iwslt.org/`

Table 7.20: Results for the final BOLT evaluation within the IBM team on the *telephony - automatic transcripts* condition. The evaluation measure is $\frac{\text{TER}-\text{BLEU}}{2}$, denoted as (T-B), where smaller numbers are better. All numbers are computed by IBM.

Group	`P3R1-test` **(T-B)** [%]
RWTH Aachen University	**27.9**
IBM Forest Syntax	28.0
University of Le Mans	28.3
IBM TreeToString	29.5
IBM Direct Translation Model	29.9
University of Maryland	30.9
Stanford University	30.9

aforementioned language pairs. Here, we will focus on the MT results.

7.7.1 Evaluation Systems

RWTH Aachen University's evaluation system is described in detail in [Wuebker & Peitz$^+$ 14]. The system architectures are identical for both language pairs. We applied the standard phrase-based decoder from *Jane* (cf. Section 5), which was developed by the author of this work as a by-product of this thesis. As additional components it contains the hierarchical reordering model[8] (cf. Section 4.2.8) and a 7-gram word class language model (cf. Section 4.2.9). Further, we performed maximum expected BLEU training (cf. Section 6.3) of phrasal, lexical and reordering features and applied rescoring using recurrent neural network language and translation models.[9] Altogether, all components of the evaluation systems were either developed or co-developed by the author of this work, or under the author's supervision. The enhancements presented in this thesis are crucial for the overall performance of the evaluation systems. For an evaluation of each component's contribution please refer to [Wuebker & Peitz$^+$ 14].

RWTH Aachen University also participated in a joint submission of the partners of the EU-Bridge project.[10] It is a system combination of the translation engines from the project partners RWTH Aachen University, University of Edinburgh, Karlsruhe Institute of Technology, and Fondazione Bruno Kessler. Details about this submission can be found in [Freitag & Wuebker$^+$ 14]. We will report its results but do not consider it a competing single engine.

7.7.2 Results

The official results for the German→English MT task are presented in Table 7.21. Our system performs best in both BLEU and TER.

On the English→French language pair, the organizers performed a human evaluation. This is measured in the HTER [%] metric, which is similar to TER but replaces the reference translations with post-edited versions of the machine translation output. These results are shown in Table 7.22. RWTH Aachen University is the best performing single system according to both HTER [%] and TER, but in third position according to BLEU.

[8] Implemented during the course of a Bachelor thesis under the author's supervision.
[9] Co-developed by the author.
[10] `http://www.eu-bridge.eu`

Table 7.21: Official results of the best five single engine submissions for the IWSLT 2014 German→English machine translation task. All components of the RWTH Aachen University submission were developed or co-developed by the author, or under the author's supervision.

Group	**BLEU** [%]	**TER** [%]
EU-Bridge (system combination)	*25.8*	*54.6*
RWTH Aachen University	**25.0**	**55.5**
Karlsruhe Institute of Technology	24.6	55.6
NTT Communication Science Labs, Japan	23.8	56.4
University of Edinburgh	23.3	57.5
Fondazione Bruno Kessler	20.5	63.4

Table 7.22: Official results of the best five single engine submissions for the IWSLT 2014 English→French machine translation task. All components of the RWTH Aachen University submission were developed or co-developed by the author, or under the author's supervision.

Group	**HTER** [%]	**BLEU** [%]	**TER** [%]
EU-Bridge (system combination)	*19.2*	*37.0*	*45.2*
RWTH Aachen University	**19.3**	35.7	**44.5**
Karlsruhe Institute of Technology	20.9	**36.2**	45.2
University of Edinburgh	21.5	35.9	45.8
Massachusetts Institute of Technology	22.6	35.5	45.7
Fondazione Bruno Kessler	22.9	34.2	46.8

7.8 Overview of Evaluation Campaigns

The goal of this section is a schematic overview of the most important evaluation campaigns to which the author of this work contributed or where the RWTH Aachen University submission system contains components developed by the author. The overview is given in Table 7.23.b The WMT and IWSLT systems are described in the corresponding system papers [Huck & Wuebker$^+$ 11, Peitz & Mansour$^+$ 13, Peitz & Wuebker$^+$ 14, Peter & Toutounchi$^+$ 15b, Peitz & Mansour$^+$ 12, Wuebker & Peitz$^+$ 13a, Wuebker & Peitz$^+$ 14, Peter & Toutounchi$^+$ 15a]. We only report results for single systems. In addition to RWTH Aachen University's position in the official results and the total number of participants, we list the following key components that were used in the submission system and to which the author of this thesis contributed.

Phrase-based decoder. The author implemented the phrase-based decoder into RWTH Aachen University's open source machine translation toolkit *Jane* [Wuebker & Huck$^+$ 12].

Language model look-ahead. This technique was developed and implemented by the author [Wuebker & Ney$^+$ 12b].

Recurrent neural network models. The models were co-developed with Martin Sundermeyer and Tamer Alkhouli and published in [Sundermeyer & Alkhouli$^+$ 14].

Forced alignment training. The procedure was developed by the author of this work during his Diploma thesis [Wuebker 09] and published in [Wuebker & Mauser+ 10].

Language model data selection. The author implemented the technique presented in [Moore & Lewis 10] into RWTH Aachen University's software toolkit.

Hierarchical reordering model. The implementation of the model [Galley & Manning 08] was performed under the author's supervision during the course of Felix Rietig's Bachelor thesis [Rietig 13].

Word class language model. This is one of the word class based smoothing models proposed by the author [Wuebker & Peitz+ 13b].

Maximum expected Bleu training. The discriminative training procedure described in [Wuebker & Muehr+ 15] was developed by the author together with Sebastian Mühr and Patrick Lehnen.

Components that are part of the contribution of this thesis are printed in boldface in Table 7.23.

7.9 Conclusion

In this chapter, we have performed an experimental analysis of the novel methods introduced in Chapters 4, 5 and 6. Section 7.1 is concerned with pre-sorting the phrase translation candidates with a language model score estimate and the look-ahead techniques presented in Section 5.3. When applied in conjunction with observation histogram pruning, our pre-sorting can improve Bleu by 0.2% absolute while increasing translation speed by a factor of two. Language model look-ahead increases speed at least by another factor of two with first-word look-ahead and a factor of more than four with phrase-only and hybrid look-ahead. Hybrid look-ahead has the best quality-speed tradeoff. Compared to the popular open source decoder *Moses*, our system reaches the same top quality but is between 2.5 and 22 times faster, depending on the required translation quality.

In Section 7.2 we evaluate the effect of the class-based smoothing models described in Section 4.2.9. We can show that a linear interpolation of standard language models and word class language models can reduce perplexity. On a French→German translation task we report improvements of up to 1.4% Bleu and 1.0% Ter using translation, reordering and language models based on word classes.

Our experiments in Section 7.3 provide evidence that forced alignment training can improve translation quality by up to 0.9% Bleu and 0.4% Ter absolute on the German→English WMT 2008 task while reducing the phrase table size by more than 80%. Linear interpolation with the baseline phrase table yields further gains. Applying length-incremental training avoids the need for a Viterbi word alignment and reaches a performance similar to the baseline. Our analysis shows that the resulting phrase table only has a small overlap with the heuristically extracted one and contains shorter phrases.

Section 7.4 experimentally compares different update strategies for maximum expected Bleu training. RPROP performs best and we also show the importance of the leave-one-out heuristic. Our system outperforms previously published results on two large-scale tasks and on the German→English WMT 2014 task we perform discriminative training on the largest training data set reported in the literature to date.

We re-visit several of the previously described methods for a comprehensive comparison on the German→English WMT 2015 shared task in Section 7.5. While the word class translation and reordering models as well as forced alignment training have little impact, we show that the word

Table 7.23: Schematic overview of the results of evaluation campaigns to which the author of this work contributed and where the RWTH Aachen University submission system contains components developed by the author. The components marked in **bold** are contributions of this thesis. The author's contributions to the remaining components referred to in this table are described in Section 7.8. The results of the IWSLT 2015 evaluation were not yet available. All results refer to single systems only.

campaign	year	task	position	# participants	components							
					phrase-based decoder *Jane* (cf. Chapters 4 and 5)	**language model look-ahead** (cf. Section 5.3)	language model data selection [Moore & Lewis 10]	hierarchical reordering model (cf. Section 4.2.8)	**word class language model** (cf. Section 4.2.9)	**maximum expected BLEU training** (cf. Section 6.3)	recurrent neural network models [Sundermeyer & Alkhouli+ 14]	forced alignment training (cf. Section 6.2.2)
WMT	2011	De→En	3	12			✓					✓
	2013	En→Fr	2	12	✓	✓		✓	✓			
		De→En	3	13	✓	✓	✓					
	2014	De→En	4	9	✓	✓	✓	✓	✓	✓		
	2015	De→En	3	10	✓	✓	✓	✓	✓	✓	✓	
IWSLT	2012	De→En	1	4	✓	✓	✓		✓			✓
	2013	En→De	2	5	✓	✓	✓	✓	✓	✓		
	2014	De→En	1	6	✓	✓	✓	✓	✓	✓	✓	
		En→Fr	1	7	✓	✓	✓	✓	✓	✓	✓	
	2015	De→En	1	3	✓	✓	✓	✓	✓	✓	✓	
BOLT Zh→En	2014	discussion forum	2	7			✓	✓	✓	✓	✓	
		SMS/Chat	1									
		telephony automatic transcripts	4									
		telephony human transcripts	1									

class language model and maximum expected BLEU training are crucial components of RWTH Aachen University's current state-of-the-art statistical machine translation engine. We further show that maximum expected BLEU training performs on par with the joint translation and reordering (JTR) models as well as with a set of four neural network language, translation and joint models.

The word class language model and maximum expected BLEU training are also important parts of RWTH Aachen University's BOLT system. The results of the IBM-internal evaluation are presented in Section 7.6. Out of four different conditions, RWTH Aachen University reached two first positions and one second position.

Section 7.7 reports results of the IWSLT 2014 evaluation campaign. Here, the RWTH Aachen University system scored first position on both the German→English (according to BLEU and TER) and the English→French (according to HTER and TER) tasks. Again, word class language models and maximum expected BLEU training are key to its performance.

Finally, a schematic overview of several public and project-internal evaluation campaigns can be found in Section 7.8. It summarizes the individual components of the RWTH Aachen University evaluation systems as well as the final positions and number of participants. It illustrates the impact and continued usage of the author's contributions on RWTH Aachen University's state-of-the-art translation system, which achieved high ranking positions in a large number of evaluation campaigns.

7.10 Contributions

The experiments on language model look-ahead presented in Section 7.1 were performed exclusively by the author of this work. With the exception of the results on hybrid language model look-ahead, they were already published in [Wuebker & Ney$^+$ 12b].

All phrase-based results with word class smoothing models in Section 7.2 were also produced by the author. The experiments on the hierarchical phrase-based decoder (cf. Table 7.5) were performed by Stephan Peitz. With the exception of the perplexities in Table 7.3, the experiments were published in [Wuebker & Peitz$^+$ 13b].

The content of Section 7.3.1 is joint work with Arne Mauser [Wuebker & Mauser$^+$ 10], where all experiments were performed by the author of this work. The experiments and analyses on length-incremental training (Section 7.3.2) were again performed solely by the author of this work. In addition to the work in [Wuebker & Ney 13], we presented novel results on source- and target-length-incremental training.

The author is also exclusively responsible for the results with discriminative training in Section 7.4, which can partially also be found in [Wuebker & Muehr$^+$ 15]. Here, we additionally include the effect of the scaling factor ω (cf. Figure 7.9) and the results on the large IWSLT 2013 setup (cf. Table 7.12, Figure 7.10).

The baseline system for the WMT 2015 large-scale comparison in Section 7.5 was prepared by Andreas Guta. The experimental results on word class models, forced alignment training and maximum expected BLEU training were produced by the author of this work. Additional comparative experiments are again courtesy of Andreas Guta and Jan-Thorsten Peter, as marked in Table 7.16.

RWTH Aachen University's translation systems for the BOLT evaluation (cf. Section 7.6) were trained and maintained by the author of this work. The scores were computed by IBM.

The German→English translation system for the IWSLT 2014 evaluation campaign was trained by Stephan Peitz. The English→French baseline system was provided by Andreas Guta and then augmented with maximum expected BLEU training and recurrent neural network models by the author of this work [Wuebker & Peitz$^+$ 14]. The scoring was performed by the workshop

organizers [Cettolo & Niehues$^+$ 14].

The evaluation systems described in Section 7.8 are joint work with many colleagues and the section is intended to illustrate the author's impact rather than as a scientific contribution.

8. Conclusion and Scientific Achievements

Chapter 2 defined the scientific goals of this thesis. We will now discuss how each of them was accomplished in the previous chapters.

- In Section 5.3 we presented three variants of language model look-ahead as novel pruning techniques and experimentally evaluated their effectiveness in Section 7.1. First-word language model look-ahead is guaranteed not to produce additional search errors and increases speed by a factor of two. Phrase-only language model look-ahead is more than four times faster than the baseline, but results in a slightly lower translation quality due to search errors. Hybrid language model look-ahead combines the advantages of both previously mentioned techniques. It offers nearly the translation speed of phrase-only look-ahead with negligible loss in translation quality compared to the baseline, resulting in the best quality-speed tradeoff among the presented techniques. The results were analyzed and verified in detail by measuring the number of language model queries and generated partial hypotheses. In final experiments our system is between 2.5 and 22 times faster than the popular Moses toolkit, depending on how much pruning is applied. The look-ahead techniques were implemented into the translation toolkit *Jane* and are used in all phrase-based translation systems at RWTH Aachen University.

- The experiments presented in Section 7.1 also showed that pre-sorting the phrases with heuristic language model estimates increase both translation quality and time efficiency in search, given that observation histogram pruning is applied.

- We described several novel word class based smoothing models in Section 4.2.9 and evaluated them in Section 7.2. While our experiments showed that neither of them can replace their word based counterpart, each of them improves the overall translation quality when added into the log-linear model combination. In total we observe absolute improvements of 1.4% Bleu and 1.0% Ter on the French→German Quaero task. Of particular interest is the word class language model. The smaller vocabulary allows us to increase the context length without compromising efficiency. It has become an integral part of RWTH Aachen University's translation systems.

- Section 6.2.2 described several improvements over the forced alignment phrase training technique introduced in [Wuebker & Mauser$^+$ 10]. Local language models allow a better phrase pre-selection by the decoder. Backoff phrases can be generated on-the-fly during search and lead to higher data coverage. In order to simultaneously achieve higher data coverage and more constrained search conditions, we further introduced fallback decoding runs.

- A novel length-incremental phrase training algorithm was detailed in Section 6.2.4. We used the identical models at training time as in unconstrained translation. Different from

previous work, this method does not rely on a Viterbi word alignment for initialization or phrase selection. The experiments in Section 7.3.2 provided a detailed analysis of the technique. We investigated the effect of meta-parameters, show how translation quality increases through the training iterations, and compared the resulting phrase table with the baseline phrase table in terms of size and phrase length distribution. By phrase table interpolation we were able to achieve an absolute improvement of 0.8% BLEU and 0.8% TER over the baseline on the IWSLT 2011 Arabic→English task.

- We introduced the resilient backpropagation algorithm (RPROP) for discriminative maximum expected BLEU training in Section 6.3. It was experimentally compared with growth transformation (also referred to as extended Baum-Welch), stochastic gradient descent and AdaGrad in Section 7.4. RPROP performs superior to its competitors on two tasks, in total reaching an absolute improvement over the baseline of 0.9% BLEU on IWSLT German→English, up to 0.8% BLEU on BOLT Chinese→English and up to 0.6% BLEU on the WMT German→English task. On the latter two tasks our results are competitive with or superior to previous work. Maximum expected BLEU training can also be applied as a domain adaptation technique, reaching 1.7% BLEU absolute improvement on the IWSLT German→English task.

- Section 7.4 also proved that RPROP is significantly more time and memory efficient than the previously proposed growth transformation algorithm. In our experimental setup, it is six times faster and requires only a third of the memory. In our WMT German→English experiment, we used the largest training data set applied for discriminative training in the literature.

- In the experiments in Section 6.3 we showed that the application of leave-one-out is important for maximum expected BLEU training. We observe an absolute improvement of 0.6% BLEU and 0.8% TER.

- We performed extensive large-scale comparisons with internal results as well as other research groups around the world in Sections 7.5, 7.6 and 7.7.
  (a) On the large-scale WMT 2015 German→English task (Section 7.5) we verified the effectiveness of language model look-ahead within RWTH Aachen University's current state-of-the-art translation system. We further compared the performance of the word class models, forced alignment training and discriminative training with other popular techniques. We obtained clear improvements with the word class language model and with maximum expected BLEU training, reaching results that are on par with popular neural network models [Peter & Toutounchi$^+$ 15b] and with the joint translation and reordering sequences [Guta & Alkhouli$^+$ 15]. We show that both techniques were key to RWTH Aachen University's performance in the official evaluation results.
  (b) We report results of the internal evaluation of the IBM team connected with the official BOLT evaluation in Section 7.6. Several of the novel methods described in this thesis were key components of RWTH Aachen University's evaluation system, which performed best on two out of four conditions. In the remaining two conditions, we reached the positions two and four out of seven participating systems.
  (c) The official results of the German→English and English→French machine translation tasks were given in Section 7.7. The word class language model as well as maximum expected BLEU training were essential for the performance of RWTH Aachen University's evaluation system. We submitted the best performing system according to BLEU and TER on German→English and the best system according to HTER and TER on English→French.

8.1 Outlook

In this thesis, we presented improvements to the three core tasks of statistical machine translation: modeling, search and training. The modeling problem was addressed by applying unsupervised word classes to alleviate data sparsity, which has proven particularly effective for the language model. In search, we introduced several language model look-ahead techniques for improved time efficiency. Finally, we investigated both generative and discriminative approaches to train the underlying models.

Recently, due to promising initial results [Devlin & Zbib$^+$ 14, Sutskever & Vinyals$^+$ 14], neural machine translation (NMT) has become a popular research area with many unanswered questions. From our results, we can derive several possible directions for future research on NMT. One of the major drawbacks of NMT is the typically rather small vocabulary size. It is conceivable that unsupervised word classes can be a helpful tool to address this problem. Further, to find the translation output, current NMT systems rely on a primitive beam search decoder. It is likely that the results can be improved considerably by a more structured search procedure like the one applied in this work, especially in combination with alignment mechanisms. Look-ahead techniques may further increase efficiency. With regard to training, it has already been shown that using an expected BLEU objective for neural language models can improve translation quality [Auli & Gao 14]. Extending this idea to end-to-end NMT, it seems a promising direction to train the underlying networks with an objective function that has a direct connection to translation quality, like expected BLEU. However, this is non-trivial and requires solutions to a number of technical problems.

Returning to the standard phrase-based translation paradigm, our results leave the question open, whether it is possible to combine the advantages of our generative and discriminative training approaches. The obvious idea would be to devise a method that estimates its parameters in a discriminative fashion, while also updating the set of available phrase pairs by making use of the translation decoder.

9. Individual Contributions vs. Team Work

This chapter aims at highlighting the author's individual contributions as opposed to team work with respect to previously published material, summarizing the Sections 4.3, 5.5, 6.4 and 7.10.

The word class based smoothing models detailed in Section 4.2.9 were previously published in [Wuebker & Peitz$^+$ 13b]. They were developed by the author of this work, who also implemented them into the phrase-based decoder and performed the corresponding experiments, while the implementation and experimentation on hierarchical phrase-based translation was performed by Stephan Peitz. Yunsu Kim later refined the implementation as part of his Master Thesis [Kim 15].

The author of this dissertation is responsible for the entire implementation of the phrase-based search algorithm described in Chapter 5 into RWTH Aachen University's open source toolkit *Jane* [Vilar & Stein$^+$ 10, Wuebker & Huck$^+$ 12]. The look-ahead techniques (cf. Section 5.3) were previously published in [Wuebker & Ney$^+$ 12b] with the exception of hybrid look-ahead. The original idea of first-word language model look-ahead is due to Richard Zens, while the author of this work developed phrase-only and hybrid look-ahead and performed the entire implementation and experimentation.

The generative training technique introduced in Section 6.2.1 was initially developed as part of the author's diploma thesis [Wuebker 09] and was also published in [Wuebker & Mauser$^+$ 10]. The idea and the implementation of the forward-backward model (cf. Section 6.2.3) are due to Arne Mauser, while the remaining implementation and all experiments were executed by the author of this thesis. The refinements of the procedure and the length-incremental training algorithm published in [Wuebker & Ney 12a, Wuebker & Hwang$^+$ 12, Wuebker & Ney 13] were solely developed by the author of this dissertation. The author is also responsible for the entire re-implementation into RWTH Aachen University's open source toolkit *Jane*.

The discriminative training technique with the RPROP algorithm detailed in Section 6.3 was joint work of the author of this thesis with Sebastian Mühr and Patrick Lehnen and published in [Wuebker & Muehr$^+$ 15]. The author contributed the original idea, Patrick Lehnen suggested using the RPROP algorithm, L_2-regularization and condensation into separate log-linear models, while Sebastian Mühr executed the initial implementation as part of his Master Thesis [Muehr 15] under the author's supervision. Later, the author contributed significant improvements to the implementation resulting in more efficient software that allowed large training sets. The author further integrated additional update strategies for comparative experiments. Among the discriminative models introduced in Section 4.2.10, Sebastian Mühr developed the phrasal, lexical and triplet features, while the remainder is due to the author of this dissertation, who also performed all experiments described in Section 7.4.

The baseline system in Section 7.5 was prepared by Andreas Guta, while the results on word class models, generative training and maximum expected Bleu training were produced by the author of this work. Additional comparative experiments provided by Andreas Guta and Jan-Thorsten Peter are clearly marked in Table 7.16. RWTH Aachen University's translation systems for the BOLT evaluation (cf. Section 7.6) were trained and maintained by the author of this work.

The IWSLT 2014 evaluation systems (cf. Section 7.7) are described in [Wuebker & Peitz$^+$ 14]. The German→English translation system was trained by Stephan Peitz, while the English→French baseline system was provided by Andreas Guta and then augmented with maximum expected BLEU training and recurrent neural network models by the author of this thesis.

A. Overview of the Corpora

A.1 Workshop on Statistical Machine Translation

The *Workshop on Statistical Machine Translation* (WMT) is an annual evaluation campaign that provides training and test data for a number of European language pairs. Its main focus is the translation of news texts.

A.1.1 WMT 2008 German→English

Statistics for training, development and test data of the German→English task provided for the *ACL 2008 Third Workshop on Statistical Machine Translation*[1] are given in Table A.1. Training, development and test data are from the Europarl corpus [Koehn 05].

A.1.2 WMT 2011 German→English

Statistics for training, development and test data of the German→English task provided for the *EMNLP 2011 Sixth Workshop on Statistical Machine Translation*[2] are given in Table A.2. In addition to a simple tokenization scheme, we apply the frequency-based compound splitting method proposed by [Koehn & Knight 03] and the part-of-speech-based long-range reordering rules described in [Popović & Ney 06] on the German source side. `newstest2008` is used for parameter optimization, `newstest2009` as a blind test set.

A.1.3 WMT 2014 German→English

Statistics for training, development and test data of the German→English task provided for the *ACL 2014 Ninth Workshop on Statistical Machine Translation*[3] are presented in Table A.3. We apply the same preprocessing as in Section A.1.2. We use the concatenated `newstest2011` and `newstest2012` for parameter optimization and `newstest2013` and `newstest2014` as test sets. `newstest2008` through `newstest2010` are incorporated in the training data.

A.1.4 WMT 2015 German→English

Statistics for training, development and test data of the German→English task provided for the *EMNLP 2015 Tenth Workshop on Statistical Machine Translation*[4] are presented in Table A.4. We apply the same preprocessing as in Section A.1.2. We use the concatenated `newstest2011` and `newstest2012` for parameter optimization and `newstest2013`, `newstest2014` and `newstest2015` as test sets. `newstest2008` through `newstest2010` are incorporated in the training data.

[1] `http://www.statmt.org/wmt08`
[2] `http://www.statmt.org/wmt11`
[3] `http://www.statmt.org/wmt14`
[4] `http://www.statmt.org/wmt15`

Table A.1: Corpus statistics for the bilingual data provided for WMT 2008. All training and test data is taken from European parliament speeches. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	1.3M	
	Run. Words	34M	36M
	Vocabulary	336K	118K
	Singletons	169K	48K
dev:	Sentences	2000	
	Run. Words	55K	59K
	Vocabulary	9211	6549
	OOVs	284 (0.5%)	77 (0.1%)
test:	Sentences	2000	
	Run. Words	57K	60K
	Vocabulary	9254	6497
	OOVs	266 (0.5%)	89 (0.1%)

Table A.2: Corpus statistics for the bilingual data provided for WMT 2011. `newstest2008` is used as development set (dev) and `newstest2009` as blind test set (test). Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	1.9M	
	Running Words	50M	51M
	Vocabulary Size	215K	137K
	Singletons	92K	56K
dev:	Sentences	2051	
	Running Words	50K	50K
	Vocabulary Size	10038	8196
	OOVs	1358 (2.7%)	1282 (2.6%)
test:	Sentences	2525	
	Running Words	66K	66K
	Vocabulary Size	11743	9434
	OOVs	1739 (2.6%)	1681 (2.5%)

A.2 International Workshop on Spoken Language Translation

The *International Workshop on Spoken Language Translation* (IWSLT) is an annual evaluation campaign that focuses on the translation of technical lectures, the *TED talks*[5]. In addition to several European language pairs, it covers Arabic-English and Chinese-English.

[5] `http://www.ted.com/talks`

Table A.3: Corpus statistics for the bilingual data provided for WMT 2014. We use the concatenation of `newstest2011` and `newstest2012` as development set (dev). Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	4.09M	
	Running Words	105M	104M
	Vocabulary	659K	649K
	Singletons	324K	344K
dev:	Sentences	6006	
	Running Words	155K	148K
	Vocabulary	20.4K	15.7K
	OOVs	1362 (0.9%)	1480 (1.0%)
`newstest2013`:	Sentences	3000	
	Running Words	67.1K	64.9K
	Vocabulary	11342	9314
	OOVs	520 (0.8%)	545 (0.8%)
`newstest2014`:	Sentences	3003	
	Running Words	68.6K	68.0K
	Vocabulary	12019	10096
	OOVs	567 (0.8%)	839 (1.2%)

A.2.1 IWSLT 2011 English→French

Data statistics for the IWSLT 2011[6] shared task are shown in Table A.5.

A.2.2 IWSLT 2011 Arabic→English

Data statistics for the IWSLT 2011 Arabic→English shared task are shown in Table A.6. Our bilingual training data is composed of all available in-domain (TED) data and a selection of the out-of-domain MultiUN data provided for the evaluation campaign. The bilingual data selection is based on [Axelrod & He$^+$ 11].

A.2.3 IWSLT 2012 German→English

Data statistics for the IWSLT 2012[7] shared task are shown in Table A.7. For rapid experimentation, the training data can be limited to the in-domain portion as shown in Table A.8.

A.2.4 IWSLT 2013 German→English

Data statistics for the in-domain (TED talk) portion of the IWSLT 2013[8] shared task are shown in Table A.9. Statistics for all available training data are given in Table A.10.

[6] `http://iwslt2011.org`
[7] `http://hltc.cs.ust.hk/iwslt/`
[8] `http://www.iwslt2013.org`

Table A.4: Corpus statistics for the bilingual data provided for WMT 2015. We use the concatenation of `newstest2011` and `newstest2012` as development set (dev). Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	4.22M	
	Running Words	106M	108M
	Vocabulary	814K	773K
	Singletons	324K	344K
dev:	Sentences	6006	
	Running Words	150K	148K
	Vocabulary	22.0K	15.2K
	OOVs	1262 (0.8%)	1364 (0.9%)
`newstest2013`:	Sentences	3000	
	Running Words	65.3K	64.9K
	Vocabulary	11910	8973
	OOVs	480 (0.7%)	486 (0.7%)
`newstest2014`:	Sentences	3003	
	Running Words	66.4K	67.7K
	Vocabulary	12834	9720
	OOVs	551 (0.8%)	770 (1.1%)
`newstest2015`:	Sentences	2169	
	Running Words	46.2K	47.0K
	Vocabulary	9446	7543
	OOVs	445 (1.0%)	560 (1.2%)

Table A.5: Corpus Statistics for the bilingual data provided for IWSLT 2011. Out-of-vocabulary (OOV) numbers refer to running words.

		English	French
train:	Sentences	2.0M	
	Running Words	54M	60M
	Vocabulary	136K	159K
	Singletons	56K	61K
dev:	Sentences	934	
	Running Words	20K	20K
	Vocabulary	3267	3717
	OOVs	502 (2.5%)	202 (1.0%)
test:	Sentences	1664	
	Running Words	32K	34K
	Vocabulary	3782	4678
	OOVs	624 (2.0%)	170 (0.5%)

Table A.6: Statistics for the IWSLT 2011 Arabic→English data. Out-of-vocabulary (OOV) numbers refer to running words.

		Arabic	English
train:	Sentences	305K	
	Running Words	6.5M	6.5M
	Vocabulary	104K	74K
	Singletons	43K	31K
dev:	Sentences	934	
	Running Words	19K	20K
	Vocabulary	4293	3182
	OOVs	445 (2.3%)	182 (0.9%)
test:	Sentences	1664	
	Running Words	31K	32K
	Vocabulary	5415	3650
	OOVs	658 (2.1%)	159 (0.5%)

Table A.7: Corpus Statistics for the bilingual data provided for IWSLT 2012. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	2.2M	
	Running Words	58M	58M
	Vocabulary	212K	144K
dev:	Sentences	883	
	Running Words	20K	20K
	Vocabulary	4021	3233
	OOVs	215 (1.0%)	145 (0.7%)
test:	Sentences	1565	
	Running Words	31K	32K
	Vocabulary	4901	3764
	OOVs	227 (0.7%)	153 (0.5%)

A.2.5 IWSLT 2014 German→English

Table A.11 presents the corpus statistics for the full bilingual training data provided for the IWSLT 2014 German→English task[9].

A.2.6 IWSLT 2014 English→French

We summarize the the data statistics for the full bilingual training data provided for the IWSLT 2014 English→French task[10] in Table A.12.

[9] http://www.iwslt2014.org
[10] http://www.iwslt2014.org

Table A.8: Corpus Statistics for the in-domain TED data of the IWSLT 2012 German→English task. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	130K	
	Running Words	2.5M	2.5M
	Vocabulary	71K	49K
dev:	Sentences	883	
	Running Words	19.6K	20.2K
	Vocabulary	4021	3233
	OOVs	398 (2.0%)	291 (1.4%)
test:	Sentences	1565	
	Running Words	31.0K	32.0K
	Vocabulary	4901	3764
	OOVs	483 (1.6%)	297 (0.9%)

Table A.9: Corpus Statistics for the bilingual in-domain (TED) data provided for the IWSLT 2013 German→English task. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	138K	
	Running Words	2.6M	2.7M
	Vocabulary	75K	50K
dev:	Sentences	887	
	Running Words	19.6K	20.1K
	Vocabulary	4048	3207
	OOVs	386 (2.0%)	282 (1.4%)
test:	Sentences	1565	
	Running Words	31.1K	32.0K
	Vocabulary	4949	3758
	OOVs	478 (1.5%)	277 (0.9%)
tst2011:	Sentences	1436	
	Running Words	27.1K	27.1K
	Vocabulary	4611	3520
	OOVs	441 (1.6%)	329 (1.2%)

A.3 Quaero

Quaero[11] is a large-scale research project funded by Oséo, the french agency for innovation. The internal evaluations are performed on the three languages French, German and English.

[11] http://www.quaero.org

Table A.10: Corpus Statistics for all bilingual training data provided for the IWSLT 2013 German→English task. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	4.3M	
	Running Words	108M	109M
	Vocabulary	836K	792K
dev:	Sentences	887	
	Running Words	19.6K	20.1K
	Vocabulary	4048	3207
	OOVs	96 (0.5%)	82 (0.4%)
test:	Sentences	1565	
	Running Words	31.1K	32.0K
	Vocabulary	4949	3758
	OOVs	94 (0.3%)	68 (0.2%)

Table A.11: Corpus Statistics for the bilingual data provided for the IWSLT 2014 German→English task. Out-of-vocabulary (OOV) numbers refer to running words.

		German	English
train:	Sentences	4.4M	
	Running Words	107M	109M
	Vocabulary	821K	778K
dev:	Sentences	887	
	Running Words	19.4K	20.1K
	Vocabulary	3994	3181
	OOVs	83 (0.4%)	81 (0.4%)
test:	Sentences	1565	
	Running Words	30.9K	32.0K
	Vocabulary	4886	3697
	OOVs	64 (0.2%)	67 (0.2%)

A.3.1 Quaero 2012 French→German

In addition to some project-internal web-crawled data, we leverage all data available for the *NAACL 2012 Seventh Workshop on Statistical Machine Translation*[12]. The bilingual training data is sentence-aligned by using English as a pivot, discarding all non-matching sentences. Data statistics are shown in Table A.13. The development and test sets are selected from internal data sources and contain two reference translations.

[12] `http://statmt.org/wmt12/`

Table A.12: Corpus Statistics for the bilingual data provided for the IWSLT 2014 English→French task. Out-of-vocabulary (OOV) numbers refer to running words.

		English	French
train:	Sentences	26.0M	
	Running Words	698M	810M
	Vocabulary	2.1M	2.1M
dev:	Sentences	934	
	Running Words	20.1K	20.3K
	Vocabulary	3189	3690
	OOVs	45 (0.2%)	63 (0.3%)
test:	Sentences	1664	
	Running Words	32.0K	33.8K
	Vocabulary	3682	4631
	OOVs	39 (0.1%)	42 (0.1%)

Table A.13: Corpus statistics for the Quaero 2012 French→German task. Out-of-vocabulary (OOV) numbers refer to running words.

		French	German
train:	Sentences	1.9M	
	Running Words	57M	50M
	Vocabulary	160K	383K
dev:	Sentences	1900	
	Running Words	61K	55K
	Vocabulary	8165	9976
	OOVs	277 (0.5%)	1030 (1.9%)
test:	Sentences	2037	
	Running Words	60K	54K
	Vocabulary	8517	10228
	OOVs	279 (0.5%)	992 (1.8%)

A.4 BOLT

The DARPA BOLT (Broad Operational Language Translation) Program aims to develop genre-independent machine translation systems. BOLT is particularly concerned with improving translation for less-formal genres with user-contributed content, e.g. discussion forums and chats.

A.4.1 BOLT Chinese→English

Table A.14 presents the corpus statistics for the Chinese→English BOLT task. DEV10-tune and DEV10-dev contain web-data. NIST MT06 is the official test set of the NIST 2006 open machine translation evaluation[13]. For DEV10-tune, DEV10-dev and MT06 we consider only the first of four reference translations in the statistics in order for the numbers to be comparable and

[13] http://www.itl.nist.gov/iad/mig/tests/mt/2006/

Table A.14: Corpus statistics for the Chinese→English BOLT task. Out-of-vocabulary (OOV) numbers refer to running words.

		Chinese	English
train:	Sentences	4.08M	
	Running Words	78.3M	85.9M
	Vocabulary	384K	817K
DEV10-tune:	Sentences	1275	
	Running Words	33.2K	40.6K
	Vocabulary	5885	5147
	OOVs	130 (0.4%)	72 (0.2%)
DEV10-dev:	Sentences	1239	
	Running Words	32.5K	34.3K
	Vocabulary	5940	5113
	OOVs	110 (0.3%)	136 (0.4%)
NIST MT06:	Sentences	1664	
	Running Words	38.5K	46.8K
	Vocabulary	6470	5581
	OOVs	754 (2.0%)	168 (0.4%)
P1R6-dev:	Sentences	1124	
	Running Words	23.4K	25.2K
	Vocabulary	4261	3679
	OOVs	8 (0.03%)	227 (0.9%)

Table A.15: Corpus statistics for the discussion-forum portion of the Chinese→English BOLT training data. Out-of-vocabulary (OOV) numbers refer to running words.

		Chinese	English
train:	Sentences	67.8K	
	Running Words	1.45M	1.81M
	Vocabulary	41.2K	29.8K

interpretable. P1R6-dev is taken from the discussion forum domain. A small part of the training data is also from the discussion forum domain, see Table A.15. IBM did not provide the P1R6-dev reference translation in the format required to compute the statistics.

List of Figures

List of Tables

B. Bibliography

[Auli & Galley+ 14] M. Auli, M. Galley, J. Gao: Large-scale Expected BLEU Training of Phrase-based Reordering Models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1250–1260, Doha, Qatar, Oct. 2014.

[Auli & Gao 14] M. Auli, J. Gao: Decoder Integration and Expected BLEU Training for Recurrent Neural Network Language Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 136–142, Baltimore, Maryland, 2014.

[Axelrod & He+ 11] A. Axelrod, X. He, J. Gao: Domain Adaptation via Pseudo In-Domain Data Selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 355–362, Edinburgh, Scotland, UK, July 2011.

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

[Berger & Brown+ 96] A.L. Berger, P.F. Brown, S.A.D. Pietra, V.J.D. 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.

[Birch & Callison-Burch+ 06] A. Birch, C. Callison-Burch, M. Osborne, P. Koehn: Constraining the Phrase-Based, Joint Probability Statistical Translation Model. In *NAACL 2006 Workshop on Statistical Machine Translation*, pp. 154–157, New York City, NY, June 2006.

[Blunsom & Cohn+ 08] P. Blunsom, T. Cohn, M. Osborne: A Discriminative Latent Variable Model for Statistical Machine Translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 200–208, Columbus, Ohio, June 2008.

[Blunsom & Cohn+ 09] P. Blunsom, T. Cohn, C. Dyer, M. Osborne: A Gibbs Sampler for Phrasal Synchronous Grammar Induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 782–790, Suntec, Singapore, Aug. 2009.

[Bojar & Buck+ 14] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, A. Tamchyna: Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA, June 2014.

[Boudahmane & Buschbeck+ 11] K. Boudahmane, B. Buschbeck, E. Cho, J.M. Crego, M. Freitag, T. Lavergne, H. Ney, J. Niehues, S. Peitz, J. Senellart, A. Sokolov, A. Waibel, T. Wandmacher, J. Wuebker, F. Yvon: Advances on Spoken Language Translation in the Quaero Program.

In *International Workshop on Spoken Language Translation*, pp. 114–120, San Francisco, CA, USA, Dec. 2011.

[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.

[Callison-Burch & Fordyce+ 08] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, J. Schroeder: Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 70–106, Columbus, Ohio, USA, June 2008.

[Casacuberta & Vidal 04] F. Casacuberta, E. Vidal: Machine translation with Inferred Stochastic Finite-state Transducers. *Computational Linguistics*, Vol. 30, No. 2, pp. 205–225, June 2004.

[Cettolo & Niehues+ 14] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, M. Federico: Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014. In *International Workshop on Spoken Language Translation*, pp. 2–17, Lake Tahoe, CA, USA, Dec. 2014.

[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, Aug. 1998.

[Chen & Kuhn+ 11] B. Chen, R. Kuhn, G. Foster, H. Johnson: Unpacking and Transforming Feature Functions: New Ways to Smooth Phrase Tables. In *MT Summit XIII*, pp. 269–275, Xiamen, China, Sept. 2011.

[Cherry 13] C. Cherry: Improved Reordering for Phrase-Based Translation using Sparse Features. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pp. 22–31, Atlanta, Georgia, USA, June 2013.

[Cherry & Moore+ 12] C. Cherry, R.C. Moore, C. Quirk: On Hierarchical Re-ordering and Permutation Parsing for Phrase-based Decoding. In *Seventh Workshop on Statistical Machine Translation*, pp. 200–209, Montréal, Canada, June 2012.

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

[Chiang & Marton+ 08] D. Chiang, Y. Marton, P. Resnik: Online Large-margin Training of Syntactic and Structural Translation Features. In *Conference on Empirical Methods in Natural Language Processing*, pp. 224–233, Honolulu, Hawaii, Oct. 2008.

[Clark & Dyer+ 11] J.H. Clark, C. Dyer, A. Lavie, N.A. Smith: Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *49th Annual Meeting of the Association for Computational Linguistics*, pp. 176–181, Portland, Oregon, June 2011.

[Delaney & Shen+ 06] B. Delaney, W. Shen, T. Anderson: An Efficient Graph Search Decoder for Phrase-Based Statistical Machine Translation. In *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov. 2006.

[Dempster & Laird+ 77] A.P. Dempster, N.M. Laird, D.B. Rubin: Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. Ser. B*, Vol. 39, No. 1, pp. 1–22, 1977.

[DeNero & Buchard-Côté+ 08] J. DeNero, A. Buchard-Côté, D. Klein: Sampling Alignment Structure under a Bayesian Translation Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 314–323, Honolulu, Hawaii, October 2008.

[DeNero & Gillick+ 06] J. DeNero, D. Gillick, J. Zhang, D. Klein: Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings of the Workshop on Statistical Machine Translation*, pp. 31–38, New York City, NY, June 2006.

[DeNero & Klein 08] J. DeNero, D. Klein: The Complexity of Phrase Alignment Problems. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pp. 25–28, Morristown, NJ, USA, 2008.

[Devlin & Zbib+ 14] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, J. Makhoul: Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1370–1380, Baltimore, Maryland, June 2014.

[Duan & Li+ 12] N. Duan, M. Li, M. Zhou: Forced Derivation Tree based Model Training to Statistical Machine Translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 445–454, Jeju Island, Korea, July 2012.

[Duchi & Hazan+ 11] J. Duchi, E. Hazan, Y. Singer: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159, July 2011.

[Duchi & Singer 09] J. Duchi, Y. Singer: Efficient Online and Batch Learning Using Forward Backward Splitting. *Journal of Machine Learning Research*, Vol. 10, pp. 2899–2934, Dec. 2009.

[Ehling & Zens+ 07] N. Ehling, R. Zens, H. Ney: Minimum Bayes Risk Decoding for BLEU. In *45th Annual Meeting of the Association for Computational Linguistics*, pp. 101–104, Prague, Czech Republic, June 2007.

[Feng & Cohn 13] Y. Feng, T. Cohn: A Markov Model of Machine Translation using Non-parametric Bayesian Inference. In *51st Annual Meeting of the Association for Computational Linguistic*, pp. 333–342, Sofia, Bulgaria, Aug. 2013.

[Feng & Schmidt+ 11] M. Feng, C. Schmidt, J. Wuebker, S. Peitz, M. Freitag, H. Ney: The RWTH Aachen System for NTCIR-9 PatentMT. In *Ninth NTCIR Workshop Meeting*, pp. 600–605, Tokyo, Japan, Dec. 2011.

[Feng & Schmidt+ 13] M. Feng, C. Schmidt, J. Wuebker, M. Freitag, H. Ney: The RWTH Aachen System for NTCIR-10 PatentMT. In *Proceedings of the 10th NTCIR Conference*, pp. 309–314, Tokyo, Japan, June 2013.

[Ferrer & Juan 09] J. Ferrer, A. Juan: A Phrase-based Hidden Semi-Markov Approach to Machine Translation. In *Procedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT)*, pp. 168–175, Barcelona, Spain, May 2009.

[Freitag & Huck+ 14] M. Freitag, M. Huck, H. Ney: Jane: Open Source Machine Translation System Combination. In *14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 29–32, Gothenburg, Sweden, April 2014.

[Freitag & Leusch+ 11] M. Freitag, G. Leusch, J. Wuebker, S. Peitz, H. Ney, T. Herrmann, J. Niehues, A. Waibel, A. Allauzen, G. Adda, J.M. Crego, B. Buschbeck, T. Wandmacher, J. Senellart: Joint WMT Submission of the QUAERO Project. In *EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pp. 358–364, Edinburgh, UK, July 2011.

[Freitag & Peitz+ 13] M. Freitag, S. Peitz, J. Wuebker, H. Ney, N. Durrani, M. Huck, P. Koehn, T.L. Ha, J. Niehues, M. Mediani, T. Herrmann, A. Waibel, N. Bertoldi, M. Cettolo, M. Federico: EU-BRIDGE MT: Text Translation of Talks in the EU-BRIDGE Project. In *International Workshop on Spoken Language Translation*, pp. 128–135, Heidelberg, Germany, Dec. 2013.

[Freitag & Peitz+ 14] M. Freitag, S. Peitz, J. Wuebker, H. Ney, M. Huck, R. Sennrich, N. Durrani, M. Nadejde, P. Williams, P. Koehn, T. Herrmann, E. Cho, A. Waibel: EU-BRIDGE MT: Combined Machine Translation. In *ACL 2014 Ninth Workshop on Statistical Machine Translation*, pp. 105–113, Baltimore, MD, USA, June 2014.

[Freitag & Wuebker+ 14] M. Freitag, J. Wuebker, S. Peitz, H. Ney, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. Cho, A. Waibel, N. Bertoldi, M. Cettolo, M. Federico: Combined Spoken Language Translation. In *International Workshop on Spoken Language Translation*, pp. 57–64, Lake Tahoe, CA, USA, Dec. 2014.

[Galley & Manning 08] M. Galley, C.D. Manning: A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 848–856, Honolulu, Hawaii, 2008.

[Gao & He 13] J. Gao, X. He: Training MRF-Based Phrase Translation Models using Gradient Ascent. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, pp. 450–459, Atlanta, Georgia, USA, Jun 2013.

[Green & Cer+ 14] S. Green, D. Cer, C.D. Manning: An Empirical Comparison of Features and Tunign for Phrase-based Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 466–476, Baltimore, Maryland, USA, June 2014.

[Green & Chuang+ 14] S. Green, J. Chuang, J. Heer, C.D. Manning: Predictive Translation Memory: A Mixed-initiative System for Human Language Translation. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pp. 177–187, New York, NY, USA, 2014.

[Green & Wang+ 13] S. Green, S. Wang, D. Cer, C.D. Manning: Fast and Adaptive Online Training of Feature-Rich Translation Models. In *51st Annual Meeting of the Association for Computational Linguistics*, pp. 311–321, Sofia, Bulgaria, Aug. 2013.

[Guta & Alkhouli+ 15] A. Guta, T. Alkhouli, J.T. Peter, J. Wuebker, H. Ney: A Comparison between Count and Neural Network Models Based on Joint Translation and Reordering Sequences. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1401–1411, Lisbon, Portugal, Sept. 2015.

[Guta & Wuebker+ 15] A. Guta, J. Wuebker, M. Graa, Y. Kim, H. Ney: Extended Translation Models in Phrase-based Decoding. In *EMNLP 2015 Tenth Workshop on Statistical Machine Translation*, pp. 282–293, Lisbon, Portugal, Sept. 2015.

[Hahn & Dinarelli+ 11] S. Hahn, M. Dinarelli, C. Raymond, F. Lefevre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, G. Riccardi: Comparing Stochastic Approaches to Spoken Language Understanding in Multiple Languages. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19, No. 6, pp. 1569–1583, Aug. 2011.

[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, Oct. 2008.

[He & Deng 12] X. He, L. Deng: Maximum Expected BLEU Training of Phrase and Lexicon Translation Models. In *50th Annual Meeting of the Association for Computational Linguistics*, pp. 292–301, Jeju, Republic of Korea, July 2012.

[Heafield 11] K. Heafield: KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197, Edinburgh, Scotland, UK, July 2011.

[Heafield & Pouzyrevsky+ 13] K. Heafield, I. Pouzyrevsky, J.H. Clark, P. Koehn: Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 690–696, Sofia, Bulgaria, August 2013.

[Heger & Wuebker+ 10a] C. Heger, J. Wuebker, M. Huck, G. Leusch, S. Mansour, D. Stein, H. Ney: The RWTH Aachen Machine Translation System for WMT 2010. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pp. 93–97, Uppsala, Sweden, July 2010.

[Heger & Wuebker+ 10b] C. Heger, J. Wuebker, D. Vilar, H. Ney: A Combination of Hierarchical Systems with Forced Alignments from Phrase-Based Systems. In *International Workshop on Spoken Language Translation*, pp. 291–297, Paris, France, Dec. 2010.

[Heigold & Hahn+ 11] G. Heigold, S. Hahn, P. Lehnen, H. Ney: EM-Style Optimization of Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4920–4923, Prague, Czech Republic, May 2011.

[Hopkins & May 11] M. Hopkins, J. May: Tuning as ranking. In *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1352–1362, Edinburgh, Scotland, July 2011.

[Huck & Wuebker+ 11] M. Huck, J. Wuebker, C. Schmidt, M. Freitag, S. Peitz, D. Stein, A. Dagnelies, S. Mansour, G. Leusch, H. Ney: The RWTH Aachen Machine Translation System for WMT 2011. In *EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pp. 405–412, Edinburgh, UK, July 2011.

[Huck & Wuebker+ 13] M. Huck, J. Wuebker, F. Rietig, H. Ney: A Phrase Orientation Model for Hierarchical Machine Translation. In *Eighth Workshop on Statistical Machine Translation*, pp. 452–463, Sofia, Bulgaria, Aug. 2013.

[Ittycheriah & Roukos 07] A. Ittycheriah, S. Roukos: Direct Translation Model 2. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, pp. 57–64, Rochester, NY, USA, Apr 2007.

[Jelinek 97] F. Jelinek: *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, 1997.

[Kim 15] Y. Kim: Phrase Table Smoothing with Word Classes. Master's thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, Oct. 2015.

[Kneser & Ney 95] R. Kneser, H. Ney: Improved Backing-off for M-gram Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processingw*, Vol. 1, pp. 181–184, May 1995.

[Knight 99] K. Knight: Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, Vol. 25, No. 4, pp. 607–615, Dec. 1999.

[Koehn 04] P. Koehn: Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 388–395, Barcelona, Spain, July 2004.

[Koehn 05] P. Koehn: Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X*, Phuket, Thailand, Sept. 2005.

[Koehn & Hoang 07a] P. Koehn, H. Hoang: Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 868–876, Prague, Czech Republic, June 2007.

[Koehn & Hoang$^+$ 07b] 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 Association for Computational Linguistics*, pp. 177–180, Prague, Czech Republic, June 2007.

[Koehn & Knight 03] P. Koehn, K. Knight: Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association 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 *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pp. 127–133, Edmonton, Alberta, 2003.

[Lavergne & Allauzen$^+$ 11] T. Lavergne, A. Allauzen, J.M. Crego, F. Yvon: From n-gram-based to CRF-based Translation Models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 542–553, Edinburgh, Scotland, July 2011.

[Lehnen & Peter$^+$ 13] P. Lehnen, J.T. Peter, J. Wuebker, S. Peitz, H. Ney: (Hidden) Conditional Random Fields Using Intermediate Classes for Statistical Machine Translation. In *Machine Translation Summit XIV*, pp. 151–158, Nice, France, Sept. 2013.

[Levenshtein 66] V.I. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10, pp. 707–710, February 1966.

[Liang & Buchard-Côté$^+$ 06] P. Liang, A. Buchard-Côté, D. Klein, B. Taskar: An End-to-End Discriminative Approach to Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 761–768, Sydney, Australia, 2006.

[Mansour & Peitz$^+$ 10] S. Mansour, S. Peitz, D. Vilar, J. Wuebker, H. Ney: The RWTH Aachen Machine Translation System for IWSLT 2010. In *International Workshop on Spoken Language Translation*, pp. 163–168, Paris, France, Dec. 2010.

[Mansour & Wuebker+ 11] S. Mansour, J. Wuebker, H. Ney: Combining Translation and Language Model Scoring for Domain-Specific Data Filtering. In *International Workshop on Spoken Language Translation*, pp. 222–229, San Francisco, CA, USA, Dec. 2011.

[Marcu & Wong 02] D. Marcu, W. Wong: A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 133–139, Philadelphia, Pennsylvania, USA, July 2002.

[Mariño & Banchs+ 06] J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, M.R. Costa-jussà: N-gram-based Machine Translation. *Computational Linguistics*, Vol. 32, No. 4, pp. 527–549, Dec. 2006.

[Mauser 15] A. Mauser: *Training and Models for Statistical Machine Translation.* Ph.D. thesis, Lehrstuhl für Informatik 6, Computer Science Department, RWTH Aachen University, Aachen, Germany, 2015.

[Mediani & Zhang+ 12] M. Mediani, Y. Zhang, T.L. Ha, J. Niehues, E. Cho, T. Herrmann, A. Waibel: The KIT Translation Systems for IWSLT 2012. In *Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2012)*, Hong Kong, 2012.

[Moore & Lewis 10] R.C. Moore, W. Lewis: Intelligent Selection of Language Model Training Data. In *48th Annual Meeting of the Association for Computational Linguistics*, pp. 220–224, Uppsala, Sweden, July 2010.

[Moore & Quirk 07a] R.C. Moore, C. Quirk: An Iteratively-Trained Segmentation-Free Phrase Translation Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 112–119, Prague, June 2007.

[Moore & Quirk 07b] R.C. Moore, C. Quirk: Faster Beam-Search Decoding for Phrasal Statistical Machine Translation. In *Proceedings of MT Summit XI*, Copenhagen, Denmark, Sept. 2007.

[Muehr 15] S. Muehr: Discriminative Phrase Training for Statistical Machine Translation. Master's thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, Oct. 2015.

[Mylonakis & Sima'an 08] M. Mylonakis, K. Sima'an: Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 630–639, Honolulu, Hawaii, Oct. 2008.

[Neubig & Watanabe+ 11] G. Neubig, T. Watanabe, E. Sumita, S. Mori, T. Kawahara: An Unsupervised Model for Joint Phrase Alignment and Extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pp. 632–641, Portland, Oregon, 2011.

[Ney 01] H. Ney: Stochastic Modelling: From Pattern Classification to Language Translation. In *39th Annual Meeting of the Association for Computational Linguistics - joint with EACL 2001: Proceedings of the Workshop on Data-Driven Machine Translation*, pp. 1–5, Morristown, NJ, July 2001.

[Nocedal & Wright 99] J. Nocedal, S.J. Wright: *Numerical Optimization.* Springer, 1999.

[Nolden & Ney+ 11] D. Nolden, H. Ney, R. Schlüter: Exploiting Sparseness of Backing-Off Language Models for Efficient Look-Ahead in LVCSR. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4684–4687, Prague, Czech Republic, May 2011.

[Och 99] F.J. Och: An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 8–12, Bergen, Norway, June 1999.

[Och 03] F.J. Och: Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 160–167, Sapporo, Japan, July 2003.

[Och & Ney 02] F.J. Och, H. Ney: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 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, Dec. 2004.

[Och & Tillmann$^+$ 99] F.J. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, June 1999.

[Ortiz & Garcia-Varea$^+$ 06] D. Ortiz, I. Garcia-Varea, F. Casacuberta: Generalized Stack Decoding Algorithms for Statistical Machine Translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pp. 64–71, New York City, NY, June 2006.

[Ortmanns & Ney$^+$ 98] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. In *International Conference on Spoken Language Processing*, pp. 2095–2098, Sydney, Australia, Oct. 1998.

[Papineni & Roukos$^+$ 98] K.A. Papineni, S. Roukos, R.T. Ward: Maximum Likelihood and Discriminative Training of Direct Translation Models. In *Proceedings of the IEEE International Conference 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 *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002.

[Peitz & Mansour$^+$ 12] S. Peitz, S. Mansour, M. Freitag, M. Feng, M. Huck, J. Wuebker, M. Nuhn, M. Nußbaum-Thom, H. Ney: The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012. In *International Workshop on Spoken Language Translation*, pp. 69–76, Hong Kong, Dec. 2012.

[Peitz & Mansour$^+$ 13] S. Peitz, S. Mansour, J.T. Peter, C. Schmidt, J. Wuebker, M. Huck, M. Freitag, H. Ney: The RWTH Aachen Machine Translation System for WMT 2013. In *ACL 2013 Eighth Workshop on Statistical Machine Translation*, pp. 193–199, Sofia, Bulgaria, Aug. 2013.

[Peitz & Mauser$^+$ 12] S. Peitz, A. Mauser, J. Wuebker, H. Ney: Forced Derivations for Hierarchical Machine Translation. In *24th International Conference on Computational Linguistics*, pp. 933–942, Mumbai, India, Dec. 2012.

[Peitz & Wuebker+ 14] S. Peitz, J. Wuebker, M. Freitag, H. Ney: The RWTH Aachen German-English Machine Translation System for WMT 2014. In *ACL 2014 Ninth Workshop on Statistical Machine Translation*, pp. 157–162, Baltimore, MD, USA, June 2014.

[Peter & Toutounchi+ 15a] J.T. Peter, F. Toutounchi, S. Peitz, P. Bahar, A. Guta, H. Ney: The RWTH Aachen German to English MT System for IWSLT 2015. In *International Workshop on Spoken Language Translation*, pp. 15–22, Da Nang, Vietnam, Dec. 2015.

[Peter & Toutounchi+ 15b] J.T. Peter, F. Toutounchi, J. Wuebker, H. Ney: The RWTH Aachen German-English Machine Translation System for WMT 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 158–163, Lisbon, Portugal, September 2015.

[Popović & Ney 06] M. Popović, H. Ney: POS-based Word Reorderings for Statistical Machine Translation. In *5th International Conference on Language Resources and Evaluation*, pp. 1278–1283, Genoa, Italy, May 2006.

[Riedmiller & Braun 93] M. Riedmiller, H. Braun: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *IEEE International Conference on Neural Networks*, pp. 586–591, San Francisco, CA, USA, March 1993.

[Rietig 13] F. Rietig: Lexicalized Reordering Models For Phrase-based Statistical Machine Translation. Master's thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, Feb. 2013.

[Saers & Wu 11] M. Saers, D. Wu: Principled Induction of Phrasal Bilexica. In *Proceedings of the 15th International Conference of the European Association for Machine Translation*, pp. 313–320, Leuven, Belgium, May 2011.

[Setiawan & Zhou 13] H. Setiawan, B. Zhou: Discriminative Training of 150 Million Translation Parameters and Its Application to Pruning. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 335–341, Atlanta, GA, USA, June 2013.

[Simianer & Riezler+ 12] P. Simianer, S. Riezler, C. Dyer: Joint Feature Selection in Distributed Stochastic Learning for Large-Scale Discriminative Training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11–21, Jeju Island, Korea, July 2012.

[Snover & Dorr+ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul: A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pp. 223–231, Cambridge, Massachusetts, USA, August 2006.

[Stein & Vilar+ 11] D. Stein, D. Vilar, S. Peitz, M. Freitag, M. Huck, H. Ney: A Guide to Jane, an Open Source Hierarchical Translation Toolkit. *The Prague Bulletin of Mathematical Linguistics*, Vol. 95, pp. 5–18, April 2011.

[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, Sept. 1994.

[Stolcke 02] A. Stolcke: SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pp. 901–904, Denver, Colorado, USA, Sept. 2002. ISCA.

[Sundermeyer & Alkhouli+ 14] M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney: Translation Modeling with Bidirectional Recurrent Neural Networks. In *Conference on Empirical Methods in Natural Language Processing*, pp. 14–25, Doha, Qatar, Oct. 2014.

[Sundermeyer & Ney+ 15] M. Sundermeyer, H. Ney, R. Schlüter: From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 23, No. 3, pp. 517–529, March 2015.

[Sundermeyer & Schlüter+ 12] M. Sundermeyer, R. Schlüter, H. Ney: LSTM Neural Networks for Language Modeling. In *Interspeech*, Portland, OR, USA, Sept. 2012.

[Sundermeyer & Schlüter+ 14] M. Sundermeyer, R. Schlüter, H. Ney: rwthlm - The RWTH Aachen University Neural Network Language Modeling Toolkit. In *Interspeech*, pp. 2093–2097, Singapore, Sept. 2014.

[Sutskever & Vinyals+ 14] I. Sutskever, O. Vinyals, Q. Le: Sequence to Sequence Learning with Neural Networks. In *Conference on Neural Information Processing Systems (NIPS)*, pp. 3104–3112, Montréal, Canada, Dec. 2014.

[Tillmann 04] C. Tillmann: A Unigram Orientation Model for Statistical Machine Translation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 101–104, Boston, Massachusetts, 2004.

[Tillmann & Ney 03] C. Tillmann, H. Ney: Word re-ordering and DP beam search for statistical machine translation. *Computational Linguistics*, Vol. 1, No. 29, pp. 97–133, 2003.

[Tromble & Kumar+ 08] R. Tromble, S. Kumar, F. Och, W. Macherey: Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 620–629, Honolulu, Hawaii, October 2008.

[Ueffing & Och+ 02] N. Ueffing, F. Och, H. Ney: Generation of Word Graphs in Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, pp. 156–163, Philadelphia, PA, USA, July 2002.

[Vauquois 68] B. Vauquois: A survey of formal grammars and algorithms for recognition and transformation in machine translation. In *International Federation for Information Processing Congress*, Vol. 2, pp. 254–260, Edinburgh, UK, Aug. 1968.

[Vilar & Stein+ 10] D. Vilar, D. Stein, M. Huck, H. Ney: Jane: Open Source Hierarchical Translation, Extended with Reordering and Lexicon Models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pp. 262–270, Uppsala, Sweden, July 2010.

[Vogel & Ney+ 96] S. Vogel, H. Ney, C. Tillmann: HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th conference on Computational linguistics*, Vol. 2, pp. 836–841, Copenhagen, Denmark, Aug. 1996.

[Watanabe & Suzuki+ 07] T. Watanabe, J. Suzuki, H. Tsukada, H. Isozaki: Online Large-margin Training for Statistical Machine Translation. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing*, pp. 764–773, Prague, Czech Republic, June 2007.

[Weaver 55] W. Weaver: Translation. In W.N. Locke, A.D. Booth, editors, *Machine Translation of Languages: fourteen essays*, pp. 15–23. MIT Press, Cambridge, MA, 1955.

[Wiesler & Richard+ 13] S. Wiesler, A. Richard, R. Schlüter, H. Ney: A Critical Evaluation of Stochastic Algorithms for Convex Optimization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 6955–6959, Vancouver, Canada, May 2013.

[Wuebker 09] J. Wuebker: Training Phrase Models for Statistical Machine Translation. Master's thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, April 2009.

[Wuebker & Green+ 15] J. Wuebker, S. Green, J. DeNero: Hierarchical Incremental Adaptation for Statistical Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1059–1065, Lisbon, Portugal, Sept. 2015.

[Wuebker & Huck+ 11] J. Wuebker, M. Huck, S. Mansour, M. Freitag, M. Feng, S. Peitz, C. Schmidt, H. Ney: The RWTH Aachen Machine Translation System for IWSLT 2011. In *International Workshop on Spoken Language Translation*, pp. 106–113, San Francisco, CA, USA, Dec. 2011.

[Wuebker & Huck+ 12] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.T. Peter, S. Mansour, H. Ney: Jane 2: Open Source Phrase-based and Hierarchical Statistical Machine Translation. In *24th International Conference on Computational Linguistics*, pp. 483–491, Mumbai, India, Dec. 2012.

[Wuebker & Hwang+ 12] J. Wuebker, M.Y. Hwang, C. Quirk: Leave-One-Out Phrase Model Training for Large-Scale Deployment. In *Proceedings of the NAACL 2012 Seventh Workshop on Statistical Machine Translation*, pp. 460–467, Montreal, Canada, June 2012.

[Wuebker & Mauser+ 10] J. Wuebker, A. Mauser, H. Ney: Training Phrase Translation Models with Leaving-One-Out. In *48th Annual Meeting of the Association for Computational Linguistics*, pp. 475–484, Uppsala, Sweden, July 2010.

[Wuebker & Muehr+ 15] J. Wuebker, S. Muehr, P. Lehnen, S. Peitz, H. Ney: A Comparison of Update Strategies for Large-Scale Maximum Expected BLEU Training. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1516–1526, Denver, CO, USA, June 2015.

[Wuebker & Ney 12a] J. Wuebker, H. Ney: Phrase Model Training for Statistical Machine Translation with Word Lattices of Preprocessing Alternatives. In *NAACL 2012 Seventh Workshop on Statistical Machine Translation*, pp. 450–459, Montreal, Canada, June 2012.

[Wuebker & Ney+ 12b] J. Wuebker, H. Ney, R. Zens: Fast and Scalable Decoding with Language Model Look-Ahead for Phrase-based Statistical Machine Translation. In *50th Annual Meeting of the Association for Computational Linguistics*, pp. 28–32, Jeju, Republic of Korea, July 2012.

[Wuebker & Ney 13] J. Wuebker, H. Ney: Length-incremental Phrase Training for SMT. In *EighthWorkshop on Statistical Machine Translation*, pp. 309–319, Sofia, Bulgaria, Aug. 2013.

[Wuebker & Ney+ 14] J. Wuebker, H. Ney, A. Martinez-Villaronga, A. Giménez, A. Juan, C. Servan, M. Dymetman, S. Mirkin: Comparison of Data Selection Techniques for the Translation of Video Lectures. In *The Eleventh Biennial Conference of the Association for Machine Translation in the Americas*, pp. 193–207, Vancouver, BC, Canada, Oct. 2014.

[Wuebker & Peitz+ 13a] J. Wuebker, S. Peitz, T. Alkhouli, J.T. Peter, M. Feng, M. Freitag, H. Ney: The RWTH Aachen Machine Translation Systems for IWSLT 2013. In *International Workshop on Spoken Language Translation*, pp. 88–93, Heidelberg, Germany, Dec. 2013.

[Wuebker & Peitz+ 13b] J. Wuebker, S. Peitz, F. Rietig, H. Ney: Improving Statistical Machine Translation with Word Class Models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1377–1381, Seattle, WA, USA, Oct. 2013.

[Wuebker & Peitz+ 14] J. Wuebker, S. Peitz, A. Guta, H. Ney: The RWTH Aachen Machine Translation Systems for IWSLT 2014. In *International Workshop on Spoken Language Translation*, pp. 150–154, Lake Tahoe, CA, USA, Dec. 2014.

[Yu & Huang+ 13] H. Yu, L. Huang, H. Mi, K. Zhao: Max-Violation Perceptron and Forced Decoding for Scalable MT Training. In *Conference on Empirical Methods in Natural Language Processing*, pp. 1112–1123, Seattle, USA, Oct. 2013.

[Zens 08] R. Zens: *Phrase-based Statistical Machine Translation: Models, Search, Training*. Ph.D. thesis, Lehrstuhl für Informatik 6, Computer Science Department, RWTH Aachen University, Feb. 2008.

[Zens & Ney 08] R. Zens, H. Ney: Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation. In *International Workshop on Spoken Language Translation*, pp. 195–205, Honolulu, HI, USA, Oct. 2008.