Statistical Machine Translation: From Single-Word Models to Alignment Templates

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Informatiker Franz Josef Och

aus

Ebermannstadt

Berichter:
Universitätsprofessor Dr.–Ing. Hermann Ney
Universitätsprofessor Dr.–Ing. Heinrich Niemann

Tag der mündlichen Prüfung: Dienstag, 8. Oktober 2002

Wovon man nicht sprechen kann,
darüber muß man schweigen.

Ludwig Wittgenstein, Tractatus logico-philosophicus

Acknowledgements

An erster Stelle, möchte ich mich bei meinem Doktorvater Prof. Dr.-Ing. Hermann Ney bedanken. Seine konstante Unterstützung und konstruktive Kritik waren von unschätzbaren Wert. Weiterhin möchte ich mich bei Prof. Dr.-Ing. Heinrich Niemann dafür bedanken, die Rolle des Zweitgutachters übernommen zu haben.
Besonderer Dank geht an alle meine Kollegen am Lehrstuhl für Informatik VI und allen Forscherkollegen von anderen Instituten mit denen ich im Laufe der letzten viereinhalb Jahre viele hilfreiche Diskussionen führen durfte und von denen ich zahlreiche Hilfestellungen erhalten habe: Christoph, Daniel, Florian, Gregor, Hassan, Ismael, Kevin, Klaus, Maja, Max, Michael, Nicola, Oliver, Ralf, Richard, Shankar, Shahram, Sonja, Stephan, Stephan, Wolfgang, und vielen weiteren.
Ich möchte meinen Eltern dafür danken, daß sie mir das Studium der Informatik ermöglicht und mich auf meinem Weg immer unterstützt haben.
Ganz besonderen Dank möchte ich Dimitra aussprechen, die mit ihrem Verständnis, ihren Kochkünsten und ihrer Zuneigung vielfältig und entscheidend am Erfolg dieser Arbeit beteiligt war.

Kurzfassung

In diesear Arbeit werden neue Ansätze zur Sprachübersetzung basierend auf statistischen Verfahen vorgestellt. Als Verallgemeinerung zu dem üblicherweise verwendeten Source-Channel Modell wird ein allgemeineres Modell basierend auf dem Maximum-Entropie-Prinzip vorgeschlagen.
Es werden verschiedene Verfahren zur Bestimmung von Wort-Alignments unter Nutzung von statistischen und heuristischen Modellen beschrieben. Dabei werden insbesondere verschiedene Glättungsverfahren, Methoden zur Integration zusätzlicher Lexika und Trainingsverfahren verglichen. Eine detaillierte Bewertung der Alignment-Qualität wird durchgeführt indem die automatisch erstellten Wort-Alignments mit manuell erstellten Alignments verglichen werden. Aufbauend auf diesen grundlegenden einzelwortbasierten Alignment-Modellen wird dann ein phrasenbasiertes statistisches Übersetzungsmodell, das Alignment Template Modell, vorgeschlagen. Für dieses Modell wird ein Trainingsverfahren und ein effizienter Suchalgorithmus basierend auf dem Prinzip der dynamischer Programmierung und Strahlsuche entwickelt. Weiterhin werden für zwei spezielle Anwendungsszenarien (interaktive Übersetzung und Übersetzung basierend auf verschiedenen mehrsprachigen Quelltexten) spezielle Suchverfahren entwickelt.
Der beschriebene Übersetzungsansatz wurde getestet für das deutsch-englische Verbmobil Korpus, das französisch-englische Hansards Korpus und für chinesisch-englische Nachrichtentexte. Das entwickelte System erzielt dabei häufig deutlich bessere Ergebnisse als alternative Verfahren zur maschinellen Übersetzung.

Abstract

In this work, new approaches for machine translation using statistical methods are described. In addition to the standard source-channel approach to statistical machine translation, a more general approach based on the maximum entropy principle is presented.
Various methods for computing single-word alignments using statistical or heuristic models are described. Various smoothing techniques, methods to integrate a conventional dictionary and training methods are analyzed. A detailed evaluation of these models is performed by comparing the automatically produced word alignment with a manually produced reference alignment. Based on these fundamental single-word based alignment models, a new phrase-based translation model—the alignment template model—is suggested. For this model, a training and an efficient search algorithm is developed. For two specific applications (interactive translation and multi-source translation) specific search algorithms are developed.
The suggested machine translation approach has been tested for the German-English Verbmobil task, the French-English Hansards task and for Chinese-English news text translation. Often, the obtained results are significantly better than those obtained with alternative approaches to machine translation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Machine Translation

Machine translation (MT) is the use of a computer to translate texts or utterances of a natural language into another natural language. An MT system expects texts in a specific language as input and produces a text with a corresponding meaning in a different language as output. Hence, machine translation is a decision problem where we have to decide on the best of target language text matching a source language text. This viewpoint shall be taken throughout this work.

MT has a long history [Arnold & Balkan$^+$ 94, Hutchins 95]. A broad interest on MT started after World War II, initiated by a famous publication of Warren Weaver [Weaver 55]. Despite the intensive research for a long time, it seems that many experts in the area agree that the performance of MT technology after 50 years of development leaves much to be desired [Cole & Mariani$^+$ 95].

What makes MT so hard? An important reason is that natural languages are highly complex. Many words have various meanings and different possible translations. Sentences might have various readings and the relationship between linguistic entities are often vague. In some languages such as Chinese or Japanese, not even the word boundaries are given. Certain grammatical relations in one language might not exist in another language and sentences involving these relations need to be significantly reformulated. In addition, there are nonlinguistic factors such as the problem that performing a translation might need world knowledge.

To perform MT, many dependencies have to be taken into account. Often, these dependencies are weak and vague, which makes it rarely possible to describe simple and relevant rules that hold without exception in the translation process. From a linguistic viewpoint, we have to consider various types of dependencies: morphologic, syntactic, semantic and pragmatic dependencies [Jurafsky & Martin 00]. In this work, MT is treated as a decision problem, where we have to to decide upon a sequence of target language words, given a sequence of source language words. Therefore, all these dependencies are ultimately dependencies between observable entities, namely words. More specifically, there are dependencies that relate source and target language words, which describe that certain words or phrases can be translations of each other. Some dependencies relate only target language words describing the well-formedness of the produced translation. To develop an MT system, we have to find a general framework, which is able to deal with the weak and vague dependencies. Having such a framework, we have to develop methods that allow us to obtain efficiently the large amount of relevant dependencies.

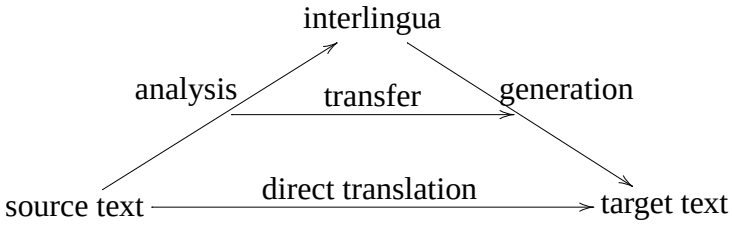


Figure 1.1: Different levels of analysis in an MT system.

1.2 Classification of MT Systems

MT systems can be distinguished according to different criteria. In the following, we distinguish the type of the input text, the application, the level of analysis and the used technology.
Most MT systems deal with text input. Here, the input text can typically be expected to be grammatical and well-formed. The task is more complicated in the case of a speech translation system. Then, the system has to deal with speech recognition errors and spontaneous speech phenomena such as ungrammatical utterances, false starts or hesitations. Therefore, a speech MT system has to be able to deal with 'wrong' input. In this thesis, we describe both, text and speech translation systems.
There are various types of applications for MT technology. In *gisting*, the aim is to produce an understandable raw translation. A possible goal is that a human is able to decide whether a foreign language text contains relevant information. To extract this information from the document, typically a human translation would be performed. In *post-editing applications*, the aim is to produce a translation that is then corrected by a human translator. In *fully automatic MT*, the computer is used to produce a high quality translation. Using state-of-the-art technology, this is only possible for very restricted domains. An example is the METEO system [Chandioux & Grimaila 96], which translates only weather forecasts from English into French and achieves a very high translation quality. In this thesis, we are concentrating on MT systems that generate understandable translations. Yet, in Chapter 10, we shall show that the developed methods are well suited for post-editing applications.
Typically, three different types of MT systems are distinguished according to the level of analysis that is performed. Figure 1.1 gives the standard visualization of the three approaches direct translation, transfer approach and interlingua approach.
The simplest approach is the *direct translation approach* where a word-by-word translation from the source language to the target language is performed. In the *transfer approach*, the translation process is decomposed into the three steps analysis, transfer and generation. In the analysis step, the input sentence is analyzed syntactically and semantically producing an abstract representation of the source sentence. In the *transfer step*, this representation is transferred into a corresponding representation in the target language. In the *generation step*, the target language sequence is produced. In the *interlingua approach*, a very fine-grained analysis produces a completely language independent representation of the input sentence. This representation is used to produce the target language sentence. An often claimed advantage of the interlingua approach is that developing translation systems between all pairs of a set of $n \gg 1$ languages is more efficient. There are only n components needed to translate into the interlingua and n components are needed to translate from it. In a transfer approach or a direct translation approach, the development of $n \cdot (n-1)$ components for each pair of languages is

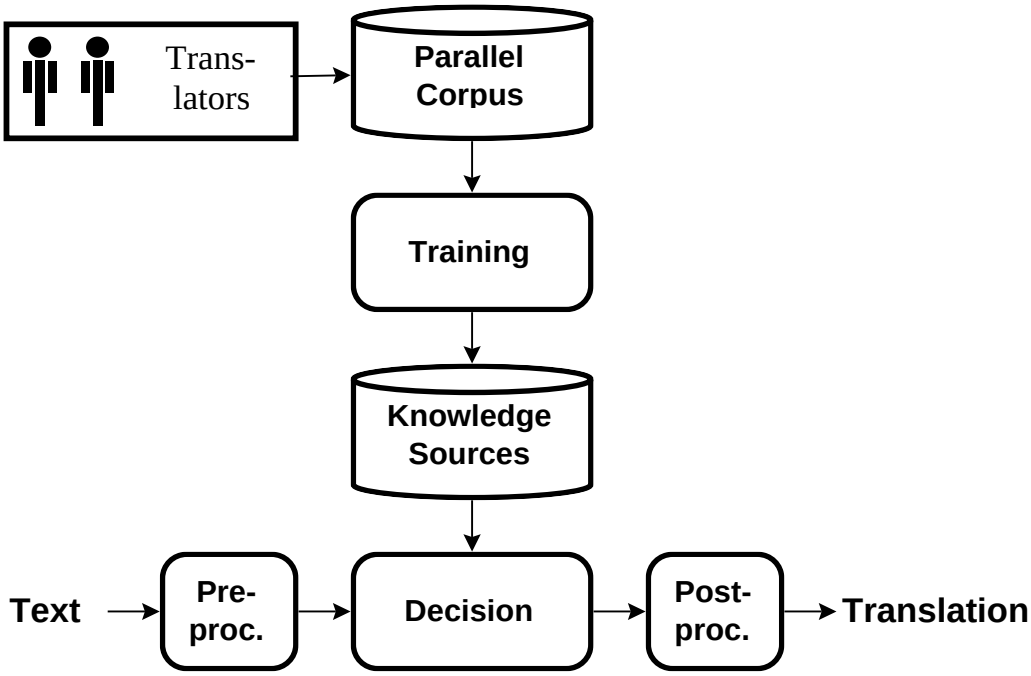


Figure 1.2: Architecture of an empirical MT system.

needed.

MT systems can be distinguished according to the core technology that is used. Here, we distinguish rule-based and empirical approaches. In the rule-based approaches, human experts specify a set of rules, which are aimed at describing the translation process. This is typically a very expensive work for which linguistic experts are needed. The rule-based approach is also predominant in existing textbooks on MT [Hutchins & Somers 92, Dorr 93, Arnold & Balkan$^+$ 94].

Using an empirical approach, the knowledge sources to develop an MT system are computed automatically by analyzing example translations. A major advantage of empirical approaches to MT is that MT systems for new language pairs and domains can be developed very quickly, provided sufficient training data is available. Figure 1.2 shows the architecture of an empirical MT system. In a fully-fledged empirical approach, the starting point is a parallel training corpus that consists of translation examples, which were produced by human translators. In the training phase, the necessary knowledge sources are computed automatically. The search or decision process has to achieve an optimal combination of the knowledge sources to perform an optimal translation. In addition, we may explicitly allow optional transformations (preprocessing) to simplify the translation task for the algorithm.

An empirical approach might pursue a direct or a transfer approach. The translation models pursued in this thesis mainly perform a refined word-by-word translation and hence follow the paradigm of the direct translation approach.

In the empirical approaches, we can distinguish example-based MT and statistical MT. In example-based MT, a translation of a new sentence is performed by analyzing similar translation examples previously seen. In statistical MT, the translation examples are used to train a statistical translation model. The decision rule used to decide for the actual translation is derived from statistical decision theoretic considerations. Here, we pursue the statistical MT

approach.

1.3 Statistical MT

The goal is the translation of a text given in some source language into a target language. We are given a source ('French') sentence $f_1^J = f_1, \ldots, f_j, \ldots, f_J$, which is to be translated into a target ('English') sentence $e_1^I = e_1, \ldots, e_i, \ldots, e_I$. Among all possible target sentences, we will choose the sentence with the highest probability:[1]

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

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language.

1.3.1 Source–Channel Model

Using Bayes' decision rule, we can equivalently to Eq. 1.1 perform the following maximization:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \tag{1.2}$$

This approach is referred to as source–channel approach to statistical MT [Brown & Cocke$^+$ 90] and sometimes also as the 'fundamental equation of statistical MT' [Brown & Della Pietra$^+$ 93b]. In the field of pattern recognition, this approach has a long history [Duda & Hart 73]. Here, $Pr(e_1^I)$ is the language model of the target language, whereas $Pr(f_1^J|e_1^I)$ is the translation model. Typically, Eq. 1.2 is favored over the direct translation model of Eq. 1.1 with the argument that it yields a modular approach. Instead of modeling one probability distribution, we obtain two different knowledge sources that are trained independently.

The overall architecture of the source–channel approach is summarized in Figure 1.3. In general, as shown in this figure, there may be additional transformations to make the translation task simpler for the algorithm. The transformations may range from the categorization of single words and word groups to more complex preprocessing steps that require some parsing.

Typically, training is performed by applying a maximum likelihood approach. If the language model $Pr(e_1^I) = p_\gamma(e_1^I)$ depends on parameters γ and the translation model $Pr(f_1^J|e_1^I) = p_\theta(f_1^J|e_1^I)$ depends on parameters θ, then the optimal parameter values are obtained by maximizing the likelihood on a parallel training corpus $\mathbf{f}_1^S, \mathbf{e}_1^S$ [Brown & Della Pietra$^+$ 93b, Och & Ney 00b]:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left\{ \prod_{s=1}^{S} p_\theta(\mathbf{f}_s|\mathbf{e}_s) \right\} \tag{1.3}$$

$$\hat{\gamma} = \underset{\gamma}{\operatorname{argmax}} \left\{ \prod_{s=1}^{S} p_\gamma(\mathbf{e}_s) \right\} \tag{1.4}$$

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

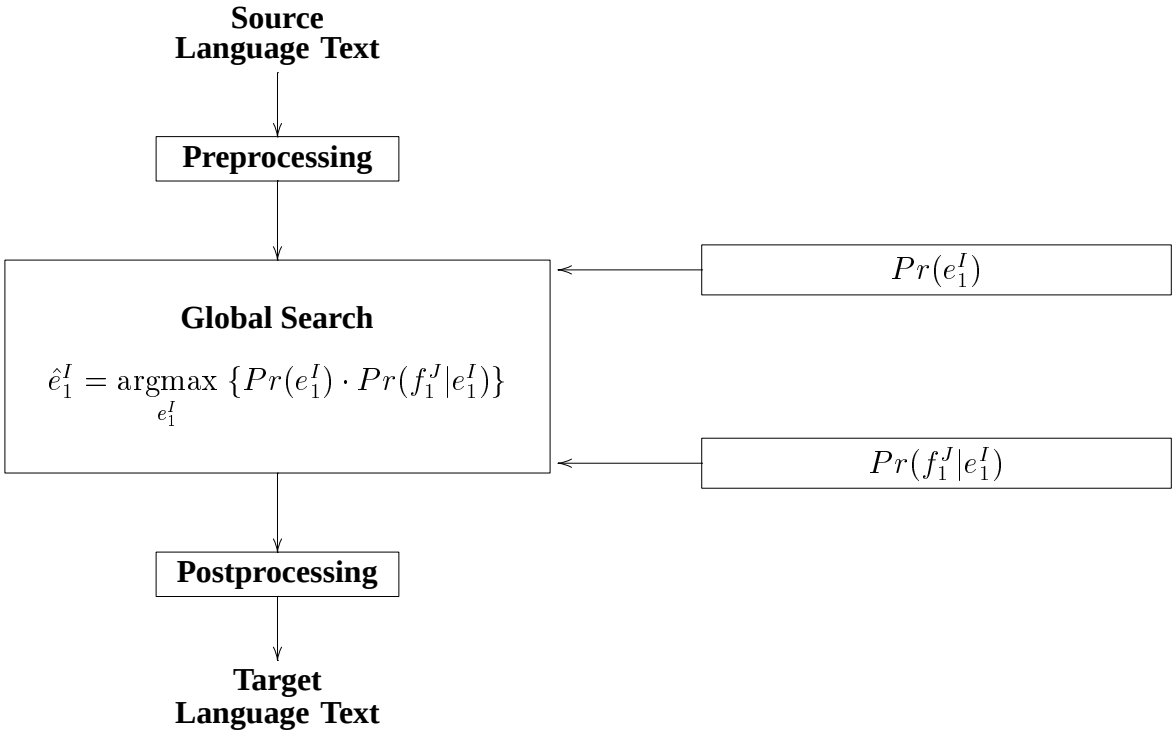


Figure 1.3: Architecture of the translation approach based on source–channel models.

We obtain the following decision rule:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \{p_{\hat{\gamma}}(e_1^I) \cdot p_{\hat{\theta}}(f_1^J|e_1^I)\} \tag{1.5}$$

Typically, state-of-the-art statistical MT systems are based on this approach. Yet, the use of this decision rule has various problems:

1. The combination of the language model $p_{\hat{\gamma}}(e_1^I)$ and the translation model $p_{\hat{\theta}}(f_1^J|e_1^I)$ as shown in Eq. 1.5 can only be shown to be optimal if the true probability distributions $p_{\hat{\gamma}}(e_1^I) = Pr(e_1^I)$ and $p_{\hat{\theta}}(f_1^J|e_1^I) = Pr(f_1^J|e_1^I)$ are used. Yet, we can only expect to obtain poor approximations of the true probability distributions. Therefore, a different combination of language model and translation model might yield better results.

2. The extension of a baseline statistical translation model by including additional dependencies is typically very complicated.

3. Often, we observe that comparable results are obtained by using the following decision rule instead of Eq. 1.5 [Och & Tillmann$^+$ 99]:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \{p_{\hat{\gamma}}(e_1^I) \cdot p_{\hat{\theta}}(e_1^I|f_1^J)\} \tag{1.6}$$

   Here, we replaced $p_{\hat{\theta}}(f_1^J|e_1^I)$ by $p_{\hat{\theta}}(e_1^I|f_1^J)$. From a theoretical framework of the source–channel approach, this approach is hard to justify. Yet, as the experimental results will show (Section 8.1.2) both decision rules obtain a comparable translation quality. Hence, we can use the decision rule that is better suited for efficient search.

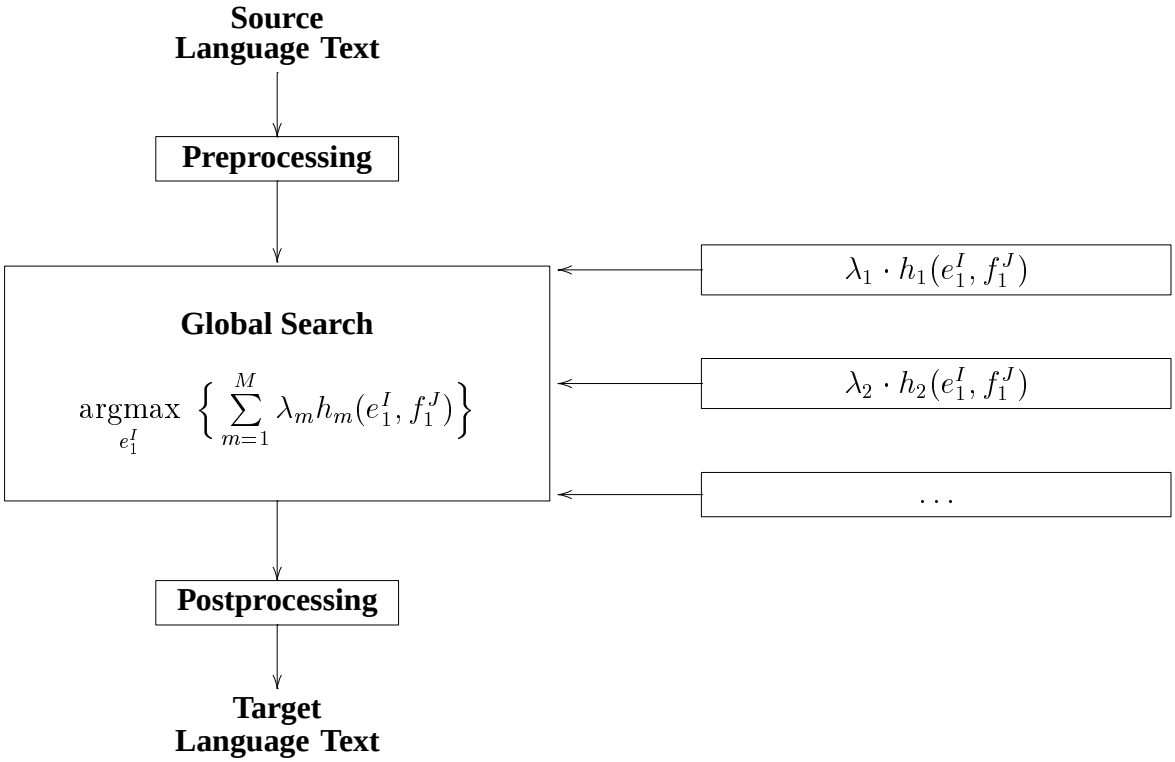


Figure 1.4: Architecture of the translation approach based on direct maximum entropy models.

1.3.2 Direct Maximum Entropy Translation Model

As alternative to the source–channel approach, we can directly model the posterior probability $Pr(e_1^I|f_1^J)$. An especially well-founded framework for doing this is maximum entropy [Berger & Della Pietra$^+$ 96]. In this framework, we have a set of M feature functions $h_m(e_1^I, f_1^J), m = 1, \ldots, M$. For each feature function, there exists a model parameter $\lambda_m, m = 1, \ldots, M$. The direct translation probability is given by:

$$Pr(e_1^I|f_1^J) = p_{\lambda_1^M}(e_1^I|f_1^J) \tag{1.7}$$

$$= \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)]}{\sum_{\tilde{e}_1^I} \exp[\sum_{m=1}^{M} \lambda_m h_m(\tilde{e}_1^I, f_1^J)]} \tag{1.8}$$

This approach has been suggested by [Papineni & Roukos$^+$ 97, Papineni & Roukos$^+$ 98] for a natural language understanding task.
We obtain the following decision rule:

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

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

Hence, the time-consuming renormalization in Eq. 1.8 is not needed in search. The overall architecture of the direct maximum entropy models is summarized in Figure 1.4.

Interestingly, this framework contains as special case the source channel approach (Eq. 1.5) if we use the following two feature functions:

$$h_1(e_1^I, f_1^J) = \log p_{\hat{\gamma}}(e_1^I) \quad (1.9)$$
$$h_2(e_1^I, f_1^J) = \log p_{\hat{\theta}}(f_1^J|e_1^I) \quad (1.10)$$

and set $\lambda_1 = \lambda_2 = 1$. Optimizing the corresponding parameters λ_1 and λ_2 of the model in Eq. 1.8 is equivalent to the optimization of model scaling factors, which is a standard approach in other areas such as speech recognition or pattern recognition.
The use of an 'inverted' translation model in the unconventional decision rule of Eq. 1.6 results if we use the feature function $\log Pr(e_1^I|f_1^J)$ instead of $\log Pr(f_1^J|e_1^I)$. In this framework, this feature can be as good as $\log Pr(f_1^J|e_1^I)$. It has to be empirically verified, which of the two features yields better results. We even can use both features $\log Pr(e_1^I|f_1^J)$ and $\log Pr(f_1^J|e_1^I)$, obtaining a more symmetric translation model.
As training criterion, we use the maximum class posterior probability criterion:

$$\hat{\lambda}_1^M = \operatorname*{argmax}_{\lambda_1^M} \left\{ \sum_{s=1}^{S} \log p_{\lambda_1^M}(\mathbf{e}_s|\mathbf{f}_s) \right\} \quad (1.11)$$

This corresponds to maximizing the equivocation or maximizing the likelihood of the direct translation model. This optimization problem has one global optimum and the optimization criterion is convex.

1.3.3 Alignment Models and Maximum Approximation

Typically, the probability $Pr(f_1^J|e_1^I)$ is decomposed via additional hidden variables. In statistical alignment models $Pr(f_1^J, a_1^J|e_1^I)$, the alignment a_1^J is introduced as a hidden variable:

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

The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$.
Typically, the search is performed using the so-called maximum approximation:

$$\begin{aligned} \hat{e}_1^I &= \operatorname*{argmax}_{e_1^I} \left\{ Pr(e_1^I) \cdot \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \right\} \\ &\approx \operatorname*{argmax}_{e_1^I} \left\{ Pr(e_1^I) \cdot \max_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \right\} \end{aligned}$$

Hence, the search space consists of the set of all possible target language sentences e_1^I and all possible alignments a_1^J.
Generalizing this approach to direct translation models, we extend the feature functions to include the dependence on the additional hidden variable. Using M feature functions of the form

$h_m(e_1^I, f_1^J, a_1^J), m = 1, \ldots, M$, we obtain the following model and decision rule:

$$\begin{aligned} Pr(e_1^I, a_1^J | f_1^J) &= \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, a_1^J)\right)}{\sum_{\tilde{e}_1^I, \tilde{a}_1^J} \exp\left(\sum_{m=1}^M \lambda_m h_m(\tilde{e}_1^I, f_1^J, \tilde{a}_1^J)\right)} \\ \hat{e}_1^I &= \underset{e_1^I}{\operatorname{argmax}} \left\{ \max_{a_1^J} \left[\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, a_1^J) \right] \right\} \end{aligned}$$

Obviously, we can perform the same step for translation models with an even richer structure of hidden variables than only the alignment a_1^J. To simplify the notation, we shall omit in the following the dependence on the hidden variables of the model.

1.3.4 Tasks in Statistical MT

Independent of the chosen starting point to statistical translation, we have to solve the following specific problems in the development of a statistical MT system:

- Modeling: introducing structures into the probabilistic dependencies to model the sentence translation probability $Pr(f_1^J|e_1^I)$ or $Pr(e_1^I|f_1^J)$.

  In the source–channel approach, we have to construct a statistical translation model:

  $$Pr(f_1^J|e_1^I) = p_\theta(f_1^J|e_1^I) \tag{1.12}$$

  This model typically contains a set of free parameters θ. To reduce the notational overhead, we shall omit the index θ if not explicitly needed.

  In a direct translation approach, we have to develop various feature functions $h_m(e_1^I, f_1^J)$, $m = 1, \ldots, M$. The free model parameters are in this case the parameters λ_1^M of Eq. 1.8.

- Training: training the free model parameters of the chosen statistical translation model using parallel and monolingual training data.

  A standard training criterion for the translation model in the source–channel approach is the maximum likelihood criterion, where we define as optimal parameter values those that maximize the likelihood on a parallel training corpus f_1^J, e_1^I:

  $$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p_\theta(f_1^J|e_1^I) \tag{1.13}$$

  Depending on the structure of the model, we might use relative frequencies or optimization algorithms such as the EM algorithm [Dempster & Laird$^+$ 77] for models with hidden variables.

  As training criterion for maximum entropy based translation models, we use the maximum class posterior probability:

  $$\hat{\lambda}_1^M = \underset{\lambda_1^M}{\operatorname{argmax}} \left\{ \sum_{s=1}^S \log p_{\lambda_1^M}(\mathbf{e}_s|\mathbf{f}_s) \right\} \tag{1.14}$$

This direct optimization of the posterior probability in Bayes decision rule is referred to as discriminative training [Ney 95] because it directly takes into account the overlap in the probability distributions. The functional form of the optimization problem is that of maximum entropy modeling, for which the GIS algorithm [Darroch & Ratcliff 72] allows an efficient optimization.

- Search: performing the argmax operation of Eq. 1.2 or Eq. 1.6 in an efficient way.

  To obtain an efficient structure of the search space, dynamic programming is often used [Bellman 57]. The actual search can be performed using A* [Nilsson 82], stack decoding [Jelinek 69], beam search [Ney & Mergel$^+$ 87] or greedy search algorithms.

- Preprocessing: finding appropriate transformation steps for both the source and the target languages to improve the translation process.

  Today's statistical translation models $p(f_1^J|e_1^I)$ are only rough approximations to the 'true' probability distributions $Pr(f_1^J|e_1^I)$. Therefore, certain natural language phenomena cannot be handled well. In preprocessing, we deal with these problems by removing these phenomena by suitable transformations. This might be easier to do instead of changing the statistical translation model.

In these tasks, linguistic knowledge is only needed in modeling and preprocessing. The other problems are mainly mathematical and computer science problems involving the development of efficient algorithms.

1.3.5 Advantages of the Statistical Approach for MT

In the following, we summarize various arguments supporting a statistical approach in MT. All these arguments cannot prove a general superiority of the statistical approach over other approaches. This can only be done by performing systematic evaluations.

- MT is a decision problem: given the source language words, we have to decide upon the target language words. Hence, it makes sense to solve it with the methods from statistical decision theory leading to the suggested statistical approach.

- The relationships between linguistic objects such as words, phrases or grammatical structures are often weak and vague. To model those dependencies, we need a formalism, such as offered by probability distributions, that is able to deal with these dependencies.

- To perform MT, we typically need to combine many knowledge sources. In statistical MT, we have a mathematically well-founded machinery to perform an optimal combination of these knowledge sources.

- In statistical MT, translation knowledge is learned automatically from example data. As a result, the development of an MT system based on statistical methods is very fast compared to the rule-based approach.

- Statistical MT is well suited for embedded applications where MT is part of a larger application. For example, in speech translation there is an additional speech recognition

engine, which introduces speech recognition errors. Statistical MT seems to be especially well suited for this application as it has a natural robustness. Another example is interactive MT (Chapter 10).

- The 'correct' representation of syntactic, semantic and pragmatic relationships is not known. Hence, the formalism should as much as possible not rely on constraints induced by such hypothetical levels of description. Instead, in the statistical approach, the modeling assumptions are empirically verified on training data.

- Statistical MT has shown to obtain very good results. The system developed in this thesis has significantly outperformed other classical approaches in a large-scale evaluation (Section 8.1.3).

1.4 Related Work

Statistical MT is a new research area. Therefore, the amount of research performed in this area is limited. So far, only few research groups are active in this field.
Statistical MT has been introduced by the seminal work of a research group at IBM [Brown & Cocke$^+$ 90]. They introduced the concept of alignment models to describe the dependencies between source and target language words [Brown & Della Pietra$^+$ 93b, Berger & Brown$^+$ 94] and developed a search algorithm for these models based on the paradigm of stack decoding [Berger & Brown$^+$ 96].
Unfortunately, even for simple translation models, the search problem in statistical MT is NP complete [Knight 99a]. Various research groups tried to extend the IBM work to develop more efficient search algorithms by using suitable simplifications and applying better optimization methods. Beam search and dynamic programming based monotone search with a time complexity linear to the input length has been suggested in [Tillmann & Vogel$^+$ 97a, Tillmann & Vogel$^+$ 97b]. In [Tillmann & Ney 00, Tillmann 01], this was extended to handle also word reordering. [Nießen & Vogel$^+$ 98] suggested a simplified recombination rule in dynamic programming search to obtain a polynomial time search algorithm even in the case of general reordering.
Various researchers suggested greedy or perturbation search approaches [Berger & Brown$^+$ 94, Wang 98, Germann & Jahr$^+$ 01]. Here, some initially chosen translation is iteratively improved by performing in a greedy manner small perturbations. A similar iterative search approach is used by [García-Varea & Casacuberta$^+$ 98, García-Varea & Casacuberta 01], which iteratively improves an initial translation using a dynamic programming based search architecture.
A recent innovative approach has been integer programming as framework for an optimal search algorithm [Germann & Jahr$^+$ 01] for Model 4. Here, the search problem is reformulated as an integer programming optimization problem and a standard toolkit is used to solve it. Yet, this approach is only applicable to very short sentences.
[Wu 96] suggested an approach where the possible word orders were restricted using so-called stochastic inverse transduction grammars yielding a polynomial time search algorithm.
A major disadvantage of the baseline IBM alignment models is that they do not take word context into account. A partial solution to this problem, which works for frequent words was introduced by [Berger & Della Pietra$^+$ 96] and continued by [García-Varea & Och$^+$ 01], which suggested a maximum entropy based context-dependent lexicon model. [Wang & Waibel 98]

introduced a phrase-based translation model that is an extension of the original IBM translation models.
A different approach based on maximum entropy has been suggested by [Foster 00a, Foster 00b]. In this approach, language and translation model features are learned in combination. Here, the translation model is not structured using hidden alignments. The goal of this approach is not to perform fully automatic MT, but to predict very efficiently the most probable extension of a translation prefix.
Various researchers suggested to apply finite state technology to MT. The so-called *head transducer* approach was introduced by [Alshawi & Bangalore$^+$ 98, Alshawi & Bangalore$^+$ 00]. This approach can be seen as a statistical bilingual lexicalized grammar, based on finite state transducers. Another statistical approach based on finite state transducers is the Onward Subsequential Transducer Inference Algorithm (OSTIA) [Castellanos & Galiano$^+$ 94] and its extension the so-called OMEGA algorithm [Vilar & Vidal$^+$ 96]. The basic approach is to store the translation examples in a finite state transducer, which corresponds to a prefix tree representation of the source language and to perform an iterative state merging. The finite state approaches are especially suited to speech translation as a straightforward combination of recognition and translation is possible.
The area of statistical natural language understanding (NLU) is also related to statistical MT. The difference to the problem of machine translation is that the target language language is not a natural but a formal language. Various approaches have been developed and evaluated in the context of the ATIS project [Price 90]. The work on the IBM translation models has been adapted in the NLU field by [Epstein & Papineni$^+$ 96, Della Pietra & Epstein$^+$ 97]. Hidden Markov models have been used in [Miller & Bobrow$^+$ 94, Haas & Hornegger$^+$ 97]. Whole-sentence direct maximum entropy translation models for natural language understanding have been suggested by [Papineni & Roukos$^+$ 97, Papineni & Roukos$^+$ 98]. Phrase-based translation models have been suggested by [Macherey & Och$^+$ 01].
An important component of almost all statistical MT systems is a word alignment model. For this problem, various statistical or statistically motivated alignment models have been suggested [Dagan & Church$^+$ 93, Vogel & Ney$^+$ 96, Smadja & McKeown$^+$ 96, Ker & Chang 97, Melamed 00, Huang & Choi 00].
In automatic speech recognition [Rabiner & Juang 93, Jelinek 97] and pattern recognition [Duda & Hart 73, Niemann 90, Fukunaga 90, Duda & Hart$^+$ 00], many of the statistical methods used in this thesis have been applied for many years.

Chapter 2

Scientific Goals

The aim of this work is to extend the state-of-the-art in MT by developing new statistical translation models and efficient training and search algorithms. In addition, new innovative applications for statistical MT are developed. In particular, the following scientific goals are pursued:

- So far, the literature rarely contains descriptions of complete statistical MT systems. Often, only certain components—for example, only alignment models—are analyzed. In this thesis, we describe in detail the development of a statistical MT system in all its aspects: data collection, preprocessing, modeling, training and search.

- There is a vast literature on the topic of computing a word alignment from sentence aligned bilingual corpora and many different systems have been proposed to solve this problem. Yet, the literature does not include a systematic comparison of different alignment methods. Therefore, it remains unclear, which methods should be used to produce good word alignments. In this thesis, we provide a quantitative comparison of various word alignment models. In addition, new models and new efficient training algorithms are developed that yield significantly better word alignment quality.

- A general deficiency of the baseline alignment models is that they are only able to model correspondences between single words. We develop a new phrase-based[1] statistical MT system — the alignment template approach — that allows for general many-to-many relations between words. In various evaluations, it has shown to outperform other classical or statistical translation approaches.

- So far, the source–channel approach is the standard approach to develop statistical MT systems. Yet, as has been shown in pattern recognition and speech recognition, directly modeling the posterior probability typically achieves better results. We suggest using the framework of maximum entropy models. This allows not only a better exploitation of conventional translation models, but also an extension of statistical MT systems easily by adding new feature functions.

- A standard method in language modeling are word classes to obtain better generalizing language models [Brown & Della Pietra$^+$ 92, Kneser & Ney 93]. We extend these methods for using them in the context of statistical MT allowing for more general phrase-based translation lexicons.

[1] In this thesis, the term *phrase* simply refers to a consecutive sequence of words.

- In the literature, various search algorithms have been proposed to deal with the search problem in statistical MT. In left-to-right search algorithms, the hypotheses are formed with increasing length. Typically, the scoring of the search hypotheses takes into account only the current translation prefix probability. We propose to improve search efficiency by including an admissible heuristic function.

- The statistical approach to MT not only results to be a very competitive new method for performing MT, but also allows opens up interesting possibilities for new applications. One of these is multi-source translation, which is the use of multiple source languages to produce one target language translation. This has various advantages for disambiguation and word reordering.

- Current MT technology is not able to produce high quality MT output. Hence, post-editing of the MT output is often necessary. We suggest an interactive MT environment that supports the human translator by interactively reacting to user input providing an *auto-typing* facility that suggest to the human translator an extension of the sentence that he is typing.

Chapter 3

System Overview

One of the major advantages of statistical MT is that it can be learned automatically. This is also the main reason why the process of developing a statistical MT system differs significantly from a classical rule-based system. In this chapter, we provide an overview on the development of a statistical MT system.

3.1 Development Cycle of Statistical MT Systems

Figure 3.1 presents the development cycle of a statistical MT system. A major difference to the development cycle of classical MT systems is that an *evolutionary rapid prototyping approach* [Connell & Shafer 94] can be pursued. An initial baseline with a reasonable quality can be bootstrapped very quickly if sufficient training data are available. Afterwards, an iterative improvement process starts.
The first step is the collection of training data. Here, we need to obtain parallel texts, perform sentence alignment and extract the suited translation pairs. In the second step, we perform an automatic training of the MT system. The output of this step is an operative MT system. Typically, this step is quite fast and needs no human supervision.
Afterwards, the MT system is tested and an error analysis is performed. Taking into account the architecture of a statistical MT system (Figure 1.3), we can distinguish different error types: *search errors*, *modeling errors*, *training errors*, *training corpus errors* and *preprocessing errors*. Depending on the result of this error analysis, various modifications are performed:

- **Better models**: Here, the goal is to develop models, which better capture the properties of natural language and whose free parameters can be estimated reliably from training data.

- **Better training**: The used training algorithms are often based on maximum likelihood, which is prone to overfitting. In addition, the used baseline training algorithm might find only locally optimal parameters (Eq. 1.13), which is for example a problem in the use of the EM algorithm for the statistical alignment models of Chapter 4.

  For certain model parameters, different parameter values have to be tested with respect to development corpus error rate. This is typically called *parameter tuning*. To do this efficiently, an automatic evaluation procedure is important.

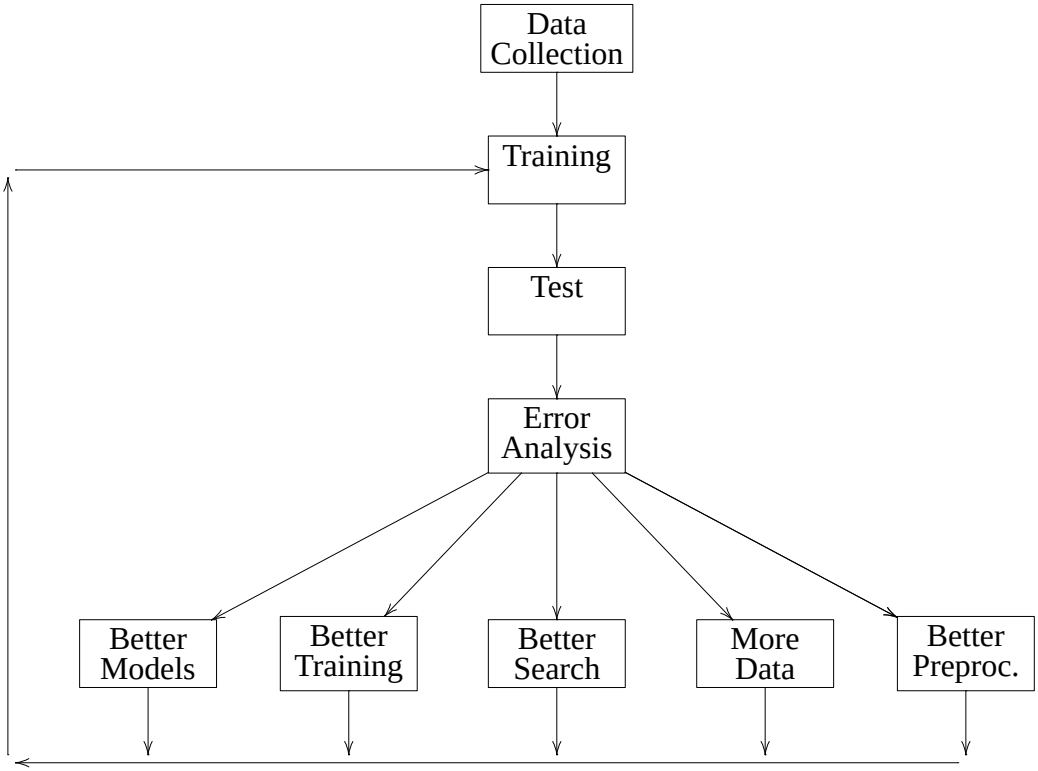


Figure 3.1: Development cycle of a statistical MT system.

- **Better search**: A search error occurs if the search algorithm produces a translation, which is different from the optimizing translation $\hat{e}_1^I$ defined in Eq. 1.2 or Eq. 1.6. The search problem in statistical MT is typically NP-complete. Therefore, suitable approximations in search need to be performed to obtain a good trade-off between translation quality and efficiency.

- **More training data**: Typically, translation quality improves if the training corpus size increases. The learning curve of an MT system shows how much training data are needed to obtain a certain performance. An additional error source are wrong or too free translations in the training data. To avoid these errors, manual or automatic filtering or correction of these translation examples needs to be done.

- **Better preprocessing**: Various natural language phenomena are notoriously difficult to handle for state-of-the-art statistical approaches. One method for dealing with this problem is to preprocess the text such that the text is better suited for the statistical translation models. Here, rule-based MT technology can be used. Typically, simple text transformations are performed that yield a normalized source and target language.

An important property of the development cycle of statistical MT systems is that we can have typical turnaround times of a few hours or days. Hence, the development cycle is gone through very often. This allows quickly improving the MT system. In addition, the error analysis always depends on the final MT performance. Hence, the decision on system modifications can be based directly on the ultimate goal of high MT quality.

Table 3.1: Corpus statistics of EU bulletin task.

Language	Sentences	Words	Voc.
French	117K	2.32M	50462
Spanish	120K	2.32M	50949
Portuguese	120K	2.30M	50216
Italian	120K	2.21M	54986
Swedish	125K	2.02M	72517
Danish	131K	2.21M	70713
Dutch	121K	2.30M	58550
German	139K	2.23M	73506
Greek	131K	2.28M	68811
Finnish	120K	1.61M	106159
English		˜2.1M	˜45K

3.2 Training Corpus Collection

To develop a statistical MT system, we need to have a training corpus. The training corpus should be as *large* as possible, should be from the *same domain* for which the MT system is used and should be mostly *literally translated*.
The training corpus collection results to be a very expensive process if performed manually. In particular, the collection of speech translation corpora is expensive as the speech data need to be collected in realistic scenarios, manually transcribed and translated. The VERBMOBIL and the EUTRANS speech corpora (Section 8.1.1 and Appendix A.2) have been collected in this way.

Automatic Training Corpus Collection from the Web

An efficient and cheap approach to collect parallel text is mining the Internet for parallel text. [Resnik 99] suggested such a method for automatically finding a French–English parallel corpus. In the following, we describe a method for automatically collecting a multilingual corpus with many different languages.
The data source is the *Bulletin of the European Union*, which exists in the 11 official languages of the European Union. This corpus is publically available on the Internet.[1] We performed the following steps to obtain a multilingual corpus:

1. We downloaded this corpus for all eleven languages in HTML format.

2. We performed an alignment on text level by file name matching.

3. We extracted the raw text from this corpus by extracting all text segments within HTML tags. Often, these segments correspond to paragraphs. Hence, we obtained a sequence of text segments for each text in each language.

4. We performed a segment alignment between two languages by using a dynamic programming based algorithm, which tries to map segments of equal length [Gale & Church 93].

[1] Bulletin of the European Union: `http://europa.eu.int/abc/doc/off/bull/en/welcome.htm`.

Table 3.2: Test corpus statistics of EU bulletin task.

Sentences	1 302
English words	15 048
Trigram perplexity	179
Bigram perplexity	286

We performed this segment alignment for ten language pairs. Hence, we obtained ten bilingual corpora aligned on the paragraph level.

5. We performed a sentence alignment using similar heuristics as in the paragraph alignment. Hence, we obtained ten bilingual corpora aligned on sentence level.

6. From the resulting bilingual corpora, we filtered all sentences that seem to have wrong alignments such as alignments of very long sentences with very short sentences or alignments, which have a very low probability according to the Hidden Markov alignment model (Section 4.2.2).

7. For all languages, we performed the same preprocessing. This includes tokenization, mapping of words at the beginning of a sentence to their true case and categorization of numbers.

Table 3.1 shows the corpus statistics of the collected training corpora. Due to the filtering of poor alignments, the numbers for English differ with respect to the considered language pair up to 10 percent. The vocabulary sizes differ considerably between the different languages. Languages such as Finnish with a very rich morphology have a very large vocabulary of full-form words and languages like English have a very small vocabulary. We have extracted one test corpus by finding sentences that are available in all corpora. These sentences were removed from all training corpora. Table 3.2 shows the test corpus statistics.

Conventional Dictionary

As additional knowledge source, we use a conventional bilingual dictionary if available. This dictionary can help to bootstrap the training of the statistical alignment models (Section 4.3.4) and in addition helps to cover vocabulary that does not occur in the training corpus.

Ideally, this dictionary would include those entries that are relevant for the specific domain. Yet, typically the available dictionary is not domain-specific, which might lead to the problem that out-of-domain lexicon entries hide the in-domain lexicon entries learned from the training corpus. Hence, we extract those lexicon entries from the general purpose lexicon that are relevant in the domain. We do this by extracting all lexicon entries that really co-occur in the bilingual training corpus. To do this efficiently, we use the data structure of suffix arrays [Manber & Myers 90], which allow an efficient search of arbitrary length sub-strings in the training corpus. These co-occurring entries are then given a larger weight in training.

3.3 Preprocessing

Motivation

In statistical MT, we will always face a sparse data problem. Many words and many syntactical constructions are seen only once in the training data. For these words and constructions, a robust training of the corresponding model parameters is often not possible. In addition, we will have to perform simplistic model assumptions to be practically able to train the model parameters and to perform efficient search.
Therefore, certain natural language phenomena cannot be handled adequately by these models. For example, most existing language and translation models consider only local context. This is sufficient for many sentences. Yet, sentences involving nonlocality might be translated wrongly because of the model restriction. A solution is the use of preprocessing. For example, we could perform reordering on the source sentence or the target sentence, which transforms the nonlocal phenomenon in a local phenomenon. Using the terminology of Section 1.2, we obtain a transfer-based approach if we use refined preprocessing.
Often, for an existing statistical MT system, preprocessing is a method that allows obtaining large improvements with relatively small effort.

Basic concepts

Formally, preprocessing and postprocessing can be described by functions Π_e and Π_f, which transform the source and target language sentences:

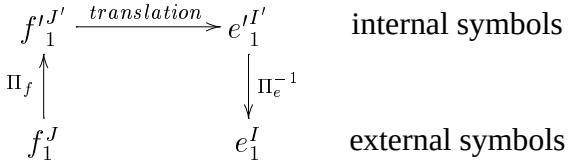
$$\begin{array}{ccc} f_1'^{J'} & \overset{training}{\longleftrightarrow} & e_1'^{I'} \\ {\scriptstyle \Pi_f}\uparrow & & {\scriptstyle \Pi_e}\uparrow \\ f_1^J & & e_1^I \end{array} \qquad \begin{array}{l} \text{internal symbols} \\ \\ \text{external symbols} \end{array}$$

To train the translation model, we use instead of the original training corpus f_1^J, e_1^I the transformed training corpus $\Pi_e(e_1^I) = e'^{I'}_1, \Pi_f(f_1^J) = f'^{J'}_1$. To train the language model, we use $\Pi_e(e_1^I) = e'^{I'}_1$.
From the statistical viewpoint, we expect the preprocessing algorithms to increase the likelihood of the training corpus or equivalently reduce training corpus perplexity. Hence, as criterion of the preprocessing quality, we use:

$$\Delta\text{PP}(\Pi_e, \Pi_f) = Pr(f_1^J|e_1^I)^{-1/J} - Pr(f'^{J'}_1|e'^{I'}_1)^{-1/J'} \tag{3.1}$$

Applied to the whole training corpus, this criterion can be used to assess the overall effect of a certain preprocessing. Applied to every single sentence of the training corpus, this criterion reveals the sentences where preprocessing makes sentences 'easier' but it also reveals the sentences where preprocessing makes sentences harder for the used statistical model. Hence, this criterion can be used to perform a detailed analysis of the effect of certain preprocessing steps.
To use the preprocessing in actual translations, we need Π_e^{-1} to transform the internal target language representation $e'^{I'}_1$ to the external representation e_1^I. We obtain the following translation

process:



The transformation Π_e^{-1} is not necessarily the exact inverse function of Π_e, but the sentences e_1^I and $\Pi_e^{-1}(\Pi_e(e_1^I))$ should have identical meaning. Hence, for training, we need Π_f and Π_e and for the final MT system, we need Π_f and Π_e^{-1}. The transformation Π_f^{-1} is not needed.
In the following, to keep the notation simple, we do not make an explicit distinction between preprocessed and not preprocessed sentences. We always use the symbols f_1^J and e_1^I for source and target language sentences.

Used preprocessing environment

In the following, we shortly review the developed preprocessing environment. The user of this preprocessing environment should specify for each operation also the corresponding inverse operation. Therefore, the same preprocessing environment can be used for both translation directions.
The used preprocessing environment consists of the following components:

- *Tokenization*: Here, the sequence of input characters is transformed into a sequence of words. In this step, for example the punctuation marks are separated from words and sentence boundaries are detected. To do this in a generic language independent way, we specify those character sequences that form words, numbers or abbreviations.

- *True case mapping*: In this step, the uppercased words at the beginning of a sentence are mapped to lower case characters if the lowercase version occurs more often than the uppercase version of the word.

- *Phrase maps*: These are transformations that map sequences of words to other sequences. This can be used to reformulate certain problematic expressions and to normalize expressions. Phrase maps can also be used for categorization purposes, for example, to replace named entities by a generic symbol for that named entity. The transformations can be specified using regular expressions.

3.4 Language Modeling

Another important element of a statistical MT system is the language model. It describes the well-formedness of the produced target language sentence. We use left-to-right language models, as these can be easily integrated in the standard left-to-right search architecture.
Typically, word-based trigram language models are used [Ney & Generet$^+$ 95]:

$$Pr(e_1^I) = \prod_{i=1}^{I+1} p(e_i|e_{i-1}, e_{i-2}) \tag{3.2}$$

Here, we assume that $e_0 = e_{-1} = e_{I+1} = \$$ is a special sentence boundary symbol. Ideally, we would like to use a very large history length n as there might exist long-range dependencies which have to be taken into account. Yet, long n-grams are seen rarely and are therefore rarely used on unseen data. Therefore, we use in addition class-based n-gram models [Brown & Della Pietra$^+$ 92] with a longer history. Here, the words e are categorized into classes $C(e)$:

$$Pr(e_1^I) = \prod_{i=1}^{I+1} p(e_i|C(e_i)) \cdot p(C(e_i)|C(e_{i-h+1}), \ldots, C(e_{i-1})) \tag{3.3}$$

Typically, we use a history length of $h = 4$ words and a class set which distinguishes 300 different classes. The word classes are automatically learned [Och 95]. In Section 8.1.2 (Table 8.18), we analyze the effect of the language model history length on translation quality.
In this thesis, we do not use grammar-based language models, which try to enforce a more grammatical target language sentence. So far, it seems that the effort needed in implementing those models and the introduced computation effort do not sufficiently pay off [Sawaf & Schütz$^+$ 00].

3.5 MT Evaluation

So far, the MT community has no generally accepted criteria for measuring the quality of MT output. Yet, MT evaluation is very important in the development process of a statistical MT system. Ideally, we would like a situation as in automatic speech recognition research, where a generally accepted evaluation criterion—word error rate (WER)—exists.
In principle, MT quality can be measured using many nonorthogonal dimensions [Hovy 99]. Ideally, we would like to have a one-dimensional evaluation criterion as in speech recognition, which makes comparing different MT systems easy. In addition, we would like to use an evaluation criterion that is cheap in its application. If the development and improvement cycle of statistical MT systems takes only a few hours or a few days, then a slow evaluation cycle would be the bottleneck for improving system quality. Hence, performing a time-consuming subjective evaluation of MT quality is not desirable.
A general problem of subjective MT evaluation is that the comparability of different results is hard to guarantee if the evaluation is not performed by the same group of humans in the same moment. One method for dealing with this problem is the use of common evaluation tools and databases [Jones & Rusk 00, Nießen & Och$^+$ 00, Vogel & Nießen$^+$ 00].
In this thesis, we distinguish objective and subjective error criteria. The objective error criteria compare the similarity of the produced translation with a set of reference translations. On the other hand, the subjective criteria depend on a human quality judgment.
We use the following objective error criteria:

- WER (word error rate):
  The WER is computed as the minimum number of substitution, insertion and deletion operations that have to be performed to convert the generated sentence into the target sentence. This performance criterion is widely used in speech recognition. This minimum is computed using a dynamic programming algorithm and is typically referred to as *edit* or *Levenshtein* distance.

- PER (position-independent word error rate):
  A shortcoming of the WER is that it requires a perfect word order. This is particularly a problem for the VERBMOBIL task, where the word order of the German–English sentence pair can be quite different. As a result, the word order of the automatically generated target sentence can be different from that of the reference sentence, but nevertheless acceptable so that the WER measure alone could be misleading. To overcome this problem, we introduce as additional measure the position-independent word error rate (PER). This measure compares the words in the two sentences *without* considering the word order. Words that have no matching counterparts are counted as substitution errors. Depending on whether the translated sentence is longer or shorter than the target translation, the remaining words result in either insertion or deletion errors in addition to substitution errors. The PER is guaranteed to be less than or equal to the WER.

- mWER (multi-reference word error rate):
  For each test sentence, there is not only used a single reference translation, as for the WER, but a whole set of reference translations. For each translated sentence, the edit distance (number of substitutions, deletions and insertions) to the most similar sentence is calculated [Nießen & Och$^+$ 00].

- BLEU score:
  This score measures the precision of unigrams, bigrams, trigrams and fourgrams with respect to a whole set of reference translations with a penalty for too short sentences [Papineni & Roukos$^+$ 01]. Unlike all other evaluation criteria used here, BLEU measures accuracy, i.e. the opposite of error rate. Hence, large BLEU scores are better.

We use the following subjective error criteria:

- SSER (subjective sentence error rate):
  For a more detailed analysis, subjective judgments by test persons are necessary. Each translated sentence is judged by a human examiner according to an error scale from 0.0 to 1.0 [Nießen & Och$^+$ 00]. A score of 0.0 means that the translation is semantically and syntactically correct, a score of 0.5 means that a sentence is semantically correct but syntactically wrong and a score of 1.0 means that the sentence is semantically wrong.

- IER (information item error rate):
  The test sentences are segmented into information items. For each of them, the human examiner decides if the candidate translation includes this information item. The translation of this information item is judged as correct, if the intended information is conveyed and there are no syntactic errors [Nießen & Och$^+$ 00].

Interestingly, the automatic evaluation criteria WER, PER, mWER and BLEU often, but not always, correlate with a subjectively evaluated translation quality. The error criteria mWER and BLEU seem to correlate especially well.
Typically, we use the objective criteria in the development cycle of an MT system. Only from time to time, we perform an expensive subjective evaluation to check that the automatic evaluation criteria do not produce misleading results.

Chapter 4

Statistical Alignment Models

In this chapter, we present and compare various methods for computing word alignments using statistical or heuristic models. We discuss the five alignment models presented in [Brown & Della Pietra$^+$ 93b], the Hidden Markov alignment model, smoothing techniques and refinements. These statistical models are compared with two heuristic models based on the Dice coefficient. We present different methods for combining directed word alignments to perform a symmetrization of directed statistical alignment models. As evaluation criterion, we use the quality of the resulting Viterbi alignment compared to a manually produced reference alignment. We evaluate the models on the German–English VERBMOBIL task and the French–English HANSARDS task. We perform a detailed analysis of various design decisions of our statistical alignment models and evaluate these on training corpora of various sizes. An important result is that refined alignment models with a first-order dependence and a fertility model yield significantly better results than simple heuristic models. In Appendix C, we present an efficient training algorithm of the presented alignment models.

4.1 Introduction

In the following, we address the problem of finding the word alignment of a bilingual sentence-aligned corpus by using language independent statistical methods. There is a vast literature on this topic and many different MT systems have been suggested to solve this problem. Our work follows and extends the methods introduced by [Brown & Della Pietra$^+$ 93b] by using refined statistical models for the translation process. The basic idea of this approach is to develop a model of the translation process with the word alignment as a hidden variable of this process, to apply statistical estimation theory to compute the 'optimal' model parameters and to perform alignment search to compute the best word alignment.

So far, refined statistical alignment models have been rarely used. One reason for this is the high complexity of these models, which makes them difficult to understand, implement and tune. Instead, heuristic models are usually used. Here, the word alignments are computed by analyzing some association score metric of a link between a source language word and a target language word. These models are relatively easy to implement.

Here, we focus on consistent statistical alignment models suggested in the literature, but we also describe a heuristic association metric (Dice coefficient). By providing a detailed description

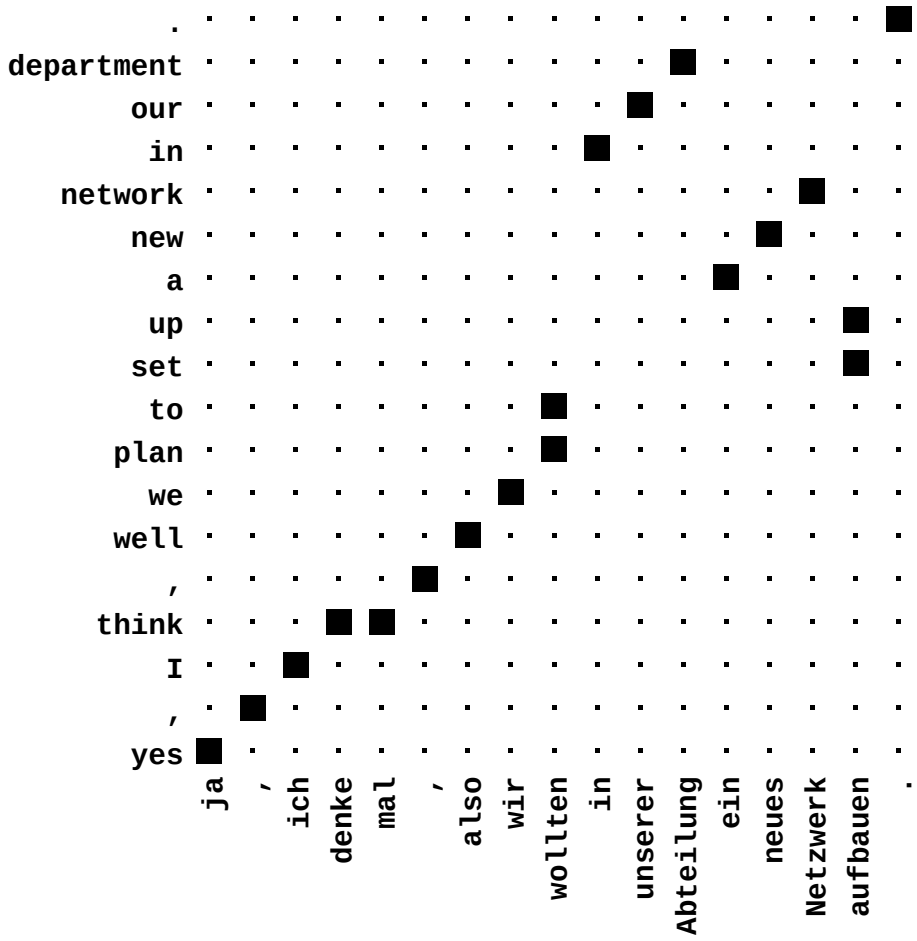


Figure 4.1: Example of a word alignment (VERBMOBIL task).

and a systematic evaluation of these alignment models, we give the reader various criteria to decide which model to use for a given task.
We propose to measure the quality of an alignment model by comparing the quality of the most probable alignment — the Viterbi alignment — with a manually produced reference alignment. This has the advantage that an automatic evaluation can be performed. In addition, we shall show that this quality measure is a precise and reliable evaluation criterion, which is well suited to guide designing and training statistical alignment models.
The software used to train the statistical alignment models is publicly available [Och 00a].

4.1.1 Problem Definition

We follow [Brown & Della Pietra$^+$ 93b] in defining the alignment as an object for indicating the corresponding words in a parallel text. Figure 4.1 and Figure 4.2 give examples of such relationships. Often, humans cannot determine easily which target words correspond to which source words. In particular, the alignment of words within idiomatic expressions, free translations, and missing function words is problematic. The problem is that the notion of 'correspondence' between words is subjective. This results to be important for the evaluation of the word alignment quality (Section 4.5).

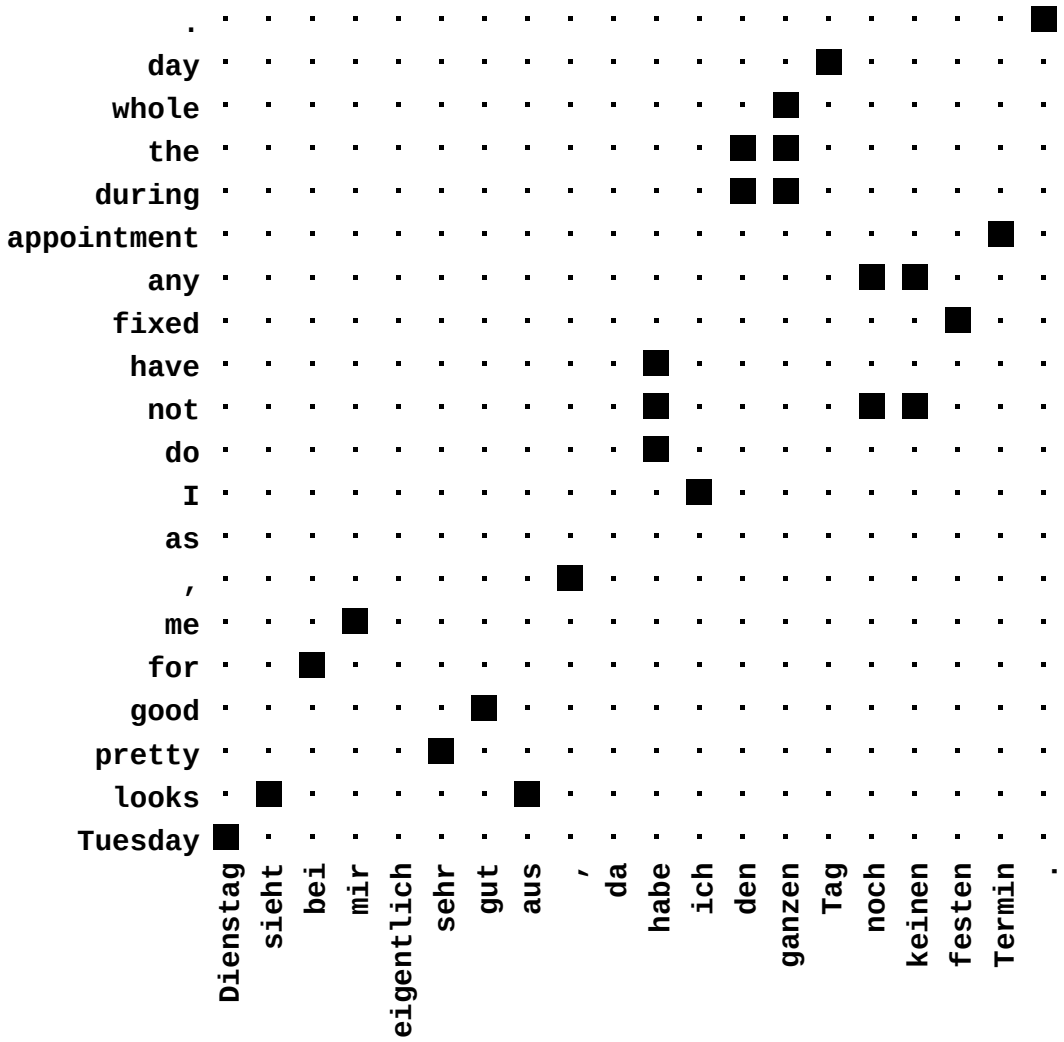


Figure 4.2: Example of a word alignment (VERBMOBIL task).

The alignment between two sentences can be quite complicated. Often, an alignment includes effects such as changes in word order, omissions, insertions, and word-to-phrase alignments. Therefore, we need a very general alignment representation. Formally, we use the following definition for alignment. We are given a source ('French') sentence $f_1^J = f_1, \ldots, f_j, \ldots, f_J$ and a target language ('English') sentence $e_1^I = e_1, \ldots, e_i, \ldots, e_I$, which have to be aligned. We define an alignment between both sentences as a subset of the Cartesian product of the word positions, i.e. an alignment $\mathcal{A}$ is defined as:

$$\mathcal{A} \subseteq \{(j, i) : j = 1, \ldots, J; i = 1, \ldots, I\} \tag{4.1}$$

Modeling the alignment as an arbitrary relation between source and target language positions is quite general. However, the development of alignment models that are able to deal with this general representation is hard. Typically, the alignment models presented in the literature have additional constraints.
Typically, the model is restricted in a way such that each source word is assigned to *exactly one* target word. These alignment models are similar to the concept of Hidden Markov models

(HMM) in speech recognition. The alignment mapping consists of associations $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. The alignment $a_1^J = a_1, \ldots, a_j, \ldots, a_J$ may contain alignments $a_j = 0$ with the 'empty' word e_0 to account for source words that are not aligned to any target word. In such a way, the alignment is not a relation between source and target language positions, but only a mapping from source to target language positions.
In [Melamed 00], a further simplification is performed that enforces a one-to-one alignment. This means that the alignment mapping a_1^J must be invertible for all word positions $a_j > 0$. Many translation phenomena cannot be handled using these restricted alignment representations. Especially, in terms of precision and recall, those methods are in principle not able to achieve a 100% recall. The problem can be reduced by corpus preprocessing steps, which perform grouping and splitting of words.
Some papers report improvements in the alignment quality of statistical methods by using linguistic knowledge [Ker & Chang 97, Huang & Choi 00]. Here, the linguistic knowledge is mainly used to remove wrong alignments. Here, we avoid making explicit assumptions concerning the used language. In such a way, we expect the approach to be applicable to almost every language pair. The only assumptions that we make are that the parallel text is segmented into aligned sentences and that the sentences are segmented into words. Obviously, there are additional implicit assumptions in the models that need to hold to obtain a good alignment quality. For example, in languages with a very rich morphology such as Finnish, a trivial segmentation produces a high number of words that occur only once and every learning method suffers from a significant data sparseness problem.

4.1.2 Applications

There are numerous applications for word alignments in natural language processing. These applications crucially depend on the quality of the word alignment [Och & Ney 00a, Yarowsky & Wicentowski 00].
An obvious application for word alignment methods is the automatic extraction of bilingual lexicons and terminology from corpora [Smadja & McKeown$^+$ 96, Melamed 00].
Statistical alignment models are often the basis of single-word based translation systems [Berger & Brown$^+$ 94, Wu 96, Wang & Waibel 97, Nießen & Vogel$^+$ 98, García-Varea & Casacuberta$^+$ 98, Och & Ueffing$^+$ 01, Germann & Jahr$^+$ 01]. In addition, these models are the starting point for refined phrase-based statistical [Och & Weber 98, Och & Tillmann$^+$ 99] or example-based translation systems [Brown 97]. Here, the quality of the MT output directly depends on the quality of the initial word alignment [Och & Ney 00a].
Another application of word alignments is in the field of word sense disambiguation [Diab 00]. In [Yarowsky & Ngai$^+$ 01], the word alignment is used to transfer text analysis tools such as morphological analyzers or part-of-speech tagger from a language such as English for which many tools exist already to languages where such resources are scarce.

4.1.3 Overview

In Section 4.2, we review various statistical alignment models and heuristic models. We present a new statistical alignment model, which is a log-linear combination of the best models of

[Vogel & Ney$^+$ 96] and [Brown & Della Pietra$^+$ 93b]. In Section 4.3, we describe the training of the alignment models and present a new training schedule, which yields significantly better results. In addition, we describe the handling of overfitting and deficient models. In Section 4.4, we present some heuristic methods for improving alignment quality by performing a symmetrization of word alignments. In Section 4.5, we describe an evaluation methodology for word alignment methods dealing with the ambiguities associated with the word alignment annotation based on generalized precision and recall measures. In Section 4.6, we present a systematic comparison of the various statistical alignment models with regard to alignment quality and translation quality. We assess the effect of training corpora of various sizes and the use of a conventional bilingual dictionary. In the literature, it is often claimed that the refined alignment models of [Brown & Della Pietra$^+$ 93b] are not suitable for small corpora due to data sparseness problems. We show that this is not the case if these models are parametrized suitably. Appendix C describes some methods for efficient training.

4.2 Review of Alignment Models

4.2.1 General Approaches

We distinguish between two general approaches to compute word alignments: statistical alignment models and heuristic models. In the following, we describe both types of models and compare them from a theoretical viewpoint.
The notational convention will be as follows. We use the symbol $Pr(.)$ to denote general probability distributions with (almost) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(.)$.

Statistical alignment models

In statistical MT, we try to model the translation probability $Pr(f_1^J|e_1^I)$, which describes the relationship between a source language sentence f_1^J and a target language sentence e_1^I. In (statistical) alignment models $Pr(f_1^J, a_1^J|e_1^I)$, a 'hidden' alignment $\mathbf{a} = a_1^J$ is introduced, which describes a mapping from a source position j to a target position a_j. The relationship between the translation model and the alignment model is given by:

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \tag{4.2}$$

The alignment a_1^J may contain alignments $a_j = 0$ with the 'empty' word e_0 to account for source words that are not aligned to any target word.
In general, the statistical model depends on a set of unknown parameters θ that is learned from training data. To express the dependence of the model on the parameter set, we use the following notation:

$$Pr(f_1^J, a_1^J|e_1^I) = p_\theta(f_1^J, a_1^J|e_1^I) \tag{4.3}$$

The art of statistical modeling is to develop specific statistical models that capture the relevant properties of the considered problem domain. Hence, the statistical alignment model has to describe the relationship between a source language string and a target language string adequately.

To train the unknown parameters θ, we are given a parallel training corpus consisting of S sentence pairs $\{(\mathbf{f}_s, \mathbf{e}_s) : s = 1, \ldots, S\}$. For each sentence pair $(\mathbf{f}_s, \mathbf{e}_s)$, the alignment variable is denoted by $\mathbf{a} = a_1^J$. The unknown parameters θ are determined by maximizing the likelihood on the parallel training corpus:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left\{ \prod_{s=1}^{S} \left[\sum_{\mathbf{a}} p_\theta(\mathbf{f}_s, \mathbf{a}|\mathbf{e}_s) \right] \right\} \tag{4.4}$$

Typically, for the kinds of models we describe here the EM algorithm [Dempster & Laird$^+$ 77] or some approximate EM algorithm is used to perform this maximization. However, to avoid a common misunderstanding, note that the use of the EM algorithm is not essential for the statistical approach, but only a useful tool for solving this parameter estimation problem.
Although for a given sentence pair there are many alignments, we can always find a best alignment:

$$\hat{a}_1^J = \underset{a_1^J}{\operatorname{argmax}} \; p_{\hat{\theta}}(f_1^J, a_1^J|e_1^I) \tag{4.5}$$

The alignment $\hat{a}_1^J$ is also called the *Viterbi alignment* of the sentence pair (f_1^J, e_1^I). For the sake of simplicity, we shall drop the index θ if not explicitly needed.
Later on, we shall evaluate the quality of this Viterbi alignment by comparing it to a manually produced reference alignment. The parameters of the statistical alignment models are optimized with respect to a maximum likelihood criterion, which is not necessarily directly related to alignment quality. However, such an approach would require a training with manually defined alignments. Experimental evidence shall show (Section 4.6) that the statistical alignment models using this parameter estimation method do indeed obtain a good alignment quality.
We use Model 1 to Model 5 described in [Brown & Della Pietra$^+$ 93b], the Hidden Markov alignment model (HMM) [Vogel & Ney$^+$ 96, Och & Ney 00a] and a new alignment model, which we call Model 6. All these models use a different decomposition of the probability $Pr(f_1^J, a_1^J|e_1^I)$.

Heuristic models

Considerably simpler methods for obtaining word alignments use a similarity function between the types of the two languages [Smadja & McKeown$^+$ 96, Ker & Chang 97, Melamed 00]. Frequently, variations of the Dice coefficient [Dice 45] are used as similarity function. For each sentence pair, a matrix including the association scores between every word at every position is then obtained:

$$\text{dice}(e_i, f_j) = \frac{2 \cdot C(e_i, f_j)}{C(e_i) \cdot C(f_j)} \tag{4.6}$$

$C(e, f)$ denotes the co-occurrence count of word e and word f in the parallel training corpus. $C(e)$ and $C(f)$ denote the count of word e in the target sentences and the count of word f in the source sentences, respectively. From this association score matrix, the word alignment is then obtained by applying suitable heuristics. One method is to choose as alignment $a_j = i$ for position j the word with the largest association score:

$$a_j = \underset{i}{\operatorname{argmax}} \{\text{dice}(e_i, f_j)\} \tag{4.7}$$

A refinement of this method is the *competitive linking algorithm* [Melamed 00]. This method requires to first align the highest ranking word position (i, j) and then to withdraw the corresponding row and column from the association score matrix. This procedure is iteratively repeated until every source or target language word is aligned. The advantage of this approach is that so-called *indirect associations*, i.e. words that co-occur often but are not translations of each other, occur less likely. The resulting alignment contains only one-to-one alignments and typically has a higher precision.

A comparison of statistical models and heuristic models

The main advantage of the heuristic models is their simplicity. They are very easy to implement and understand. Therefore, variants of the heuristic models are widely used in the word alignment literature.
A problem of heuristic models is that the use of a specific similarity function seems to be completely arbitrary. The literature contains a large variety of different scoring functions, some including empirically adjusted parameters. As we shall show later in Section 4.6, the Dice coefficient results in a worse alignment quality than the statistical models.
We think that the approach of using statistical alignment models is more coherent. The general principle to come up with an association score between words results from statistical estimation theory and the model parameters are adjusted such that the likelihood of the models on the training corpus is maximal.

4.2.2 Statistical Alignment Models

Hidden Markov alignment model

The alignment model $Pr(f_1^J, a_1^J|e_1^I)$ can be structured without loss of generality as follows:

$$Pr(f_1^J, a_1^J|e_1^I) = Pr(J|e_1^I) \cdot \prod_{j=1}^{J} Pr(f_j, a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) \tag{4.8}$$

$$= Pr(J|e_1^I) \cdot \prod_{j=1}^{J} Pr(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) \cdot Pr(f_j|f_1^{j-1}, a_1^j, e_1^I) \tag{4.9}$$

Using this decomposition, we obtain three different probabilities: a length probability $Pr(J|e_1^I)$, an alignment probability $Pr(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I)$ and a lexicon probability $Pr(f_j|f_1^{j-1}, a_1^j, e_1^I)$. In the Hidden Markov alignment model, we assume a first-order dependence for the alignments a_j and that the lexicon probability depends only on the word at position a_j:

$$Pr(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(a_j|a_{j-1}, I) \tag{4.10}$$

$$Pr(f_j|f_1^{j-1}, a_1^j, e_1^I) = p(f_j|e_{a_j}) \tag{4.11}$$

Later, we shall describe a refinement with a dependence on $e_{a_{j-1}}$ in the alignment model. Putting everything together and assuming a simple length model $Pr(J|e_1^I) = p(J|I)$, we obtain the following basic HMM-based decomposition of $p(f_1^J|e_1^I)$:

$$p(f_1^J|e_1^I) = p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} \left[p(a_j|a_{j-1}, I) \cdot p(f_j|e_{a_j})\right] \tag{4.12}$$

with the alignment probability $p(i|i', I)$ and the translation probability $p(f|e)$.
To make the alignment parameters independent of absolute word positions, we assume that the alignment probabilities $p(i|i', I)$ depend only on the jump width $(i - i')$. Using a set of nonnegative parameters $\{c(i - i')\}$, we can write the alignment probabilities in the form:

$$p(i|i', I) = \frac{c(i - i')}{\sum_{i''=1}^{I} c(i'' - i')} \tag{4.13}$$

This form ensures the alignment probabilities satisfy the normalization constraint for each conditioning word position i', $i' = 1, \ldots, I$. This model is referred to as *homogeneous HMM* [Vogel & Ney$^+$ 96]. A similar idea was suggested by [Dagan & Church$^+$ 93].
In the original formulation of the Hidden Markov alignment model, there is no 'empty' word that generates source words having no directly aligned target word. We introduce the empty word by extending the HMM network by I empty words e_{I+1}^{2I}. The target word e_i has a corresponding empty word e_{i+I}, i.e. the position of the empty word encodes the previously visited target word. We enforce the following constraints for the transitions in the HMM network ($i \leq I$, $i' \leq I$) involving the empty word e_0:[1]

$$p(i + I|i', I) = p_0 \cdot \delta(i, i') \tag{4.14}$$
$$p(i + I|i' + I, I) = p_0 \cdot \delta(i, i') \tag{4.15}$$
$$p(i|i' + I, I) = p(i|i', I) \tag{4.16}$$

The parameter p_0 is the probability of a transition to the empty word, which has to be optimized on held-out data. In the experiments, we set $p_0 = 0.2$.

Model 1 and Model 2

While the HMM is based on first-order dependencies $p(i = a_j|a_{j-1}, I)$ for the alignment distribution, Model 1 and Model 2 use zero-order dependencies $p(i = a_j|j, I, J)$:

- Model 1 uses a uniform distribution $p(i|j, I, J) = 1/(I + 1)$:

$$Pr(f_1^J, a_1^J|e_1^I) = \frac{p(J|I)}{(I + 1)^J} \cdot \prod_{j=1}^{J} p(f_j|e_{a_j}) \tag{4.17}$$

  Hence, the word order does not affect the alignment probability.

- In Model 2, we obtain:

$$Pr(f_1^J, a_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} \left[p(a_j|j, I, J) \cdot p(f_j|e_{a_j})\right] \tag{4.18}$$

  To reduce the number of alignment parameters, we ignore the dependence on J in the alignment model and use a distribution $p(a_j|j, I)$ instead of $p(a_j|j, I, J)$.

[1] $\delta(i, i')$ is the Kronecker function, which is 1 if $i = i'$ and zero otherwise.

4.2.3 Fertility-based Alignment Models

In the following, we give a short description of the fertility-based alignment models of [Brown & Della Pietra$^+$ 93b]. A gentle introduction can be found in [Knight 99b].
The fertility-based alignment models [Brown & Della Pietra$^+$ 93b] (Model 3, Model 4 and Model 5) have a significantly more complicated structure than the simple Model 1 and Model 2. The fertility ϕ_i of a word e_i in position i is defined as the number of aligned source words:

$$\phi_i = \sum_j \delta(a_j, i) \tag{4.19}$$

The fertility-based alignment models contain a probability distribution $p(\phi|e)$ that the target word e is aligned to ϕ words. In such a way, it can be modeled that for instance the German word *'übermorgen'* produces four English words (*'the day after tomorrow'*). In particular, the fertility $\phi = 0$ is used for prepositions or articles, which are frequently dropped in the translation.
To describe the fertility-based alignment models, we introduce the *inverted* alignments as an alternative alignment representation, which define a mapping from *target* to *source* positions rather the other way round. We allow *several* positions in the source language to be covered, i.e., we consider alignments B of the form:

$$B : i \rightarrow B_i \subset \{1, \ldots, j, \ldots, J\} \tag{4.20}$$

An important constraint for the inverted alignment is that *all* positions of the source sentence must be covered exactly *once*, i.e. the B_i have to form a partition of the set $\{1, \ldots, j, \ldots, J\}$. The number of words $\phi_i = |B_i|$ is the fertility of the word e_i. In the following, B_{ik} refers to the k-th element of B_i in ascending order.
The inverted alignments B_0^I are a different way to represent normal alignments a_1^J. The set B_0 contains the positions of all source words that are aligned to the empty word. Fertility-based alignment models use the following decomposition and assumptions:

$$\begin{aligned}
Pr(f_1^J, a_1^J|e_1^I) &= Pr(f_1^J, B_0^I|e_1^I) && (4.21)\\
&= Pr(B_0|B_1^I) \cdot Pr(f_1^J, B_1^I|e_1^I) && (4.22)\\
&= Pr(B_0|B_1^I) \cdot \prod_{i=1}^{I} \prod_{k=1}^{|B_i|} Pr(f_{B_{ik}}, B_{ik}|B_1^{i-1}, e_1^I, B_{i1}^{i,k-1}, f_{B_1}^{B_{i-1}}, f_{B_{i1}}^{B_{i,k-1}})\\
&= Pr(B_0|B_1^I) \cdot \prod_{i=1}^{I} \prod_{k=1}^{|B_i|} Pr(f_{B_{ik}}|B_1^{i-1}, e_1^I, B_{i1}^{i,k-1}, f_{B_1}^{B_{i-1}}, f_{B_{i1}}^{B_{i,k-1}}) \cdot\\
&\qquad Pr(B_{ik}|B_1^{i-1}, e_1^I, B_{i1}^{i,k-1}, f_{B_1}^{B_{i-1}}, f_{B_{i1}}^{B_{i,k-1}}) && (4.23)\\
&= Pr(B_0|B_1^I) \cdot \prod_{i=1}^{I} \prod_{k=1}^{|B_i|} p(f_{B_{ik}}|e_i) \cdot\\
&\qquad Pr(B_{ik}|B_1^{i-1}, e_1^I, B_{i1}^{i,k-1}, f_{B_1}^{B_{i-1}}, f_{B_{i1}}^{B_{i,k-1}}) && (4.24)
\end{aligned}$$

In Eq. 4.22, we structured our generative model such that the set B_0 of words aligned with the empty word is generated only after the nonempty positions have been covered. In Eq. 4.24,

we have assumed that the word $f_{B_{ik}}$ only depends on the aligned English word e_i at position i. Model 3, Model 4 and Model 5 differ now with respect to the following quantity:

$$Pr(B_i|B_1^{i-1}, e_1^I, f_{B_1}^{B_{i-1}}) = \prod_{k=1}^{|B_i|} Pr(B_{ik}|B_1^{i-1}, e_1^I, B_{i1}^{i,k-1}, f_{B_1}^{B_{i-1}}, f_{B_{i1}}^{B_{i,k-1}}) \tag{4.25}$$

- In Model 3, the dependence of B_i on its predecessor B_{i-1} is ignored:

$$Pr(B_i|B_1^{i-1}, e_1^I, f_{B_1}^{B_{i-1}}) = p(\phi_i|e_i)\ \phi_i! \prod_{j \in B_i} p(j|i, J) \tag{4.26}$$

  We obtain an (inverted) zero-order alignment model $p(j|i, J)$.

- In Model 4, there is a dependence of every word on the previous aligned word and a dependence on the word classes of the surrounding words. We have two (inverted) first-order alignment models: $p_=(\Delta j|C_f, C_e)$ and $p_>(\Delta j|C_f)$, where C_f corresponds to the word class of the word f_j and C_e corresponds to the word class of the preceding English word. The dependence on word classes will be described later in more detail. The difference to the first-order alignment model in the HMM lies in the fact that here we now have a dependence along the j-axis instead of a dependence along the i-axis. The model $p_=(\Delta j|C_f, C_e)$ is used to position the first word of a set B_i and the model $p_>(\Delta j|C_f)$ is used to position the remaining words from left to right:

$$\begin{aligned} Pr(B_i|B_1^{i-1}, e_1^I, f_{B_1}^{B_{i-1}}) &= p(\phi_i|e_i) \cdot p_=(B_{i1} - \overline{B_{\rho(i)}}|C(f_{B_{i1}}), C(e_{\rho(i)})) \cdot \\ &\quad \prod_{k=2}^{\phi_i} p_>(B_{ik} - B_{i,k-1}|C(f_{B_{ik}})) \end{aligned} \tag{4.27}$$

  The function $i \to i' = \rho(i)$ gives the largest value $i' < i$ for which $|B_{i'}| > 0$. The symbol $\overline{B_{\rho(i)}}$ denotes the average of all elements in $B_{\rho(i)}$.

- Both Model 3 and Model 4 ignore whether or not a source position has been chosen. In addition, probability mass is reserved for source positions outside the sentence boundaries. For both reasons, the probabilities of all valid alignments do not sum to unity. Such models are called deficient [Brown & Della Pietra$^+$ 93b]. Model 5 is a reformulation of Model 4 with a suitably refined alignment model to avoid deficiency. Here, we omit the specific formula. We only note that the number of alignment parameters for Model 5 is significantly larger than for Model 4.

Model 3, Model 4 and Model 5 define the probability $p(B_0|B_1^I)$ as uniformly distributed for the $\phi_0!$ possibilities given the number of words aligned to the empty word $\phi_0 = |B_0|$. Assuming a binomial distribution for the number of words aligned to the empty word, we obtain the following distribution for B_0:

$$p(B_0|B_1^I) = p(\phi_0|\sum_{i=1}^{I} \phi_i) \cdot \frac{1}{\phi_0!} \tag{4.28}$$

$$= \left(\begin{array}{c} J - \phi_0 \\ \phi_0 \end{array} \right) (1 - p_1)^{J-2\phi_0} p_1^{\phi_0} \cdot \frac{1}{\phi_0!} \tag{4.29}$$

The free parameter p_1 is associated with the number of words that are aligned to the empty word. There are $\phi_0!$ ways to order the ϕ_0 words produced by the empty word, and hence, the alignment model of the empty word is nondeficient. As we will see later in Section 4.3.2, this creates problems for Model 3 and Model 4. Therefore, we modify Model 3 and Model 4 slightly by replacing $\phi_0!$ in Eq. (4.29) with J^{ϕ_0}:

$$p(B_0|B_1^I) = \left(\begin{array}{c} J-\phi_0 \\ \phi_0 \end{array} \right) (1-p_1)^{J-2\phi_0} p_1^{\phi_0} \cdot \frac{1}{J^{\phi_0}} \tag{4.30}$$

In such a way, the alignment models for both nonempty words and the alignment model for the empty word are deficient.

Model 6

As we shall see, the alignment models with a first-order dependence (HMM, Model 4, Model 5) produce significantly better results than the other alignment models. The HMM predicts the distance between subsequent source language positions, whereas Model 4 predicts the distance between subsequent target language positions. This implies that the HMM makes use of locality in the source language whereas Model 4 makes use of locality in the target language. We expect better alignment quality by using a model that takes into account both types of dependencies. Therefore, we combine HMM and Model 4 in a log-linear way and call the result Model 6:

$$p_6(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{p_4(\mathbf{f}, \mathbf{a}|\mathbf{e})^\alpha \cdot p_{HMM}(\mathbf{f}, \mathbf{a}|\mathbf{e})}{\sum_{\mathbf{a}', \mathbf{f}'} p_4(\mathbf{f}', \mathbf{a}'|\mathbf{e})^\alpha \cdot p_{HMM}(\mathbf{f}', \mathbf{a}'|\mathbf{e})} \tag{4.31}$$

Here, the interpolation parameter α is employed to weigh Model 4 relative to the Hidden Markov alignment model. In our experiments, we use Model 4 instead of Model 5 which is significantly more efficient in training and obtains better results.
In general, we can perform a log-linear combination of several models $p_k(\mathbf{f}, \mathbf{a}|\mathbf{e})$, $k = 1, \ldots, K$ by:

$$p_6(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{\prod_{k=1}^K p_k(\mathbf{f}, \mathbf{a}|\mathbf{e})^{\alpha_k}}{\sum_{\mathbf{a}', \mathbf{f}'} \prod_{k=1}^K p_k(\mathbf{f}', \mathbf{a}'|\mathbf{e})^{\alpha_k}} \tag{4.32}$$

The interpolation factors α_k are determined in such a way that the alignment quality on held-out data is optimized.
We use a log-linear combination instead of the simpler linear combination because the values of $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ typically differ by orders of magnitude for HMM and Model 4. In such a case, we expect the log-linear combination to be better than a linear combination.

Overview of models

The main differences of the statistical alignment models lie in the alignment model (zero-order or first-order), the fertility model and the presence of deficiency. In addition, the models differ with regard to the efficiency of the E-step in the EM algorithm (Section 4.3.1). Table 4.1 shows an overview of the various properties of the alignment models.

Table 4.1: Overview of the alignment models.

Model	alignment model	fertility model	E-step	deficient
Model 1	uniform	no	exact	no
Model 2	zero-order	no	exact	no
HMM	first-order	no	exact	no
Model 3	zero-order	yes	approximate	yes
Model 4	first-order	yes	approximate	yes
Model 5	first-order	yes	approximate	no
Model 6	first-order	yes	approximate	yes

4.2.4 Computation of the Viterbi Alignment

We develop an algorithm to compute the Viterbi alignment for each alignment model. While there exist simple polynomial algorithms for the baseline Model 1 and Model 2, we are unaware of any efficient algorithm to compute the Viterbi alignment for the fertility-based alignment models.

For Model 2 (also for Model 1 as special case), we obtain:

$$\hat{a}_1^J = \operatorname*{argmax}_{a_1^J} Pr(f_1^J, a_1^J | e_1^I) \tag{4.33}$$

$$= \operatorname*{argmax}_{a_1^J} \left\{ p(J|I) \cdot \prod_{j=1}^{J} \left[p(a_j|j,I) \cdot p(f_j|e_{a_j}) \right] \right\} \tag{4.34}$$

$$= \left[\operatorname*{argmax}_{a_j} \left\{ p(a_j|j,I) \cdot p(f_j|e_{a_j}) \right\} \right]_{j=1}^{J} \tag{4.35}$$

Hence, the maximization over the $(I+1)^J$ different alignments decomposes into J maximizations of $(I+1)$ lexicon probabilities. Similarly, the Viterbi alignment for Model 2 can be computed with a complexity of $O(I \cdot J)$.

Finding the optimal alignment for the HMM is more complicated than in the case of Model 1 or Model 2. With dynamic programming, the Viterbi alignment can be obtained with a complexity of $O(I^2 \cdot J)$ [Vogel & Ney$^+$ 96].

However, for the refined alignment models, namely Model 3, Model 4, Model 5 and Model 6, the maximization over all alignments cannot be simply decomposed. The corresponding search problem is NP complete [Knight 99a]. For short sentences, a possible solution could be an A* search algorithm [Och & Ueffing$^+$ 01]. Here, we use a more efficient greedy search algorithm for the best alignment as suggested in [Brown & Della Pietra$^+$ 93b]. The basic idea is to compute the Viterbi alignment of a simple model (such as Model 2 or HMM). This alignment is then iteratively improved with respect to the alignment probability of the refined alignment model. For further details on the greedy search algorithm see [Brown & Della Pietra$^+$ 93b]. In the Appendix C, we present methods for performing an efficient computation of this pseudo-Viterbi alignment.

4.3 Training

4.3.1 EM algorithm

In this section, we describe the used approach to determine the model parameters θ. Every model has a specific set of free parameters. For example, the parameters θ for Model 4 consist of lexicon, alignment and fertility parameters:

$$\theta = \{\{p(f|e)\}, \{p_=(\Delta j|C_f, C_e)\}, \{p_>(\Delta j|C_f)\}, \{p(\phi|e)\}, p_1\} \tag{4.36}$$

To train the model parameters θ, we perform a maximum likelihood approach as described in Eq. 4.4. We do this by applying the EM algorithm [Baum 72]. The different models are trained in succession on the same data, where the final parameter values of a simpler model serve as the starting point for a more complex model.
In the E-step of Model 1, the lexicon parameter counts for one sentence pair $(\mathbf{e}, \mathbf{f})$ are calculated:

$$c(f|e; \mathbf{e}, \mathbf{f}) = \sum_{\mathbf{e},\mathbf{f}} N(\mathbf{e}, \mathbf{f}) \sum_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{j} \delta(f, f_j)\delta(e, e_{a_j}) \tag{4.37}$$

Here, $N(\mathbf{e}, \mathbf{f})$ is the training corpus count of the sentence pair $(\mathbf{f}, \mathbf{e})$. In the M-step, the lexicon parameters are computed:

$$p(f|e) = \frac{\sum_s c(f|e; \mathbf{f}_s, \mathbf{e}_s)}{\sum_{s,f} c(f|e; \mathbf{f}_s, \mathbf{e}_s)} \tag{4.38}$$

Similarly, the alignment and fertility probabilities can be estimated for all other alignment models [Brown & Della Pietra$^+$ 93b]. When bootstrapping from a simpler model to a more complex model, the simpler model is used to weigh the alignments and the counts are accumulated for the parameters of the more complex model.
In principle, the sum over all $(I+1)^J$ alignments has to be performed in the E-step. Evaluating this by explicitly enumerating all alignments would be infeasible. Fortunately, Model 1, Model 2 and HMM have a particularly simple mathematical form such that the EM algorithm can be performed exactly, i.e. in the E-step, all alignments can be taken into account efficiently. For the HMM, this is referred to as Baum–Welch algorithm [Baum 72].
Since we do not know of any efficient way to avoid the explicit summation over all alignments in the EM algorithm for the fertility-based alignment models, the counts are collected only over a subset of promising alignments. For Model 3 to Model 6, we perform the count collection only over a small number of good alignments. To keep the training fast, we consider only a small fraction of all alignments. We compare three different methods for using subsets of varying sizes:

- The simplest method is to perform Viterbi training using only the best alignment found. As the Viterbi alignment computation itself is very time-consuming for Model 3 to Model 6, an approxmative method for computing the Viterbi alignment is used [Brown & Della Pietra$^+$ 93b].

- [Al-Onaizan & Curin$^+$ 99] suggest using also the neighboring alignments of the best alignment found. For an exact definition of the neighborhood of an alignment, the reader is referred to the Appendix C.

- [Brown & Della Pietra$^+$ 93b] use an even larger set of alignments, including also the so-called *pegged* alignments, which is a large set of alignments with a high probability $Pr(f_1^J, a_1^J|e_1^I)$. The construction method for these alignments [Brown & Della Pietra$^+$ 93b] guarantees that for each lexical relationship in every sentence pair, at least one alignment is considered.

In Section 4.6, we show that by using the HMM instead of Model 2 in bootstrapping the fertility-based alignment models, the alignment quality can be significantly improved. In Appendix C, we present a method for performing an efficient training algorithm of the fertility-based alignment models.

4.3.2 Is Deficiency a Problem?

When using the EM algorithm on the standard versions of Model 3 and Model 4, we observe that during the EM iterations more and more words are aligned to the empty word. This results in a poor alignment quality because too many words are aligned to the empty word. This does not occur when using the other alignment models. We believe that this is due to the deficiency of Model 3 and Model 4.
The use of the EM algorithm guarantees that the likelihood increases for each iteration. This holds for both deficient and nondeficient models. However, for deficient models, by simply reducing the amount of deficiency (i.e. the 'wasted' probability mass), the likelihood increases. In Model 3 and Model 4 as defined in [Brown & Della Pietra$^+$ 93b], the alignment model for nonempty words is deficient, but the alignment model for the empty word is nondeficient. Hence, the EM algorithm can increase likelihood by simply aligning more and more words to the empty word.[2]
Therefore, we modify Model 3 and Model 4 slightly such that the empty word also has a deficient alignment model. The alignment probability is set to $p(j|i, J) = 1/J$ for each source word aligned to the empty word. Another remedy adopted in [Och & Ney 00a] is to choose a value for the parameter p_1 of the empty word fertility and keep it fixed.

4.3.3 Smoothing

To overcome the problem of overfitting on the training data and to cope better with rare words, we smooth the alignment and fertility probabilities. For the alignment probabilities of the HMM (and similarly for Model 4 and Model 5), we perform an interpolation with a uniform distribution $p(i|j, I) = 1/I$:

$$p'(a_j|a_{j-1}, I) = (1 - \alpha) \cdot p(a_j|a_{j-1}, I) + \alpha \cdot \frac{1}{I} \tag{4.39}$$

For the fertility probabilities, we assume that there is a dependence on the number of letters $g(e)$ of word e and estimate a fertility distribution $p(\phi|g)$ using the EM algorithm. Typically, longer words have a higher fertility. In such a way, we can learn that the longer words usually have a larger fertility than shorter words.

[2] This effect did not occur in [Brown & Della Pietra$^+$ 93b] as Model 3 and Model 4 were not trained directly.

Using an interpolation parameter β, the fertility distribution is then computed as:

$$p'(\phi|e) = \left(1 - \frac{\beta}{\beta + N(e)}\right) \cdot p(\phi|e) + \frac{\beta}{\beta + N(e)} \cdot p(\phi|g(e)) \tag{4.40}$$

Here, $N(e)$ denotes the frequency of word e in the training corpus. This linear interpolation ensures that for frequent words, i.e. $N(e) \gg \beta$, the specific distribution $p(\phi|e)$ dominates and that for rare words, i.e. $N(e) \ll \beta$, the general distribution $p(\phi|g(e))$ dominates.
The interpolation parameters α and β are determined in such a way that the alignment quality on held-out data is optimized.

4.3.4 Bilingual Dictionary

A conventional bilingual dictionary can be considered to be an additional knowledge source that can be used in training. We assume that the dictionary is a list of word strings $(\mathbf{e}, \mathbf{f})$. The entries for each language can be a single word or an entire phrase.
To integrate a dictionary into the EM algorithm, we compare two different methods:

- [Brown & Della Pietra$^+$ 93a] developed a multinomial model for the process of constructing a dictionary (by a human lexicographer). By applying suitable simplifications, the method boils down to adding every dictionary entry to the training corpus with an entry-specific count called *effective multiplicity* $\mu(\mathbf{e}, \mathbf{f})$.

  $$\mu(\mathbf{e}, \mathbf{f}) = \frac{\lambda(\mathbf{e}) \cdot p(\mathbf{f}|\mathbf{e})}{1 - e^{\lambda(\mathbf{e}) \cdot p(\mathbf{f}|\mathbf{e})}} \tag{4.41}$$

  This count is used instead of $N(\mathbf{e}, \mathbf{f})$ in the EM algorithm as shown in Eq. 4.37. Here, $\lambda(\mathbf{e})$ is an additional parameter describing the size of the sample, which is used to estimate the model $p(\mathbf{f}|\mathbf{e})$.

- [Och & Ney 00a] suggest setting the effective multiplicity of a dictionary entry to a large value $\mu^+ >> 1$ if the two words co-occur and to a low value otherwise.

  $$\mu(\mathbf{e}, \mathbf{f}) = \begin{cases} \mu^+ & \text{if } \mathbf{e} \text{ and } \mathbf{f} \text{ co-occur} \\ \mu^- & \text{otherwise} \end{cases} \tag{4.42}$$

  As a result, only dictionary entries that indeed occur in the training corpus are used. The motivation behind this is to avoid a deterioration of the alignment by out-of-domain dictionary entries. Every entry that does co-occur in the training corpus can be assumed correct and should therefore obtain a high count. We set $\mu^- = 0$.

4.4 Symmetrization

Here, we describe various methods for performing a symmetrization of our directed statistical alignment models by applying a heuristic postprocessing step that combines the alignments in both translation directions (source to target, target to source).
The baseline alignment model does not allow a source word to be aligned to two or more target words. Therefore, lexical correspondences like the German compound word *'Zahnarzttermin'*

for *'dentist's appointment'* cause problems because a single source word must be mapped on two or more target words. Therefore, the resulting Viterbi alignment of the standard alignment models has a systematic loss in recall.
To solve this problem, we perform a training in both translation directions (source to target, target to source). As a result, we obtain two alignments a_1^J and b_1^I for each sentence pair. Let $A_1 = \{(a_j, j)|a_j > 0\}$ and $A_2 = \{(i, b_i)|b_i > 0\}$ denote the sets of alignments in the two Viterbi alignments. To increase the quality of the alignments, we combine A_1 and A_2 into one alignment matrix A using the following combination methods:

- Intersection: $A = A_1 \cap A_2$
- Union: $A = A_1 \cup A_2$
- Refined Method: In a first step, the intersection $A = A_1 \cap A_2$ is determined. The elements of this intersection result from both Viterbi alignments and are therefore very reliable. Then, we extend the alignment A iteratively by adding alignments (i, j) occurring only in the alignment A_1 or in the alignment A_2 if neither f_j nor e_i have an alignment in A, or if the following conditions both hold:
  - the alignment (i, j) has a horizontal neighbor $(i - 1, j)$, $(i + 1, j)$ or a vertical neighbor $(i, j - 1)$, $(i, j + 1)$ that is already in A,
  - the set $A \cup \{(i, j)\}$ does not contain alignments with both horizontal and vertical neighbors.

Obviously, the intersection yields an alignment consisting of only one-to-one alignments with a higher precision and a lower recall. The union yields a higher recall and a lower precision of the combined alignment. It depends on the final application of the word alignment whether a higher precision or a higher recall is preferred. In applications such as statistical MT [Och & Tillmann$^+$ 99], a higher recall is more important [Och & Ney 00a]. In lexicography applications, we might be interested in alignments with a very high precision obtained by performing an alignment intersection.

4.5 Evaluation Methodology

In the following, we present an annotation scheme for word alignments and a corresponding evaluation criterion.
Manually performing a word alignment is a complicated and ambiguous task [Melamed 98]. Therefore, we use an annotation scheme that explicitly allows for ambiguous alignments. The persons performing the annotation are asked to specify two different kinds of alignments: a S (sure) alignment, which is used for alignments that are unambiguous and a P (possible) alignment, which is used for ambiguous alignments. The P label is used especially to align words within idiomatic expressions, free translations, and missing function words ($S \subseteq P$).
The reference alignment thus obtained may contain many-to-one and one-to-many relationships. Figure 4.3 shows an example of a manually aligned sentence with S and P labels.

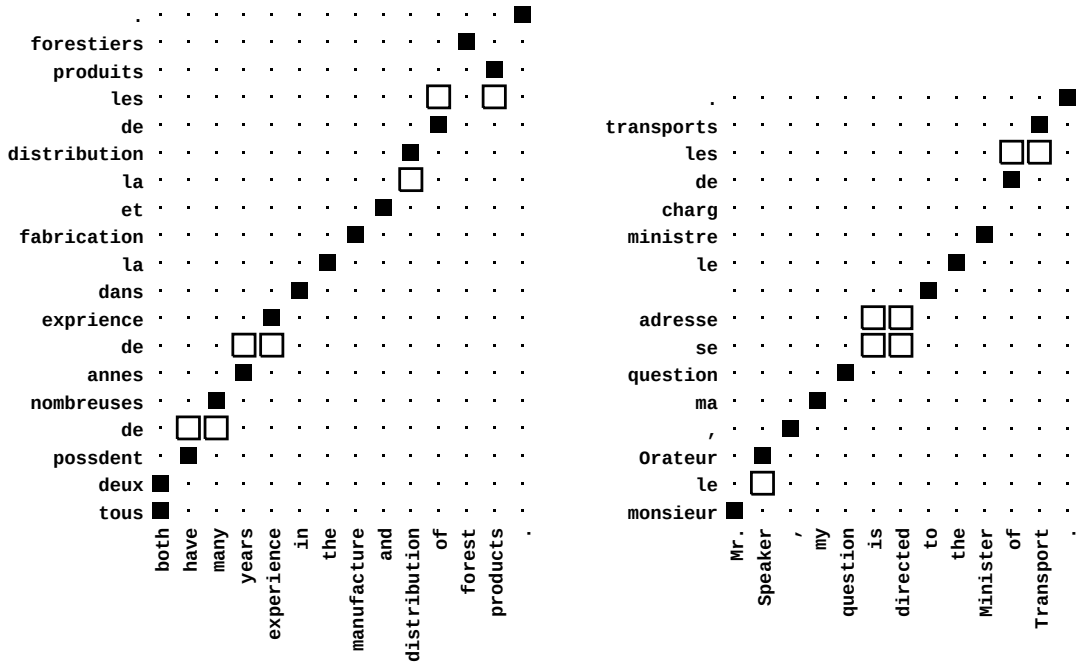


Figure 4.3: Example of a manual alignment with *S(ure)* (filled squares) and *P(ossible)* (unfilled squares) connections.

The quality of an alignment $A = \{(j, a_j)|a_j > 0\}$ is computed by appropriately redefined precision and recall measures:

$$recall = \frac{|A \cap S|}{|S|}, \quad precision = \frac{|A \cap P|}{|A|} \tag{4.43}$$

and the following alignment error rate, which is derived from the well-known F-measure:

$$AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{4.44}$$

In such a way, a recall error can only occur if a S(ure) alignment is not found and a precision error can only occur if the found alignment is not even P(ossible).
The set of sentence pairs, for which the manual alignment is produced, is randomly selected from the training corpus. It should be emphasized that all the training is done in a completely unsupervised way, i.e. no manual alignments are used. From this point of view, there is no need to have a separate test corpus.
Typically, the annotation is performed by two human annotators, producing sets S_1, P_1, S_2, P_2. To increase the quality of the reference alignment, the annotators are presented the mutual errors and asked to improve their alignments where possible. From these alignments, we finally generate a reference alignment that contains only those S(ure) connections where both annotators agree and all P(ossible) connections from both annotators. This can be done by forming the intersection of the sure alignments ($S = S_1 \cap S_2$) and the union of the possible alignments ($P = P_1 \cup P_2$), respectively. In such a way, we obtain an alignment error rate of zero percent when we compare the sure alignments of every single annotator with the combined reference alignment.

Table 4.2: Corpus statistics of VERBMOBIL task.

		German	English
Training Corpus	Sentences	34446	
	Words	329625	343076
	Vocabulary	5936	3505
	Singletons	2600	1305
Bilingual Dictionary	Entries	4404	
	Words	4758	5543
Test Corpus	Sentences	354	
	Words	3233	3109

4.6 Experiments

We present results on the VERBMOBIL and the HANSARDS task. The VERBMOBIL task [Wahlster 00] is a speech translation task in the domain of appointment scheduling, travel planning and hotel reservation. The bilingual sentences used in training are correct transcriptions of spoken dialogues. However, they include spontaneous speech effects such as hesitations, false starts and ungrammatical phrases. The French-English HANSARDS task consists of the debates in the Canadian Parliament. This task has a very large vocabulary of about $100\ 000$ French words and $80\ 000$ English words.[3]

The corpus statistics are shown in Table 4.2 and Table 4.3. The number of running words and the vocabularies are based on full-form words including the punctuation marks. We produced smaller training corpora by randomly choosing 500, $2\ 000$ and $8\ 000$ sentences from the VERBMOBIL task and 500, $8\ 000$, $128\ 000$ sentences from the HANSARDS task.

For both tasks, we manually aligned a randomly chosen subset of the training corpus. From this corpus, the first 100 sentences are used as development corpus to optimize the model parameters that are not trained via the EM algorithm, e.g. the smoothing parameters. The remaining sentences are used as test corpus.

In the following, the sequence of used models and the number of training iterations used for each model is called *training scheme*. The standard training scheme on VERBMOBIL is $1^5H^53^34^36^3$. This means 5 iterations of Model 1, 5 iterations of HMM, 3 iterations of Model 3, 3 iterations of Model 4 and 3 iterations of Model 6. On HANSARDS, we use $1^5H^{10}3^34^36^3$. This training scheme typically gives very good results and does not lead to overfitting. We use the slightly modified versions of Model 3 and Model 4 described in Section 4.3.2 and smooth the fertility and the alignment parameters. In the E-step of the EM algorithm for the fertility-based alignment models, we use the Viterbi alignment and its neighborhood. If not stated otherwise, the conventional dictionary is not used.

Models and Training Schemes.

Table 4.4 and Table 4.5 compare the alignment quality of various models and training schemes. In general, we observe that Model 4, Model 5 and Model 6 yield significantly better results than the simple Model 1 or the Dice coefficient. Typically, the best results are obtained with Model

[3] We do not use the Blinker annotated corpus described in [Melamed 98] since the domain is very special (the Bible) and a different annotation methodology is used.

Table 4.3: Corpus statistics of HANSARDS task.

		French	English
Training Corpus	Sentences	1470K	
	Words	24.33M	22.16M
	Vocabulary	100269	78332
	Singletons	40199	31319
Bilingual Dictionary	Entries	28701	
	Words	28702	30186
Test Corpus	Sentences	500	
	Words	8749	7946

Table 4.4: Comparison of alignment error rate [%] for various training schemes (VERBMOBIL task, Dice+C: Dice coefficient with competitive linking).

		Size of Training Corpus			
Model	Training Scheme	0.5K	2K	8K	34K
Dice		28.4	29.2	29.1	29.0
Dice+C		21.5	21.8	20.1	20.4
Model 1	1^5	19.3	19.0	17.8	17.0
Model 2	$1^5 2^5$	27.7	21.0	15.8	13.5
HMM	$1^5 H^5$	19.1	15.4	11.4	9.2
Model 3	$1^5 2^5 3^3$	25.8	18.4	13.4	10.3
	$1^5 H^5 3^3$	18.1	14.3	10.5	8.1
Model 4	$1^5 2^5 3^3 4^3$	23.4	14.6	10.0	7.7
	$1^5 H^5 4^3$	17.3	11.7	9.1	6.5
	$1^5 H^5 3^3 4^3$	16.8	11.7	8.4	6.3
Model 5	$1^5 H^5 4^3 5^3$	17.3	11.4	8.7	6.2
	$1^5 H^5 3^3 4^3 5^3$	16.9	11.8	8.5	5.8
Model 6	$1^5 H^5 4^3 6^3$	17.2	11.3	8.8	6.1
	$1^5 H^5 3^3 4^3 6^3$	16.4	11.7	8.0	5.7

6. This holds for an extremely small training corpus of only 500 sentences up to a training corpus of 1.5 million sentences. However, the improvement by using a larger training corpus is more significant if more refined models are used. Interestingly, already on a tiny corpus of only 500 sentences, alignment error rates under 30% are achieved for all models, and the best models are somewhat under 20%.

We observe that the quality obtained with a specific model heavily depends on the training scheme that is used to bootstrap this model.

Table 4.5: Comparison of alignment error rate [%] for various training schemes (HANSARDS task, Dice+C: Dice coefficient with competitive linking).

		Size of Training Corpus			
Model	Training Scheme	0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1^5	40.6	33.6	28.6	25.9
Model 2	$1^5 2^5$	46.7	29.3	22.0	19.5
HMM	$1^5 H^5$	26.3	23.3	15.0	10.8
Model 3	$1^5 2^5 3^3$	43.6	27.5	20.5	18.0
	$1^5 H^5 3^3$	27.5	22.5	16.6	13.2
Model 4	$1^5 2^5 3^3 4^3$	41.7	25.1	17.3	14.1
	$1^5 H^5 3^3 4^3$	26.1	20.2	13.1	9.4
	$1^5 H^5 4^3$	26.3	21.8	13.3	9.3
Model 5	$1^5 H^5 4^3 5^3$	26.5	21.5	13.7	9.6
	$1^5 H^5 3^3 4^3 5^3$	26.5	20.4	13.4	9.4
Model 6	$1^5 H^5 4^3 6^3$	26.0	21.6	12.8	8.8
	$1^5 H^5 3^3 4^3 6^3$	25.9	20.3	12.5	8.7

Heuristic models vs. Model 1.

We have pointed out in Section 4.2 that from a theoretical viewpoint, the main advantage of statistical alignment models in comparison to heuristic models is the well-founded mathematical theory that underlies their parameter estimation. Table 4.4 and Table 4.5 show that the statistical alignment models significantly outperform the heuristic Dice coefficient and the heuristic Dice coefficient with competitive linking (Dice+C). The simple Model 1 already achieves better results.
Analyzing the alignment quality obtained in the EM training of Model 1 is very instructive. Figure 4.4 shows the alignment quality over the iteration numbers of Model 1. We see that the first iteration of Model 1 achieves significantly worse results than the Dice coefficient, but already the second iteration of Model 1 gives better results.

Model 2 vs. HMM.

An important result is that the Hidden Markov alignment model achieves significantly better results than Model 2. We attribute this to the fact that the HMM is a homogeneous first-order alignment model, which is able to better represent the locality and monotonicity properties of natural languages. Both models have the important property that they allow an efficient implementation of the EM algorithm (Section 4.3). On the largest VERBMOBIL task, the HMM achieves an improvement of 3.8 % over Model 2. On the largest HANSARDS task, the improvement is 8.7 %. Interestingly, this advantage continues to hold after bootstrapping more refined models. On Model 4, the improvement is 1.4% and 4.8 %, respectively. We conclude that it is important to bootstrap the refined alignment models with good initial parameters. Obviously, if we use Model 2 for bootstrapping, we eventually obtain a poor local optimum.

Table 4.6: Effect of using more alignments in training fertility models on alignment error rate [%] (VERBMOBIL task).

		Size of Training Corpus			
Training Scheme	Alignment Set	0.5K	2K	8K	34K
$1^5H^53^34^36^3$	Viterbi	17.8	12.6	8.6	6.6
	+neighbors	16.4	11.7	8.0	5.7
	+pegging	16.4	11.2	8.2	5.7
$1^52^53^34^35^3$	Viterbi	24.1	16.0	11.6	8.6
	+neighbors	22.9	14.2	9.8	7.6
	+pegging	22.0	13.3	9.7	6.9

Table 4.7: Effect of using more alignments in training fertility models on alignment error rate [%] (HANSARDS task).

		Size of Training Corpus		
Training Scheme	Alignment Set	0.5K	8K	128K
$1^5H^{10}3^34^36^3$	Viterbi	25.8	20.3	12.6
	+neighbors	25.9	20.3	12.5
	+pegging	25.8	19.9	12.6
$1^52^53^34^35^3$	Viterbi	41.9	25.1	17.6
	+neighbors	41.7	24.8	16.1
	+pegging	41.2	23.7	15.8

The number of alignments in training.

Table 4.6 and Table 4.7 show the results obtained by using different numbers of alignments in the training of the fertility-based alignment models. We compare the three different approaches described in Section 4.3: using only the Viterbi alignment, using in addition the neighborhood of the Viterbi alignment and using the pegged alignments. To reduce the training time, we restrict the number of pegged alignments by using only those where $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ is not much smaller than the probability of the Viterbi alignment. This reduces the training time drastically. However, for the large HANSARDS corpus, there still is an unacceptable large training time. Therefore, we report the results for only up to 128K training sentences.
The effect of pegging strongly depends on the quality of the starting point used for training the fertility-based alignment models. If we use Model 2 as starting point, we observe a significant improvement by using the neighborhood alignments and the pegged alignments. If we only use the Viterbi alignment, the results are significantly worse than using additionally the

Table 4.8: Computing time on the 34K VERBMOBIL task (on 600 MHz Pentium III machine).

	Seconds per Iteration		
Alignment Set	Model 3	Model 4	Model 5
Viterbi	48.0	251.0	248.0
+neighbors	101.0	283.0	276.0
+pegging	129.0	3348.0	3356.0

Table 4.9: Effect of smoothing on alignment error rate [%] (VERBMOBIL task, Model 6).

	Size of Training Corpus			
Smoothing Method	0.5K	2K	8K	34K
no	19.7	14.9	10.9	8.3
Fertility	18.4	14.3	10.3	8.0
Alignment	16.8	13.2	9.1	6.4
Alignment and Fertility	16.4	11.7	8.0	5.7

Table 4.10: Effect of smoothing on alignment error rate [%] (HANSARDS task, Model 6).

	Size of Training Corpus			
Smoothing Method	0.5K	8K	128K	1470K
no	28.6	23.3	13.3	9.5
Fertility	28.3	22.5	12.7	9.3
Alignment	26.5	21.2	13.0	8.9
Alignment and Fertility	25.9	20.3	12.5	8.7

neighborhood of the Viterbi alignment. If we use HMM as starting point, we observe a much smaller effect. We conclude that using more alignments in training is a way to avoid a poor local optimum.
Table 4.8 shows the computing time for performing one iteration of the EM algorithm. Using a larger set of alignments, the training time for Model 4 and Model 5 increases significantly. Since using the pegging alignments yields only a moderate improvement, all following results are obtained by using the neighborhood of the Viterbi alignment without pegging.

Effect of smoothing.

Table 4.9 and Table 4.10 show the effect of smoothing the alignment and fertility probabilities on the alignment error rate. We observe a significant improvement by smoothing the alignment probabilities and a minor improvement by smoothing the fertility probabilities. An analysis of the alignments shows that the smoothing the fertility probabilities significantly reduces the problem that rare words frequently form 'garbage collectors' in that they tend to align to too many words [Brown & Della Pietra$^+$ 93a].
Without smoothing, we observe early overfitting: alignment error rate increases after the second iteration of HMM as is shown in Figure 4.5. On VERBMOBIL, the best alignment error rate is obtained in the second iteration. On HANSARDS, the best alignment error rate is obtained in the sixth iteration. In the following iterations, the alignment error rate increases significantly. With smoothing the alignment parameters, we obtain a lower alignment error rate, overfitting occurs later and its effect is smaller.

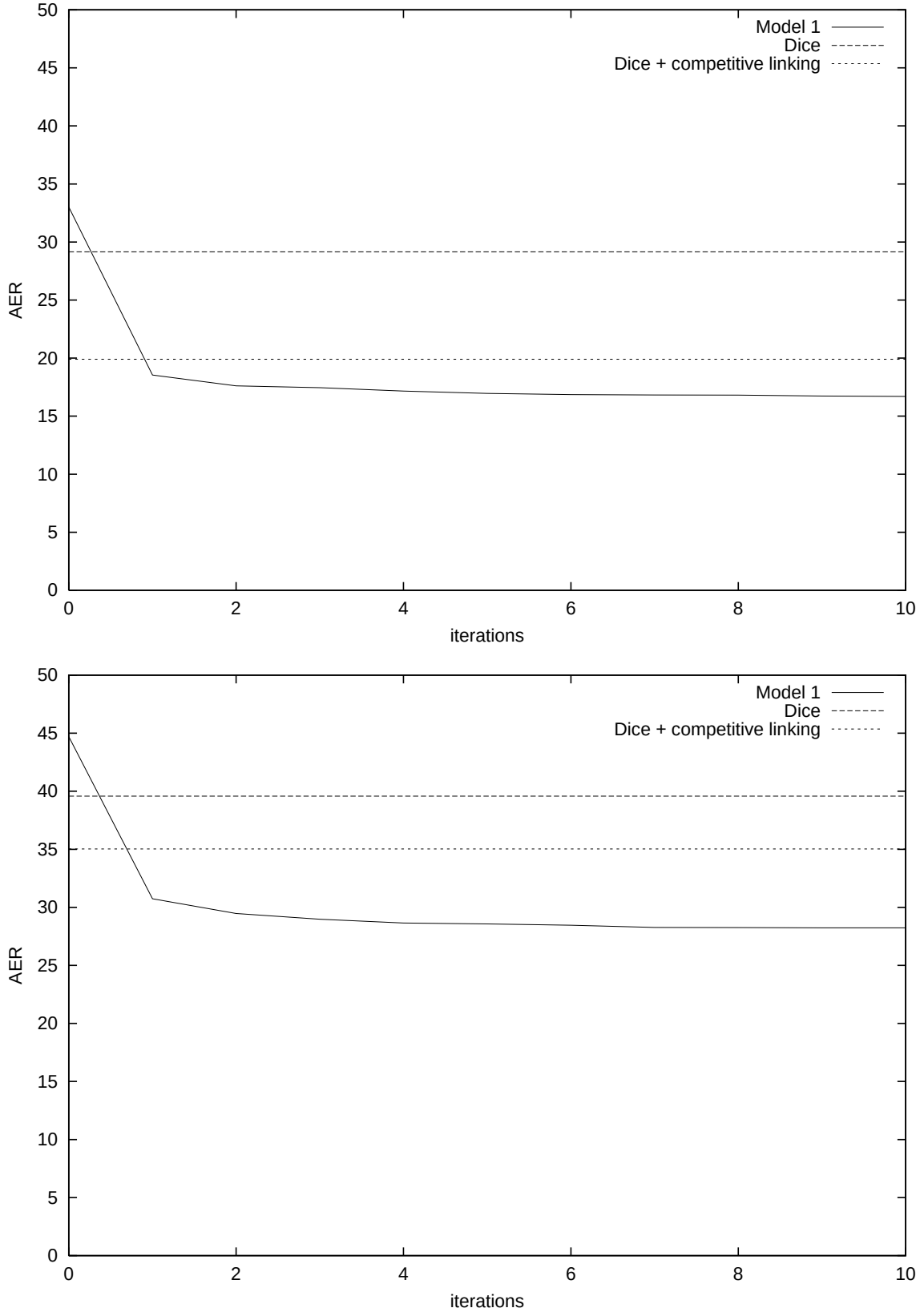


Figure 4.4: Comparison of alignment error rate [%] for Model 1 and Dice coefficient (34K VERBMOBIL task, 128K HANSARDS task).

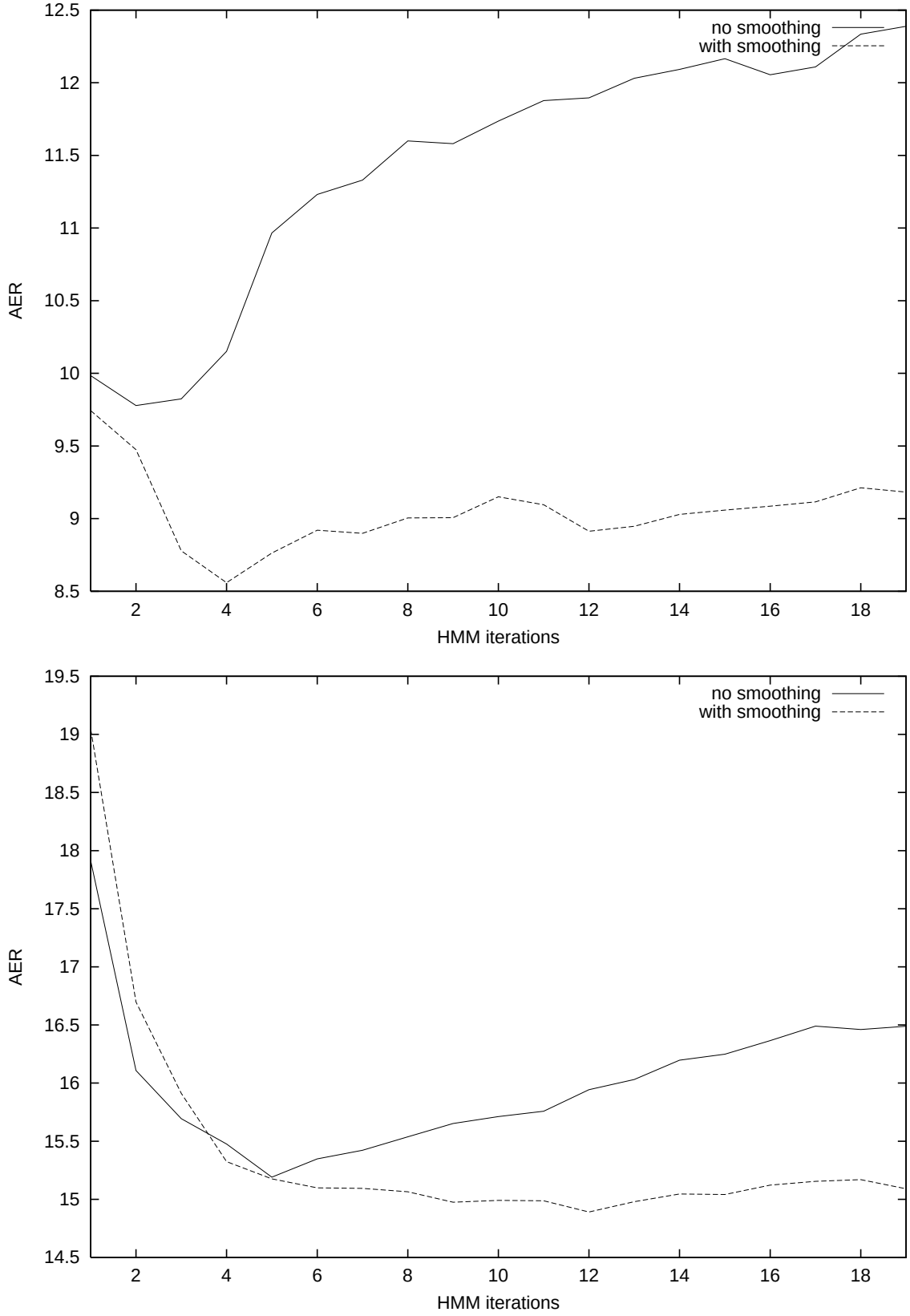


Figure 4.5: Overfitting on the training data with the Hidden Markov alignment model using various smoothing parameters (34K VERBMOBIL task, 128K HANSARDS task).

Table 4.11: Effect of word classes on alignment error rate [%] (VERBMOBIL task).

	Size of Training Corpus			
Word Classes	0.5K	2K	8K	34K
no	16.5	11.7	8.0	6.3
yes	16.4	11.7	8.0	5.7

Table 4.12: Effect of word classes on alignment error rate [%] (HANSARDS task).

	Size of Training Corpus			
Word Classes	0.5K	8K	128K	1470K
no	25.5	20.7	12.8	8.9
yes	25.9	20.3	12.5	8.7

Alignment models depending on word classes.

Table 4.11 and Table 4.12 show the effects of including a dependence on word classes in the alignment model as described in Section 4.2.3. The word classes are always trained on the same subset of the training corpus as used for the training of the alignment models. We do not observe any significant improvement when using a small training corpus. A possible reason is that either the word classes themselves or the resulting large number of alignment parameters cannot be estimated reliably using a small training corpus. When using a large training corpus, there is a clear improvement on the VERBMOBIL task.

Using a conventional bilingual dictionary.

Table 4.13 shows the effect of the conventional dictionary in the training on VERBMOBIL and HANSARDS. We compare the two methods for using the dictionary described in Section 4.3.4. We observe that the method with a fixed threshold of $\mu^+ = 16$ gives the best results. The method with a varying μ gives worse results, but this method has one parameter less to be optimized on held-out data.
On small corpora, there is an improvement of up to 6.7% on the VERBMOBIL task and 3.2% on the HANSARDS task; but by using a larger training corpus, the improvements reduce to 1.1% and 0.4%, respectively. Interestingly, the overall improvement by a conventional dictionary is small compared to the improvement achieved by better alignment models.

Table 4.13: Effect of using a conventional dictionary on alignment error rate [%] (VERBMOBIL task).

	Size of Training Corpus			
Bilingual Dictionary	0.5K	2K	8K	34K
no	16.4	11.7	8.0	5.7
yes/μ var.	10.9	9.0	6.9	5.1
yes/$\mu^+ = 8$	9.7	7.6	6.0	5.1
yes/$\mu^+ = 16$	10.0	7.8	6.0	4.6
yes/$\mu^+ = 32$	10.4	8.5	6.4	4.7

Table 4.14: Effect of using a conventional dictionary on alignment error rate [%] (HANSARDS task).

	Size of Training Corpus			
Bilingual Dictionary	0.5K	8K	128K	1470K
no	25.9	20.3	12.5	8.7
yes/μ var.	23.3	18.3	12.3	8.6
yes/$\mu^+ = 8$	22.7	18.5	12.2	8.6
yes/$\mu^+ = 16$	23.1	18.7	12.1	8.6
yes/$\mu^+ = 32$	24.9	20.2	11.7	8.3

Generalized Alignments.

In the following, we compare the results obtained using different translation directions and using the symmetrization methods described in Section 4.4. Table 4.15 and Table 4.16 show precision, recall and alignment error rate for the final iteration of Model 6 for both translation directions. In this experiment, we use the conventional dictionary as well. Especially for the VERBMOBIL task with the language pair German-English, we observe that for German as the source language the alignment error rate is much higher than for English as source language. A possible reason is that the baseline alignment representation as a vector a_1^J does not allow the frequent German word compounds to be aligned to more than one English word.
The effect of merging alignments by forming the intersection, the union or the refined combination of the Viterbi alignments in both translation directions is shown in Table 4.17 and Table 4.18. Figure 4.6 shows the corresponding precision–recall graphs. By using the refined combination, we can increase precision and recall on the HANSARDS task. The lowest alignment error rate on the HANSARDS task is obtained by using the intersection method. By forming a union or intersection of the alignments, we can obtain very high recall or precision values both on HANSARDS and VERBMOBIL.
Table 4.19, Table 4.20, Table 4.21 and Table 4.22 show the corresponding results using the Hidden Markov alignment model. We observe that the Hidden Markov alignment model produces systematically worse results than Model 6, even using the symmetrization methods. Yet, the improvement by Model 6 gets very small if large amounts of training data are used.

4.7 Conclusion

We have discussed in detail various statistical and heuristic word alignment models. We have described various modifications and extensions to models known in the literature. A new statistical alignment model (Model 6) has been developed, which has yielded the best results. We have presented two methods for including a conventional dictionary in training. We have described heuristic symmetrization algorithms that combine the alignments in both translation directions producing an alignment with a higher precision, a higher recall or an improved alignment error rate.
We have suggested measuring the quality of an alignment model using the quality of the Viterbi alignment compared to a manually produced reference alignment. This quality measure has the advantage of automatic evaluation. To produce the reference alignment, we have used a

Table 4.15: Effect of training corpus size and translation direction on precision, recall and alignment error rate [%] (VERBMOBIL task + dictionary, Model 6).

	English → German			German → English		
Corpus Size	prec	rec	AER	prec	rec	AER
0.5K	87.6	93.1	10.0	77.9	80.3	21.1
2K	90.5	94.4	7.8	88.1	88.1	11.9
8K	92.7	95.7	6.0	90.2	89.1	10.3
34K	94.6	96.3	4.6	92.5	89.5	8.8

Table 4.16: Effect of alignment combination on precision, recall and alignment error rate [%] (HANSARDS task + dictionary, Model 6).

	English → French			French → English		
Corpus Size	prec	rec	AER	prec	rec	AER
0.5K	73.0	83.8	23.1	68.5	79.1	27.8
8K	77.0	88.9	18.7	76.0	88.5	19.5
128K	84.5	93.5	12.1	84.6	93.3	12.2
1470K	89.4	94.7	8.6	89.1	95.2	8.6

Table 4.17: Effect of alignment combination on precision, recall and alignment error rate [%] (VERBMOBIL task + dictionary, Model 6).

	Intersection			Union			Refined Method		
Corpus Size	prec	rec	AER	prec	rec	AER	prec	rec	AER
0.5K	97.5	76.8	13.6	74.8	96.1	16.9	87.8	92.9	9.9
2K	97.2	85.6	8.6	84.1	96.9	10.6	91.3	94.2	7.4
8K	97.5	86.6	8.0	87.0	97.7	8.5	92.8	96.0	5.8
34K	98.1	87.6	7.2	90.6	98.4	6.0	94.0	96.9	4.7

Table 4.18: Effect of alignment combination on precision, recall and alignment error rate [%] (HANSARDS task + dictionary, Model 6).

	Intersection			Union			Refined Method		
Corpus Size	prec	rec	AER	prec	rec	AER	prec	rec	AER
0.5K	91.5	71.3	18.7	63.4	91.6	29.0	75.5	84.9	21.1
8K	95.6	82.8	10.6	68.2	94.4	24.2	83.3	90.0	14.2
128K	96.7	90.0	6.3	77.8	96.9	16.1	89.4	94.4	8.7
1470K	96.8	92.3	5.2	84.2	97.6	11.3	91.5	95.5	7.0

Table 4.19: Effect of training corpus size and translation direction on precision, recall and alignment error rate [%] (VERBMOBIL task + dictionary, Hidden Markov alignment model).

	English → German			German → English		
Corpus Size	prec	rec	AER	prec	rec	AER
0.5K	87.7	88.9	11.7	76.7	76.3	23.5
2K	89.2	89.5	10.6	87.9	84.9	13.4
8K	91.6	90.6	8.9	89.1	85.4	12.5
34K	93.2	91.9	7.4	89.9	85.7	11.9

Table 4.20: Effect of alignment combination on precision, recall and alignment error rate [%] (HANSARDS task + dictionary, Hidden Markov alignment model).

	English → French			French → English		
Corpus Size	prec	rec	AER	prec	rec	AER
0.5K	72.8	78.6	25.1	67.5	74.9	29.9
8K	74.7	84.4	21.6	73.7	84.0	22.5
128K	83.0	90.8	14.1	82.1	91.5	14.4
1470K	87.1	92.9	10.6	86.1	92.8	11.4

Table 4.21: Effect of alignment combination on precision, recall and alignment error rate [%] (VERBMOBIL task + dictionary, Hidden Markov alignment model).

	Intersection			Union			Refined Method		
Corpus Size	prec	rec	AER	prec	rec	AER	prec	rec	AER
0.5K	95.7	69.5	18.0	69.8	88.1	23.7	83.6	82.8	16.7
2K	96.7	73.1	15.4	74.3	91.4	19.5	86.4	86.3	13.6
8K	97.8	77.6	12.3	79.7	93.5	15.1	90.5	88.9	10.2
34K	97.7	79.8	11.1	83.8	94.8	11.9	92.7	91.6	7.8

Table 4.22: Effect of alignment combination on precision, recall and alignment error rate [%] (HANSARDS task + dictionary, Hidden Markov alignment model).

	Intersection			Union			Refined Method		
Corpus Size	prec	rec	AER	prec	rec	AER	prec	rec	AER
0.5K	90.5	62.2	24.7	61.8	85.8	31.5	75.0	77.7	24.0
8K	94.9	72.5	16.7	64.8	92.6	27.2	82.2	85.7	16.4
128K	96.3	85.0	9.0	73.8	95.7	19.4	88.9	91.5	10.1
1470K	97.0	89.1	6.6	80.1	96.8	14.4	91.3	94.3	7.5

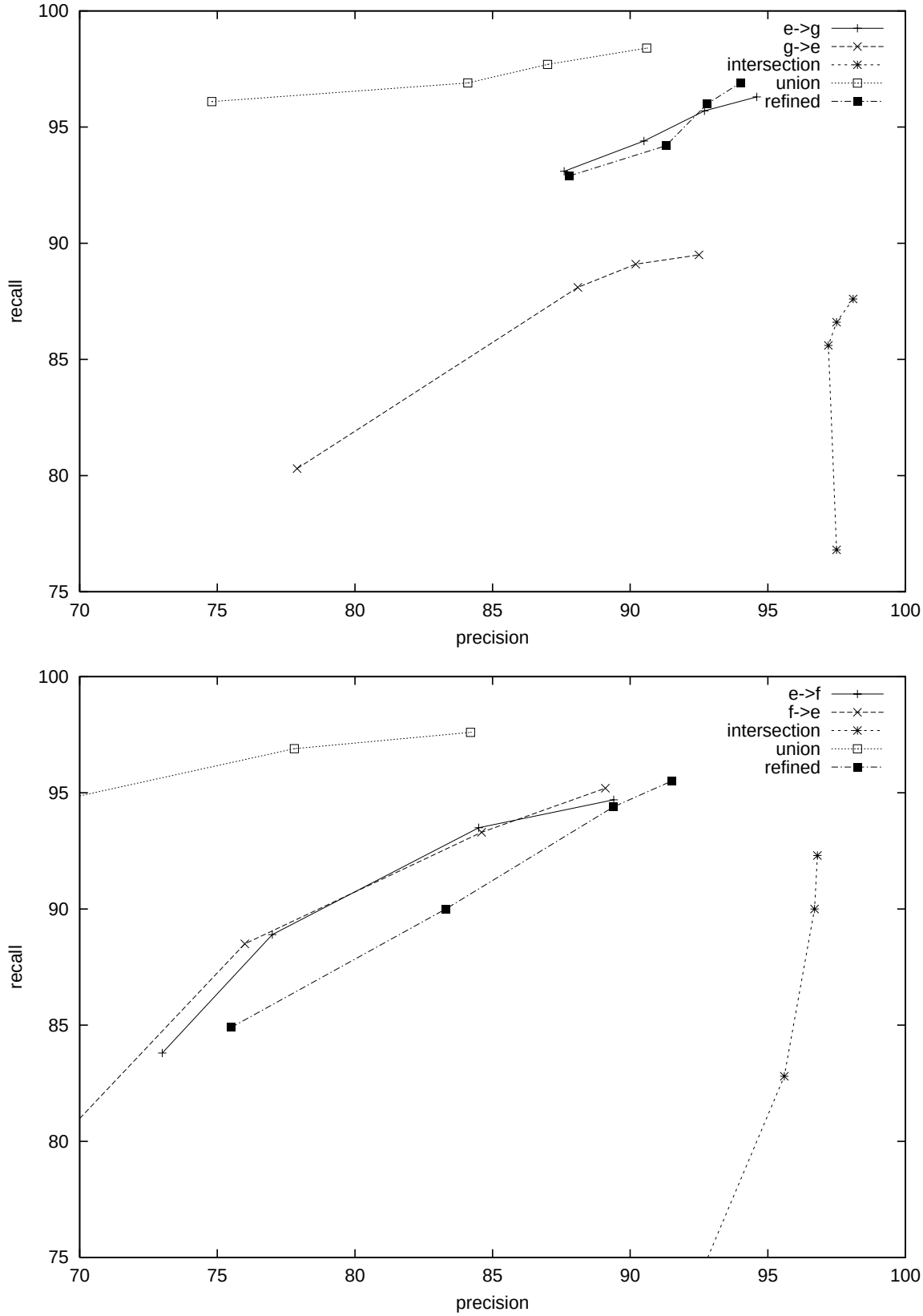


Figure 4.6: Effect of various symmetrization methods on precision and recall for the different training corpus sizes (VERBMOBIL task, HANSARDS task).

refined annotation scheme, which reduces the complications and ambiguities associated with the manual construction of a word alignment.
We have performed various experiments to assess the effect of different alignment models, training schemes and knowledge sources. The key results are:

- Statistical alignment models outperform the simple Dice coefficient.
- The best results are obtained with Model 6. In general, very important ingredients of a good model seem to be a first-order dependence of word positions and a fertility model.
- Smoothing and symmetrization have a significant effect on the alignment quality.
- The experimental results have shown that the following methods have only a minor effect on the alignment quality:
  - adding entries of a conventional dictionary to the training data,
  - making the alignment models dependent on word classes as in Model 4 and Model 5,
  - increasing the number of alignments used in the approximation of the EM algorithm for the fertility-based alignment models.

Further improvements in producing better alignments are expected by adopting cognates [Simard & Foster$^+$ 92], and from statistical alignment models based on word groups rather than single words [Och & Tillmann$^+$ 99]. The use of models that explicitly deal with the hierarchical structures of natural language is very promising [Wu 96, Yamada & Knight 01].
We plan to develop structured models for the lexicon, alignment and fertility probabilities using maximum entropy models. This is expected to allow an easy integration of more dependencies, such as a second-order alignment model without running into the problem that the number of alignment parameters gets unmanageably large.
Furthermore, it will be important to verify the applicability of these statistical alignment models to less similar language pairs such as Chinese–English and Japanese–English.

Chapter 5

Monotone Phrase-Based Translation

A fundamental problem of the single-word based alignment models is that word context is not taken into account. The results in Chapter 4 seem to suggest that this is acceptable if these models are used to compute word alignments. Yet, if these models are used for translation, then word context is more important and the language model of the target language alone is not sufficient to decide for the correct word order.

In this section, we present a method for learning phrasal translation pairs and a method for using these phrasal translation pairs to perform monotone phrase-based translation. Compared to the baseline of Model 4, the context of words has a greater influence and local changes in word order from source to target language can be learned explicitly. [1]

5.1 Motivation

Many natural language phenomena go beyond single-word dependencies. For example, German compound words such as *'Feuerwehrauto'*, *'Führerschein'* or *'Druckertreiber'* are translated by two or more English words. In addition, there are nonliteral translations, e.g. *'das wird schwierig'* by *'that will not be easy'*, where a single-word alignment is problematic. Similar problems occur in the case of proverbs, which typically have a completely different translation.

In addition, the translation of prepositions, articles and particles strongly depends on the context. A preposition such as *'of'* has in a standard dictionary 15 different major translations; the word *'off'* has 30 different major translations [Langenscheidt-Redaktion 96]. The correct translation depends mainly on the context in the source language. The single-word based translation models described in Chapter 4 do not represent this context dependence. The language model is not able to sufficiently counteract this translation model deficiency.

In this section, we present methods for extracting and using phrase translation probabilities from the single-word based alignment models. In this thesis, the term *phrase* simply refers to a sequence of words in a text. We use these phrases to describe a very simple and efficient method for performing monotone phrase-based translation with a reasonable translation quality. These phrases will also be the basis of the alignment template approach presented in Chapter 6.

[1] The training and search algorithms of this approach have been implemented by Richard Zens [Zens 02].

INPUT: e_1^I, f_1^J, A
$i_1 := 1$
WHILE $i_1 \leq I$
  $i_2 := i_1$
  WHILE $i_2 \leq I$
    $TP := \{j | \exists i : i_1 \leq i \leq i_2 \wedge A(i,j)\}$
    IF quasi-consecutive(TP)
    THEN $j_1 := \min(TP)$
      $j_2 := \max(TP)$
      $SP := \{i | \exists j : j_1 \leq j \leq j_2 \wedge A(i,j)\}$
      IF $SP \subseteq \{i_1, i_1+1, \ldots, i_2\}$
      THEN $\mathcal{BP} := \mathcal{BP} \cup \{(e_{i_1}^{i_2}, f_{j_1}^{j_2})\}$
        WHILE $j_1 > 0 \wedge \forall i : A(i,j_1) = 0$
          $j'' := j_2$
          WHILE $j'' \leq J \wedge \forall i : A(i,j'') = 0$
            $\mathcal{BP} := \mathcal{BP} \cup \{(e_{i_1}^{i_2}, f_{j_1}^{j''})\}$
            $j'' := j'' + 1$
          $j_1 := j_1 - 1$
OUTPUT: $\mathcal{BP}$

Figure 5.1: Algorithm `phrase-extract` to extract phrases from a word-aligned sentence pair.

5.2 Bilingual Contiguous Phrases

In this section, we present a method that can learn relationships between whole phrases of n source language words to m target language words. This algorithm, which will be called `phrase-extract`, takes as input a general word alignment matrix (Section 4.4). Hence, we are not restricted to one-to-many alignments but can use the refined methods for combining word alignments. The output is a set of bilingual phrases.
In the following, we describe the criterion that defines the set of phrases that *is consistent with* the word alignment matrix:

$$\mathcal{BP}(f_1^J, e_1^I, A) = \left\{ \left(f_j^{j+m}, e_i^{i+n}\right) : \forall (i',j') \in A : j \leq j' \leq j+m \leftrightarrow i \leq i' \leq i+n \right\} \quad (5.1)$$

Hence, the set of all bilingual phrases that are consistent with the alignment is constituted by all bilingual phrase pairs where all words within the source language phrase are only aligned to the words of the target language phrase and the words of the target language phrase are only aligned to words of the source language phrase.
These phrases can be computed straightforward by enumerating all possible phrases in one language and checking whether the aligned words in the other language are quasi-consecutive. A quasi-consecutive set of words has to be consecutive with the possible exception of words that are not aligned at all. Figure 5.1 gives an algorithm that computes the phrases. Table 5.1 shows the resulting bilingual phrases containing at least two words up to a length of seven words that result by applying this algorithm to the alignment of Figure 4.1.

Table 5.1: Examples of bilingual phrases obtained with at least two words up to a length of seven words that result by applying the algorithm `phrase-extract` to the alignment of Figure 4.1.

ja ,	yes ,
ja , ich	yes , I
ja , ich denke mal	yes , I think
ja , ich denke mal ,	yes , I think ,
ja , ich denke mal , also	yes , I think , well
, ich	, I
, ich denke mal	, I think
, ich denke mal ,	, I think ,
, ich denke mal , also	, I think , well
, ich denke mal , also wir	, I think , well we
ich denke mal	I think
ich denke mal ,	I think ,
ich denke mal , also	I think , well
ich denke mal , also wir	I think , well we
ich denke mal , also wir wollten	I think , well we plan to
denke mal ,	think ,
denke mal , also	think , well
denke mal , also wir	think , well we
denke mal , also wir wollten	think , well we plan to
, also	, well
, also wir	, well we
, also wir wollten	, well we plan to
also wir	well we
also wir wollten	well we plan to
wir wollten	we plan to
in unserer	in our
in unserer Abteilung	in our department
in unserer Abteilung ein neues Netzwerk	a new network in our department
in unserer Abteilung ein neues Netzwerk aufbauen	set up a new network in our department
unserer Abteilung	our department
ein neues	a new
ein neues Netzwerk	a new network
ein neues Netzwerk aufbauen	set up a new network
neues Netzwerk	new network

Performing a 'simple' maximum likelihood estimation for an arbitrary length phrase-based translation model is problematic because of severe overfitting problems. The optimal phrase translation probability according to a maximum likelihood estimation assigns each training corpus sentence $(\mathbf{f}_s, \mathbf{e}_s)$ the probability $p(\mathbf{f}_s|\mathbf{e}_s) = 1$ and tries to minimize the probabilities for all other phrases. Yet, we would like the training method not only to describe well seen sentences but to generalize to unseen sentences. This is a general problem of 'simple' maximum likelihood estimation: it prefers always the more detailed description of the training data, ignoring the generalization ability of the model.
We define a phrasal translation probability using relative frequency:

$$p(\mathbf{f}|\mathbf{e}) = \frac{N(\mathbf{f}, \mathbf{e})}{N(\mathbf{e})} \tag{5.2}$$

Here, $N(\mathbf{f}, \mathbf{e})$ denotes the count of the event that $\mathbf{f}$ has been seen as a translation of $\mathbf{e}$. If one occurrence of $\mathbf{e}$ has $N > 1$ possible translations then each of them contributes to $N(\mathbf{f}, \mathbf{e})$ with $1/N$.
We would like to mention that in principle, `phrase-extract` could be extended to also handle nonconsecutive phrases in source and target language. Informal experiments have shown that allowing for nonconsecutive phrases significantly increases the number of extracted phrases and increases the fraction of wrong phrases. Therefore, we consider only consecutive phrases.

5.3 Example-Based MT with Bilingual Phrases

In this section, we describe a method for performing example-based MT using the bilingual phrases obtained with `phrase-extract`.
In a first step, we introduce the hidden variable B that denotes a segmentation of the sentences f_1^J, e_1^I into a sequence of K phrases in source and target language: $\tilde{f}_1^K, \tilde{e}_1^K$

$$\begin{aligned} Pr(f_1^J|e_1^I) &= \sum_B Pr(B|e_1^I) \cdot Pr(f_1^J|B, e_1^I) && (5.3) \\ &= \alpha(e_1^I) \sum_B Pr(\tilde{f}_1^K|\tilde{e}_1^K) && (5.4) \end{aligned}$$

Here, we assume that all possible segmentations have the same probability $\alpha(e_1^I)$.
In the next step, we allow only monotone phrase alignments to obtain an even simpler translation model that yields a very efficient search implementation. We assume that the phrase $\tilde{f}_1$ is produced by the phrase $\tilde{e}_1$, the phrase $\tilde{f}_2$ is produced by the phrase $\tilde{e}_2$, and so on:

$$Pr(\tilde{f}_1^K|\tilde{e}_1^K) = \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \tag{5.5}$$

No reordering of phrases is performed, only within the phrases reordering is possible. Assuming a bigram language model and using the Bayes decision rule (Eq. 1.2), we obtain the following

search criterion:

$$\begin{aligned}
\hat{e}_1^I &= \operatorname*{argmax}_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} && (5.6)\\
&= \operatorname*{argmax}_{e_1^I} \left\{ \alpha(e_1^I) \prod_i p(e_i|e_{i-1}) \sum_B \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \right\} && (5.7)\\
&\approx \operatorname*{argmax}_{e_1^I, B} \left\{ \prod_i p(e_i|e_{i-1}) \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \right\} && (5.8)
\end{aligned}$$

In Eq. 5.8, we omit the dependence on the normalization constant $\alpha(e_1^I)$ and apply the maximum approximation with respect to B.
This maximization problem can be efficiently solved by using dynamic programming. We define the quantity $Q(j, e)$ for the maximal probability of a sequence of phrases covering the first j source language words whose translation ends with the word e. The quantity $Q(J+1, \$)$ specifies the maximal probability for the optimal translation. With respect to the optimization problem in Eq. 5.8, the variable j in $Q(j, e)$ is enough to represent the translation model state, because knowing j uniquely defines the starting point of the following phrase. In addition, the word e in $Q(j, e)$ completely represents the language model state, because knowing e completely specifies the bigram probability of the following word.
We obtain the following dynamic programming recursion:

$$\begin{aligned}
Q(0, \$) &= 1 && (5.9)\\
Q(j, e) &= \max_{j': j' < j, e', e_1^L : e_L = e} \left\{ Q(j', e') \cdot p(f_{j'+1}^{j}|e_1^L) \cdot \prod_{l=2}^{L} p(e_l|e_{l-1}) \cdot p(e_1|e') \right\} && (5.10)\\
Q(J+1, \$) &= \max_{e'} \{Q(J, e') \cdot p(\$|e')\} && (5.11)
\end{aligned}$$

Storing for each used (j, e) pair the maximizing arguments, we can extract after performing this maximization also the corresponding sequence of words. This method is referred to later on as `PBMonTrans`.
We see that this algorithm has a worst-case search complexity of $O(J^2 \cdot |\{e\}| \cdot |\{\tilde{e}\}|)$. Here, $|\{e\}|$ denotes the vocabulary size and $|\{\tilde{e}\}|$ denotes the number of target language phrases. Using efficient data structures and taking into account that not all possible target language phrases can occur in translating a specific source language sentence, we can perform a very efficient search. This algorithm can be seen as a straightforward extension of the monotone search algorithm [Tillmann & Vogel$^+$ 97b].
If we perform the log-linear combination of language and a direct translation model of Eq. 1.6 instead of the source–channel approach, we only have to change $p(\tilde{f}_k|\tilde{e}_k)$ into $p(\tilde{e}_k|\tilde{f}_k)$. This leads then to the same functional form and the same search algorithm can be used.
Translation results for this translation model shall be described in Chapter 8.

Chapter 6

Alignment Templates

A general deficiency of the single-word based alignment models of Chapter 4 is that they ignore the word context. A first countermeasure was introduced by the monotone phrase-based translation method in Chapter 5. Yet, this model has some obvious weaknesses. First, the used training algorithm very often assigns the probability 1 or 0 to a phrase as most phrases are seen only once. We expect that a model that smoothes these probabilities should be able to obtain better translation quality. Second, estimating $p(\tilde{f}|\tilde{e})$ by relative frequency does not generalize to unseen phrases. Knowledge learned for a specific phrase cannot be generalized to similar phrases. A more systematic approach is presented in this chapter by the alignment template approach. This approach combines the advantages of using a refined reordering model as in Model 4 and using whole phrases rather than single words as entities in the translation model.

6.1 Model

To describe the alignment template model in a formal way, we first decompose both the source sentence f_1^J and the target sentence e_1^I into a sequence of phrases ($k = 1, \ldots, K$):

$$f_1^J = \tilde{f}_1^K, \qquad \tilde{f}_k = f_{j_{k-1}+1}, \ldots, f_{j_k} \tag{6.1}$$

$$e_1^I = \tilde{e}_1^K, \qquad \tilde{e}_k = e_{i_{k-1}+1}, \ldots, e_{i_k} \tag{6.2}$$

Formally, this can be done as in Chapter 5 by introducing the hidden variable B, which denotes a segmentation of f_1^J, e_1^I into a sequence of K phrases in source and target language $\tilde{f}_1^K, \tilde{e}_1^K$:

$$Pr(f_1^J|e_1^I) = \sum_B Pr(B|e_1^I) \cdot Pr(f_1^J|B, e_1^I) \tag{6.3}$$

$$= \alpha(e_1^I) \sum_B Pr(\tilde{f}_1^K|\tilde{e}_1^K) \tag{6.4}$$

Again, we assume that all possible segmentations have the same probability $\alpha(e_1^I)$. To avoid notational overhead, we shall omit in the following description of the model for $Pr(\tilde{f}_1^K|\tilde{e}_1^K) = Pr(f_1^J|B, e_1^I)$ an explicit dependence on the segmentation B.

6.1.1 Phrase Level Alignment

To allow possible reordering of phrases, we introduce an alignment on the phrase level $\tilde{a}_1^K$ between the source phrases $\tilde{e}_1^K$ and the target phrases $\tilde{f}_1^K$:

$$\begin{aligned} Pr(\tilde{f}_1^K|\tilde{e}_1^K) &= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K, \tilde{f}_1^K|\tilde{e}_1^K) && (6.5)\\ &= \sum_{\tilde{a}_1^K} Pr(\tilde{a}_1^K|\tilde{e}_1^K) \cdot Pr(\tilde{f}_1^K|\tilde{a}_1^K, \tilde{e}_1^K) && (6.6)\\ &= \sum_{\tilde{a}_1^K} \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_1^{k-1}) \cdot p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}) && (6.7) \end{aligned}$$

For the phrase level alignment, we use a first-order alignment model $p(\tilde{a}_k|\tilde{a}_1^{k-1}) = p(\tilde{a}_k|\tilde{a}_{k-1})$, which is in addition constrained to be a permutation of the K phrases:

$$p(\tilde{a}_k|\tilde{a}_1^{k-1}) = p(\tilde{a}_k|\tilde{a}_{k-1}) \tag{6.8}$$

As for Model 4, the phrase alignment model $p(\tilde{a}_k|\tilde{a}_{k-1})$ does not reflect that the alignment has to cover all source language positions. To obtain a normalized translation model, we have to renormalize over all possible permutations:

$$Pr(\tilde{a}_1^K|\tilde{e}_1^K) = \frac{\prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1})}{\sum_{\tilde{a}_1^K \in \Pi_K} \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1})} \tag{6.9}$$

Here, Π_K denotes the set of all permutations of the numbers $1, \ldots, K$. This renormalization is computationally very expensive. We do not expect an improved translation by doing this renormalization. In addition, we use $Pr(\tilde{a}_1^K|\tilde{e}_1^K)$ as feature of a direct maximum entropy model where normalization is not needed (Section 6.1). Therefore, we do not perform this renormalization. In Section 6.2, we simplify the probability model further by making the alignment model homogeneous as for the Hidden Markov alignment model (Section 4.2.2).

6.1.2 Word Level Alignment: Alignment Templates

In the following, we suggest a different way to estimate the phrase translation probability. The key elements of the new translation model are the *alignment templates*. An alignment template z is a triple $(F_1^{J'}, E_1^{I'}, \tilde{A})$, which describes the alignment $\tilde{A}$ between a source class sequence $F_1^{J'}$ and a target class sequence $E_1^{I'}$. If each word corresponds to one class, an alignment template corresponds to a bilingual phrase together with an alignment within this phrase.
Figure 6.1 shows examples of alignment templates.
The alignment $\tilde{A}$ is represented as a matrix with $J' \cdot (I' + 1)$ elements and binary values. A matrix element with value 1 means that the words at the corresponding positions are aligned and the value 0 means that the words are not aligned. If a source word is not aligned to a target word, then it is aligned to the empty word e_0, which shall be at the imaginary position $i = 0$. This alignment representation is a generalization of the baseline alignments described in [Brown & Della Pietra$^+$ 93b] and allows for many-to-many alignments.

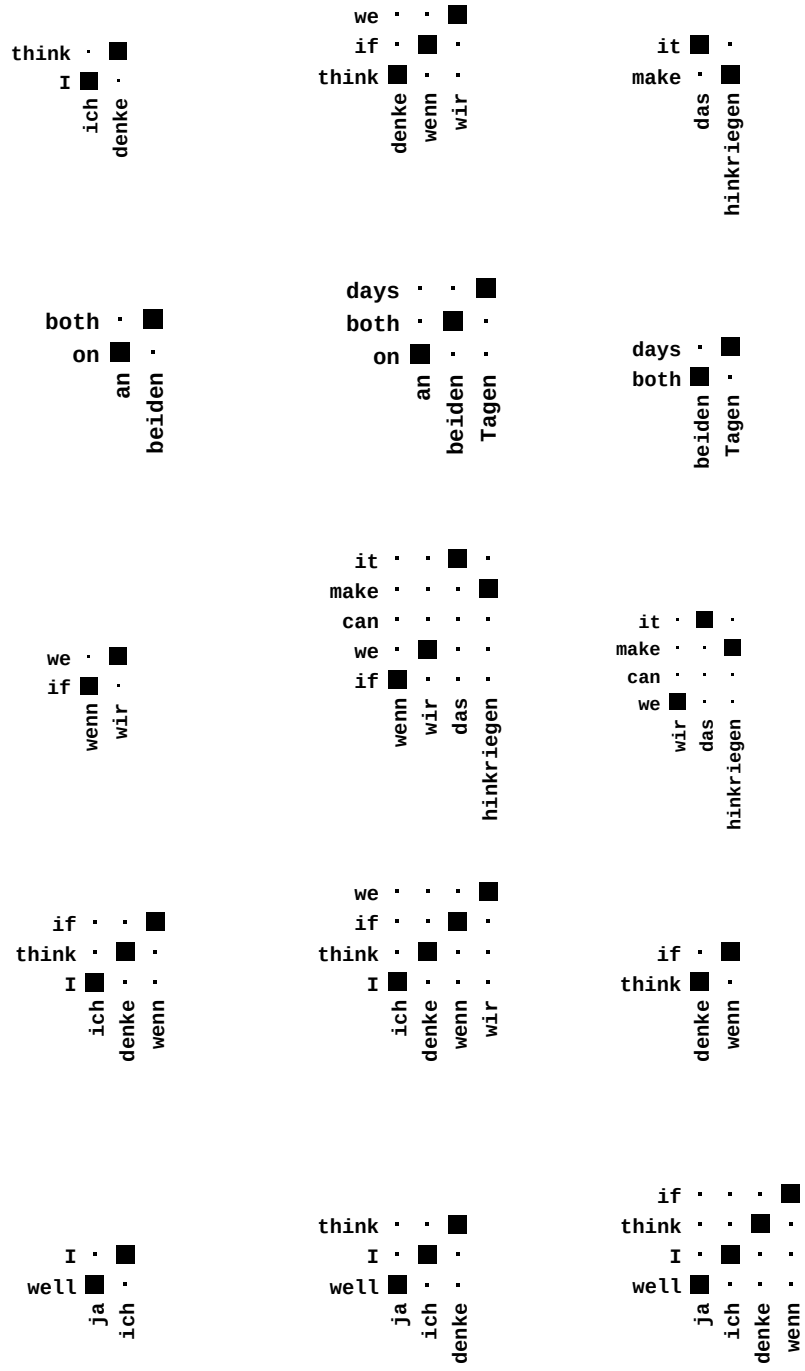


Figure 6.1: Examples of alignment templates obtained in training.

The classes used in $F_1^{J'}$ and $E_1^{I'}$ are automatically trained bilingual classes using the method described in Chapter 7 and constitute a partition of the vocabulary of source and target language. In the following, we use the class function C to map words to their classes. The use of classes

instead of the words themselves has the advantage of a better generalization. For example, if there exist classes in source and target language that contain all town names, an alignment template learned using a specific town can be generalized to all town names.
Formally, the alignment template, denoted by the variable z, is introduced as a hidden variable of the phrase translation probability $p(\tilde{f}|\tilde{e})$:

$$p(\tilde{f}|\tilde{e}) = \sum_z p(z|\tilde{e}) \cdot p(\tilde{f}|z,\tilde{e}) \tag{6.10}$$

Hence, we have to estimate two probabilities. The probability $p(z|\tilde{e})$ to apply an alignment template and the probability $p(\tilde{f}|z,\tilde{e})$ to use an alignment template.
First, we describe the model for the probability $p(\tilde{f}|z,\tilde{e})$. We define that an alignment template $z = (F_1^{J'}, E_1^{I'}, \tilde{A})$ *is applicable* to a sequence of source words $\tilde{f}$, if the alignment template classes and the classes of the source words are equal: $C(\tilde{f}) = F_1^{J'}$. The application of the alignment template z constrains the target words $\tilde{e}$ to correspond to the target class sequence $C(\tilde{e}) = E_1^{I'}$:

$$\begin{aligned}
p(\tilde{f}|z,\tilde{e}) &= p(\tilde{f} = f_1^J | z = (F_1^{J'}, E_1^{I'}, \tilde{A}), \tilde{e} = e_1^I) && (6.11)\\
&= \delta(C(e_1^I), E_1^{I'}) \cdot \prod_{j=1}^{J'} p(f_j|\tilde{A}, e_1^I) \cdot \rho(z, e_1^I)^{1-\delta(C(f_j),F_j)} && (6.12)\\
&\approx \delta(C(e_1^I), E_1^{I'}) \delta(C(f_1^J), F_1^{J'}) \cdot \prod_{j=1}^{J} p(f_j|\tilde{A}, e_1^I) && (6.13)
\end{aligned}$$

To obtain a normalized phrase-based translation model in Eq. 6.12, the function $\rho(z, e_1^I)$ has to be adjusted such that $\sum_{\tilde{f}} p(\tilde{f}|z,\tilde{e}) = 1$ holds. Avoiding this renormalization and setting $\rho(z, e_1^I) = \rho \to 0$, we obtain the deficient probability distribution for $p(\tilde{f}|z,\tilde{e})$ in Eq. 6.13. The effect of this model is to obtain a smoothed version of the 'hard' phrase translation model designed in Chapter 5.
For $p(f_j|\tilde{A}, e_1^I)$, we assume a mixture alignment between the source and target language words constrained by the alignment matrix $\tilde{A}$. A simple method for structuring the single-word probability $p(f_j|\tilde{A}, e_1^I)$ is the following:

$$\begin{aligned}
p(f_j|\tilde{A}, e_1^I) &= \sum_{i=0}^{I} p(i|j;\tilde{A}) \cdot p(f_j|e_i) && (6.14)\\
p(i|j;\tilde{A}) &= \frac{\tilde{A}(i,j)}{\sum_i \tilde{A}(i,j)} && (6.15)
\end{aligned}$$

A disadvantage of this model is that the word order is ignored in the translation model. The translations *'the day after tomorrow'* or *'after the day tomorrow'* for the German word *'übermorgen'* receive an identical contribution. Yet, the first one should obtain a significantly higher probability. Therefore, we include a dependence on the word positions in the lexicon model $p(f|e,i,j)$:

$$p(f_j|\tilde{A}, e_1^I) = \sum_{i=0}^{I} p(i|j;\tilde{A}) \cdot p(f_j|e_i, \sum_{i'=1}^{i-1} \tilde{A}(i',j), \sum_{j'=1}^{j-1} \tilde{A}(i,j')) \tag{6.16}$$

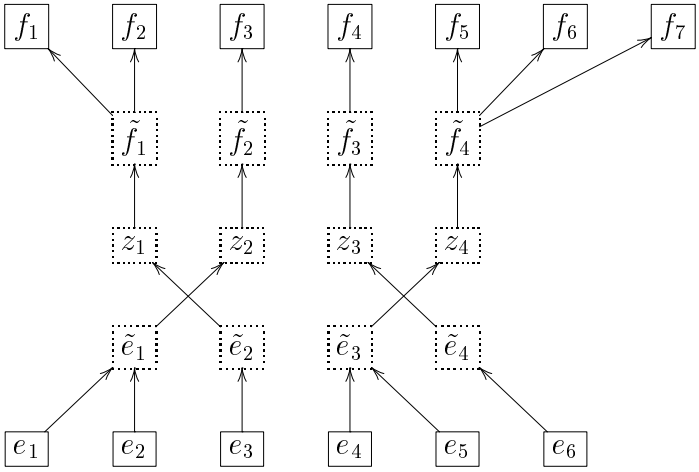


Figure 6.2: Dependencies within the alignment template model.

This model distinguishes the positions within a phrasal translation. The number of parameters of $p(f|e,i,j)$ is significantly higher than $p(f|e)$ alone. Hence, we might run into a data estimation problem especially for words that rarely occur. Performing a linear interpolation of both models with an interpolation parameter α_{L}, we try to avoid this problem:

$$p(f_j|\tilde{A},e_1^I) = \sum_{i=0}^{I} p(i|j;\tilde{A}) \cdot \quad (\quad \alpha_{\mathrm{L}} \cdot p(f_j|e_i, \sum_{i'=1}^{i-1} \tilde{A}(i',j), \sum_{j'=1}^{j-1} \tilde{A}(i,j')) \tag{6.17}$$

$$+ \quad (1-\alpha_{\mathrm{L}}) \cdot p(f_j|e_i) \quad) \tag{6.18}$$

Figure 6.2 gives an overview on the decisions taken in the alignment template model. First, the source sentence words are grouped to phrases. These phrases are reordered and for each phrase an alignment template z is chosen. Then, every phrase $\tilde{f}$ produces its translation $\tilde{e}$. Finally, the sequence of phrases $\tilde{e}_1^K$ constitutes the sequence of words e_1^I.

6.2 Training

This section describes the methods used to train the parameters of our translation model by using a parallel training corpus:

1. We compute for each sentence in the training corpus a word alignment matrix using one of the methods described in Section 4.4.

2. We use this word alignment matrix to estimate a lexicon probability $p(f|e)$ by relative frequencies:

$$p(f|e) = \frac{N_A(f,e)}{N(e)} \tag{6.19}$$

   Here, $N_A(f,e)$ is the frequency that the word f is aligned to the word e and $N(e)$ is the frequency of word e in the training corpus. Similarly, we estimate a position-dependent lexicon model $p(f|e,i,j)$ by relative frequency.

3. We determine word classes for source and target language. A naive approach for doing this would be the use of monolingually optimized word classes in source and target language. Unfortunately, we cannot expect that there is a direct correspondence between independently optimized classes. We determine correlated bilingual classes by using the method described in Chapter 7. The basic idea of this method is to apply a maximum likelihood approach to the joint probability of the parallel training corpus. The resulting optimization criterion for the bilingual word classes is similar to the one used in monolingual maximum likelihood word clustering.

4. To train the probability to apply an alignment template $p(z = (F_1^{J'}, E_1^{I'}, \tilde{A})|\tilde{e})$, we use an extended version of the method `phrase-extract` from Chapter 5. All bilingual phrases that are consistent with the alignment are extracted together with the alignment within this bilingual phrase. Thus, we obtain a count $N(z)$ of how often an alignment template occurred in the aligned training corpus. The probability of using an alignment template is estimated by relative frequency:

$$p(z = (F_1^{J'}, E_1^{I'}, \tilde{A})|\tilde{e}) = \frac{N(z) \cdot \delta(E_1^{I'}, C(\tilde{e}))}{N(C(\tilde{e}))} \tag{6.20}$$

   To reduce the memory requirement of the alignment templates, we compute these probabilities only for phrases of a certain maximal length in the source language. Depending on the size of the corpus, this maximal length is in the experiments between four and seven words.

   In addition, we remove alignment templates that have a probability lower than a certain threshold. In the experiments, we use a threshold of 0.01.

5. For the alignment probabilities $p(\tilde{a}_k|\tilde{a}_{k-1})$, we use a model that takes into account only the distance of the two phrases:

$$p(\tilde{a}_k|\tilde{a}_{k-1}) \propto p(i_{\tilde{a}_k - 1} - i_{\tilde{a}_{k-1}})$$

   Using as additional simplification a log-linear dependence on the distance, we obtain the following model:

$$p(\tilde{a}_k|\tilde{a}_{k-1}) = p_0^{|i_{\tilde{a}_k - 1} - i_{\tilde{a}_{k-1}}|} \tag{6.21}$$

   Hence, we have only one alignment parameter p_0, which is optimized on held-out data. In Section 6.5, we shall show how this parameter can be trained discriminatively. In addition, this model allows the development of a tight heuristic function in Section 6.4.

6. The interpolation parameter for the lexicon model α_{L} are trained using parameter tuning on held-out data.

6.3 Search

In this section, we describe an efficient search architecture for the alignment template model.

6.3.1 General Concept

In general, the search problem for statistical MT even using only Model 1 allowing arbitrary reordering is NP-complete [Knight 99a]. Therefore, we cannot expect to develop efficient search algorithms that guarantee to solve the search problem without search errors. Yet, for practical applications it results to be acceptable to commit some search errors (Section 8.1.2). Hence, the art of developing a search algorithm lies in finding suitable approximations and heuristics that allow an efficient search without performing too many search errors.
In the development of the search algorithm described in this section, we pursue in particular the following aims:

1. The search algorithm should be efficient. It should be possible to translate a sentence with a reasonable length within a few seconds of computing time.

2. It should be possible to reduce the number of search errors by increasing computing time. In the limit, it should be possible to perform search without search errors. The search algorithm should not impose any principal limitations.

3. The search algorithm should be able to handle even long sentences with more than a hundred words with an acceptable computing time.

To meet these aims, search has to be restricted. We do this by performing a breadth-first search with pruning: beam search. In pruning, we constrain the beam only to those nodes that have a probability similar to the highest probability node in the beam. We compare in beam search those hypotheses that cover different parts of the input sentence. This makes the comparison of the probabilities problematic. Therefore, we integrate an admissible estimation of the remaining probabilities to arrive at a complete translation. Details of the heuristic function for the alignment templates are described in Section 6.4.
Many of the other search approaches suggested in the literature do not meet the described aims:

- Both, optimal A* search [Och & Ueffing$^+$ 01] and optimal integer programming [Germann & Jahr$^+$ 01] for statistical MT do not allow efficient search for long sentences. Since the search problem is NP complete, we cannot expect to obtain an efficient optimal search, in principle.

- Greedy search algorithms [Berger & Brown$^+$ 94, Wang 98, Germann & Jahr$^+$ 01] typically commit severe search errors [Germann & Jahr$^+$ 01] and they do not seem to meet goal 2 that search errors can be significantly reduced by increasing computing time.

- Other approaches to solve the search problem obtain polynomial time algorithms by assuming monotone alignments [Tillmann & Vogel$^+$ 97a] or imposing a simplified recombination structure [Nießen & Vogel$^+$ 98]. Others make simplifying assumptions about the search space [García-Varea & Casacuberta$^+$ 98, García-Varea & Och$^+$ 01] or reduce the amount of possible reordering [Wu 96]. All these simplifications ultimately make the search problem simpler, but introduce fundamental search errors.

6.3.2 Search Problem

If we insert the alignment template model and a standard left-to-right language model in the source–channel approach (Eq. 1.2), we obtain the following search criterion in maximum approximation:

$$\begin{aligned}
\hat{e}_1^I &= \operatorname*{argmax}_{e_1^I} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} && (6.22)\\
&= \operatorname*{argmax}_{K, e_1^I = \tilde{e}_1^K, \tilde{f}_1^K, \tilde{a}_1^K \in \Pi_k, z_1^K} && (6.23)\\
&\quad \left\{ \prod_{i=1}^{I} p(e_i|e_{i-1}) \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}) \cdot p(z_k|\tilde{e}_{\tilde{a}_k}) \cdot p(\tilde{f}_k|z_k, \tilde{e}_{\tilde{a}_k}) \right\} && (6.24)
\end{aligned}$$

Here, we used a bigram language model. Obviously, we could also use language models with longer contexts. In the experiments described in Chapter 8, we shall allow the influence of long contexts by using a five-gram language model. Figure 6.3 shows a graphical representation of the dependencies occurring in Eq. 6.24.
If we insert the alignment template model in the alternative decision rule of Eq. 1.6 performing a log-linear combination of a direct translation model and a language model, we obtain:

$$\begin{aligned}
\hat{e}_1^I &= \operatorname*{argmax}_{e_1^I} \{ Pr(e_1^I) \cdot Pr(e_1^I|f_1^J) \} && (6.25)\\
&= \operatorname*{argmax}_{K, e_1^I = \tilde{e}_1^K, \tilde{f}_1^K, \tilde{a}_1^K \in \Pi_K, z_1^K} && (6.26)\\
&\quad \left\{ \prod_{i=1}^{I} p(e_i|e_{i-1}) \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}) \cdot p(z_k|\tilde{f}_{\tilde{a}_k}) \cdot p(\tilde{e}_k|z_k, \tilde{f}_{\tilde{a}_k}) \right\} && (6.27)\\
&= \operatorname*{argmax}_{K, e_1^I = \tilde{e}_1^K, \tilde{f}_1^K, \tilde{a}_1^K \in \Pi_K, z_1^K} && (6.28)\\
&\quad \left\{ \prod_{i=1}^{I} p(e_i|e_{i-1}) \cdot \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}) \cdot p(z_k|\tilde{f}_{\tilde{a}_k}) \prod_{i'=1}^{I(z_k)} p(\tilde{e}_{k,i'}|z_k, \tilde{f}_{\tilde{a}_k}) \right\} && (6.29)
\end{aligned}$$

In Eq. 6.29, the translation model factorizes over the English words. This has advantages for developing a better scoring of search hypotheses leading to a more effective pruning.

6.3.3 Structure of Search Graph

We have to structure the search space in a suitable way to perform an efficient search. The used search algorithm has a search organization along the positions of the target language sentence. Hence, in the search process, we generate search hypotheses that correspond to prefixes of a hypothetical translation of a source language sentence. A partial hypothesis is extended by appending one target word.
In a first step, we determine the set of all source phrases in f_1^J for which an applicable alignment template exists. Every possible application of an alignment template $z = (F_1^{J'}, E_1^{I'}, \tilde{A})$ to a subsequence $f_j^{j+J'-1}$ of the source sentence is called *alignment template instantiation* $Z = (z, j)$. Hence, the set of all alignment template instantiations for the source sentence f_1^J is:

$$\left\{ Z = (z, j) | z = (F_1^{J'}, E_1^{I'}, \tilde{A}) \wedge \exists j : p(z|f_j^{j+J-1}) > 0 \right\} \quad (6.30)$$

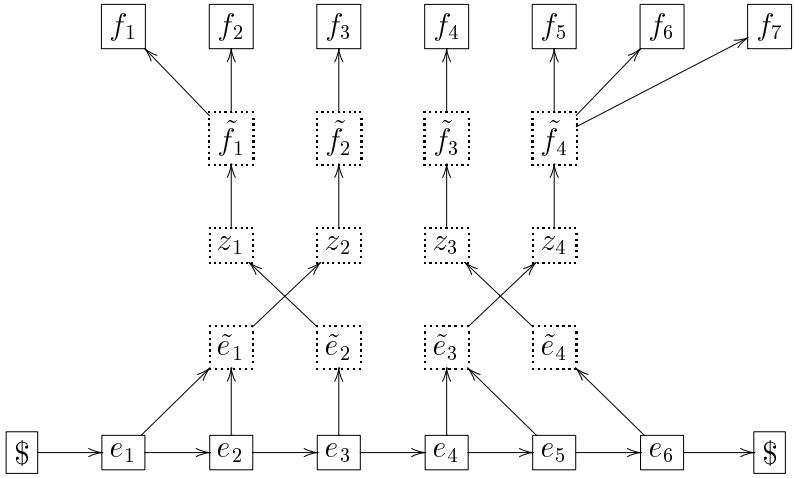


Figure 6.3: Dependencies of the combination of a left-to-right language model and the alignment template model.

If words occur that have not been seen before, we introduce a new alignment template that performs a one-to-one translation of this word by itself. This works well if the unknown word is a proper name because then typically the identity translation is correct.
In the second step, we determine a set of probable target language words for each target word position in the alignment template instantiation. Only these words are then hypothesized in search. We call this selection of highly probable words *observation pruning* [Tillmann & Ney 00]. As criterion for a word e at position i in the alignment template instantiation, we use:

$$\delta(E_i, C(e)) \cdot \sum_{j=0}^{J'} \frac{\tilde{A}(i,j)}{\sum_{i'} \tilde{A}(i',j)} \cdot p(e|f_j) \tag{6.31}$$

Only the N_{obs} best scoring words are hypothesized in search. A value $N_{obs} = 5$ typically gives good results.
A partial hypothesis is extended by appending one target word. The set of all partial hypotheses can be structured as a graph with a source node representing the sentence start, goal nodes representing complete translations and intermediate nodes representing partial hypotheses.
The edges of the search graph are the decisions for a specific target language word. A decision is a triple $d = (Z, e, k)$ consisting of an alignment template instantiation Z, the generated word e and the index of the generated word in Z. A hypothesis n corresponds to a valid sequence of decisions d_1^i. There are the following types of decisions:

1. Start a new alignment template: $d_i = (Z_i, e_i, 1)$. In this case, the index $k = 1$. This decision can only be made if the previous decision d_{i-1} finished an alignment template and if the newly chosen alignment template instantiation does not overlap with any previously chosen alignment template instantiation.

   The probability of this decision is:

   $$q(d_{i-1} \rightarrow d_i) = p(e_i|e_{i-1}) \cdot p(\tilde{a}(Z_i)|\tilde{a}(Z_{i-1})) \cdot p(Z_i) \tag{6.32}$$

   Here, $p(Z_i)$ denotes the probability of using this alignment template instantiation and $\tilde{a}(Z)$ denotes the (phrase) alignment of the alignment template instantiation Z.

2. Extend an alignment template: $d_i = (Z_i, e_i, k)$. This decision can only be made if the previous decision uses the same alignment template instantiation and has as index $k-1$: $d_{i-1} = (Z_i, e_{i-1}, k-1)$.

   The probability of this decision is:

$$q(d_{i-1} \rightarrow d_i) = \begin{cases} p(e_i|e_{i-1}) & \text{if } k < I(Z_i) \\ p(e_i|e_{i-1}) \cdot p(f_{j(Z_i)}^{j(Z_i)+J(Z_i)-1}|e_{i-I(Z_i)+1}^{i}, z(Z_i)) & \text{otherwise} \end{cases} \tag{6.33}$$

   Here, $p(f_{j(Z_i)}^{j(Z_i)+J(Z_i)-1}|e_{i-I(Z_i)+1}, z(Z_i))$ denotes the probability of using this alignment template instantiation.

3. Finishing the translation of a sentence: $d_i = (\$, \$, 0)$. In this case, the hypothesis is marked as a goal hypothesis. This decision is only possible if the previous decision d_{i-1} finished an alignment template and if the used alignment template instantiations completely cover the input sentence.

   The probability of this decision is:

$$q(d_{i-1} \rightarrow d_i) = p(\$|e_{i-1}) \tag{6.34}$$

   Here, \$ denotes the sentence end symbol.

Any valid and complete sequence of decisions d_1^{I+1} uniquely corresponds to a certain translation e_1^I, a segmentation into K phrases, a phrase alignment $\tilde{a}_1^K$ and a sequence of alignment template instantiations z_1^K. The product of the decision probabilities is equal to the probability described in Eq. 6.24.

A straightforward representation of all hypotheses would be the prefix tree of all possible sequences of decisions. Obviously, there would be a large redundancy in this representation of the search space because there are many search nodes that are indistinguishable in the sense that the sub-tree following these search nodes are identical. For these identical search nodes, we only have to maintain the most probable hypothesis. This is the concept of recombination [Bellman 57].

In general, the criterion to perform recombination of a set of nodes is that the hypotheses cannot be distinguished by neither language nor the translation model. Doing recombination, we obtain a search graph instead of a search tree. The exact criterion to perform recombination for the alignment templates shall be described in Section 6.3.5.

6.3.4 Search Algorithm

Theoretically, we could use any graph search algorithm to search the optimal path in the search graph. We use a breadth-first search algorithm with pruning. This approach offers very good possibilities to adjust the optimal trade-off between quality and efficiency. In pruning, we always compare hypotheses that have produced the same number of target words.

Figure 6.4 shows a structogramm of the used algorithm. Since the search space increases exponentially, the whole search graph cannot be represented explicitly. Therefore, we use an implicit representation of the search graph, which is performed by the functions `Extend` and `Recombine`. The function `Extend` produces all hypotheses, which can be reached by extending the

INPUT: implicitly defined search graph (functions `Recombine`, `Extend`)
$H = \{$empty-hypothesis$\}$
WHILE $H \neq \emptyset$
$H_{ext} := \emptyset$
FOR $n \in H$
IF hypothesis n is final
THEN $H_{fin} := H_{fin} \cup \{n\}$
ELSE $H_{ext} := H_{ext} \cup \texttt{Extend}(n)$
$H := \texttt{Recombine}(H_{ext})$
$\hat{Q} = \max_{n \in H} Q(n)$
$H := \{n \in H : Q(n) > t_p \cdot \hat{Q}\}$
$H :=$ HistogramPruning(H, N_p)
$\hat{n} = \text{argmax}_{n \in H_{fin}} \; Q(n)$
OUTPUT: $\hat{n}$

Figure 6.4: Algorithm to perform a breadth-first search with pruning for alignment templates.

current hypothesis by one word. Some hypotheses might be identical or indistinguishable by language and translation model. These are recombined in the step `Recombine`. We expand the search graph such that only hypotheses with the same number of target language words are recombined.
In the pruning step, we use two different types of pruning. First, we perform pruning relative to the probability of the current best hypothesis $\hat{Q}$. We ignore all hypotheses that have a probability lower than $t_p \cdot \hat{Q}$. This type of pruning can be performed already when the hypothesis extensions are computed. In histogram pruning [Steinbiss & Tran$^+$ 94], we maintain only the best N_p hypotheses. The two pruning parameters t_p and N_p have to be optimized with respect to the optimal trade-off between efficiency and quality.

6.3.5 Implementation

In this section, we describe various issues to perform an efficient implementation of a search algorithm for the alignment template approach.

Search hypothesis representation

A very important design decision in the implementation is the representation of a hypothesis. Theoretically, it would be possible to represent search hypotheses only by the associated decision and a backpointer to the previous hypothesis. Yet, this would be a very inefficient representation for the implementation of the operations that have to be performed in search. The hypothesis representation should contain all information to perform efficiently the computations needed in the search, but should not contain more information to keep the memory consumption small.
In search, we produce partial hypotheses n, each of which contains the following information:

1. e: the final target word produced,

2. h: the state of the language model,

3. $\mathbf{c} = c_1^J$: the coverage vector representing the already covered positions of the source sentence ($c_j = 1$ means the position j is covered, c_j=0 the position j is not covered),

4. Z: a reference to the alignment template instantiation, which produced the final target word,

5. k: the position of the final target word in the alignment template instantiation,

6. $Q(n)$: the accumulated probability of all previous decisions,

7. n': a reference to the previous partial hypothesis.

Using this representation, we can perform the following operations very efficiently:

- Comparing if a specific alignment template instantiation can be used to extend a hypothesis. To do that, we check if the positions of the alignment template instantiation are still free in the hypothesis coverage vector.

- Checking if a hypothesis is final. To do that, we check if the coverage vector contains no uncovered position. If the coverage vector is internally represented as bit vector, the corresponding operation can be implemented very efficiently.

- Checking if two hypotheses extensions can be recombined. The criterion to recombine two hypotheses $n_1 = (e_1, h_1, \mathbf{c}_1, Z_1, k_1)$ and $n_2 = (e_2, h_2, \mathbf{c}_2, Z_2, k_2)$ is:

  $$\begin{array}{lll} h_1 \oplus e_1 = h_2 \oplus e_2 \wedge & & \text{identical language model state} \\ \mathbf{c}_1 = \mathbf{c}_2 \wedge & & \text{identical coverage vector} \\ (\quad (Z_1 = Z_2 \wedge k_1 = k_2) \vee & & \text{alignment template instantiation is identical} \\ \quad\; (J(Z_1) = k_1 \wedge J(Z_2) = k_2) &) & \text{alignment template instantiation finished} \end{array}$$

  Here, $h \oplus e$ denotes the new language model state, which is obtained if the word e is used to extend the language model state h.

We compare in beam search those hypotheses that cover different parts of the input sentence. This makes the comparison of the probabilities problematic. Therefore, we integrate an admissible estimation of the remaining probabilities to arrive at a complete translation. Details of the heuristic function for the alignment templates are described in Section 6.4.

Efficient search

We discuss in the following various methods that significantly speed up search efficiency.
A significantly faster search is obtained using the direct search criterion of Eq. 6.29 instead of the Bayes approach. Here, both language and translation model predict target language words e. Figure 6.5 shows a graphical representation of the resulting dependencies. Hence, for each extension, we can directly compute the translation model contribution. The probability to extend a hypothesis with one target language word can then directly include the translation

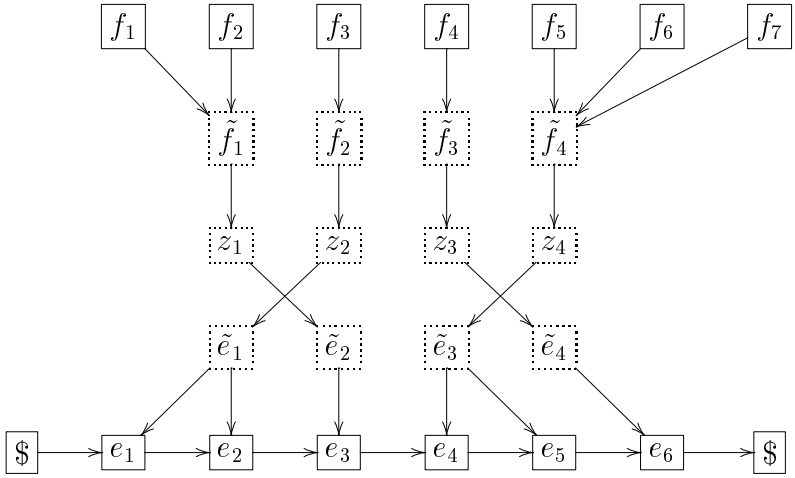


Figure 6.5: Dependencies of a log-linear combination of a left-to-right language model and the direct alignment template translation model.

model contribution. This allows a more efficient pruning of hypotheses. The probabilities to start a new alignment template (Eq. 6.32) is:

$$q(d_{i-1} \rightarrow d_i) = p(e_i|e_{i-1}) \cdot p(\tilde{a}(Z_i)|\tilde{a}(Z_{i-1})) \cdot p(Z_i) \cdot p(e_i|z(Z_i), f_{j(Z_i)}^{j(Z_i)+J(Z_i)-1}) \quad (6.35)$$

The probability to extend an alignment template (Eq. 6.33) is:

$$q(d_{i-1} \rightarrow d) = p(e_i|e_{i-1}) \cdot p(e_i|z(Z_i), f_{j(Z_i)}^{j(Z_i)+J(Z_i)-1}) \quad (6.36)$$

An even more efficient search can be obtained when a segmentation of the input sentence is performed beforehand. This can be done by determining the sequence of phrases for which the most probable alignment templates exist. We search for a sequence of phrases $\tilde{f}_1 \circ \cdots \circ \tilde{f}_k = f_1^J$ with:

$$\underset{\tilde{f}_1 \circ \cdots \circ \tilde{f}_k = f_1^J}{\text{argmax}} \left\{ \prod_{k=1}^{K} \max_z \; p(z|\tilde{f}_k) \right\} \quad (6.37)$$

This is computed efficiently by dynamic programming. This approximation might be useful in applications where very efficient search is important. All results in this thesis are obtained without performing this approximation.

An additional important element in the search algorithm is an efficient and early garbage collection of those hypotheses that are pruned away. This is important to keep the dynamic memory requirement as small as possible. We use a garbage collection algorithm based on reference counting. Each hypothesis contains an additional integer that counts the number of pointers and back-pointers that refer to this hypothesis. If this count reaches zero the hypothesis can be freed. Hence, unnecessary search hypotheses are removed from the memory as early as possible. In C++, this type of garbage collection can be performed efficiently and safe using the concept of *smart pointers* [Koenig & Moo 97].

6.4 Heuristic Function

To improve the comparability of search hypotheses, we introduce heuristic functions. An admissible heuristic function estimates optimistically the probabilities to reach the goal node from

a certain search graph node. For an A*-based search algorithm, the heuristic function is crucial to be able to translate long sentences. For a beam search algorithm, the heuristic function has a different motivation, namely to improve the scoring of search hypotheses. The goal is to make the probabilities of all hypotheses more comparable, to minimize the probability that the hypothesis leading to the optimal translation is pruned away.

Heuristic functions for search in statistical MT have been used in [Wang & Waibel 97] and [Och & Ueffing$^+$ 01]. [Wang & Waibel 97] have described a simple heuristic function for Model 2, which was non admissible. [Och & Ueffing$^+$ 01] have described an admissible heuristic function for Model 4 and an almost-admissible heuristic function which is empirically obtained.

We have to keep in mind that a heuristic function is only helpful if the introduced overhead to compute the heuristic function is over-compensated by the gain obtained using a better pruning of search hypotheses. The heuristic functions described in the following are designed such that their computation can be performed efficiently.

The basic idea for developing a heuristic function for Model 4 (and alignment models in general) is that all source sentence positions, which have not been covered so far, still have to be translated to complete the sentence. Therefore, the value of the heuristic function $H^X(n)$ for a node n can be inferred if we have an estimation $h^X(j)$ of the optimal score of translating position j (here X denotes different possibilities to choose the heuristic function):

$$H^X(n) = \prod_{j \notin c(n)} h^X(j) \tag{6.38}$$

where $c(n)$ is the coverage vector.

The situation in the case of the alignment template approach is more complicated than with Model 4 as not every word is translated alone, but typically the words are translated in context. Therefore, the basic quantity for the heuristic function in the case of the alignment template approach is a function $h(Z)$, which assigns every alignment template instantiation Z a maximal probability. Using $h(Z)$, we can induce a position-dependent heuristic function $h(j)$:

$$h(j) := \max_{Z: j(Z) \leq j \leq j(Z)+J(Z)} h(Z)^{1/J(Z)} \tag{6.39}$$

Here, $J(Z)$ denotes the number of source language words produced by the alignment template instantiation Z and $j(z)$ denotes the position of the first source language word. Now, we show that if $h(Z)$ is admissible, then also $h(j)$ is admissible. We have to show that for all nonoverlapping sequences Z_1^K holds:

$$\prod_{k=1}^{K} h(Z_k) \leq \prod_{j \in c(Z_1^K)} h(j) \tag{6.40}$$

Here, $c(Z_1^K)$ denotes the set of all positions covered by the sequence of alignment templates

Z_1^K. This can be shown easily:

$$\prod_{k=1}^{K} h(Z_k) = \prod_{k=1}^{K} \prod_{j=1}^{J(Z_k)} h(Z_k)^{1/J(Z_k)} \quad (6.41)$$

$$= \prod_{j \in c(Z_1^K)} h(Z_{k(j)})^{1/J(Z_{k(j)})} \quad (6.42)$$

$$\leq \prod_{j \in c(Z_1^K)} \max_{Z: j(Z) \leq j \leq j(Z)+J(Z)} h(Z)^{1/J(Z)} \quad (6.43)$$

In the following, we develop various heuristic functions $h(Z)$ of increasing complexity. The simplest realization of a heuristic function $h(Z)$ takes into account only the prior probability of an alignment template instantiation:

$$H^A(Z) = p(Z) \quad (6.44)$$

The lexicon model can be integrated as follows:

$$H^T(Z) = \prod_{j'=j(Z)}^{j(Z)+J(Z)-1} \max_e \; p(f_{j'}|e) \quad (6.45)$$

The language model can be incorporated by considering that for each target word there exists an optimal language model probability:

$$p^L(e) = \max_{e',e''} \; p(e|e',e'') \quad (6.46)$$

Here, we assume a trigram language model. In general, we have to maximize Eq. 6.46 over all possible different language model histories. We can also combine the language model and the lexicon model into one heuristic function:

$$H^{TL}(Z) = \prod_{j'=j(Z)}^{j(Z)+J(Z)-1} \max_e \; p(f_{j'}|e) \cdot p^L(e) \quad (6.47)$$

Here, $p^L(e)$ denotes the optimal language model probability described in Eq. 6.46.
Including the phrase alignment probability is not possible as straightforward as the alignment model in the single-word based approach where every word position has a contribution [Och & Ueffing$^+$ 01]. Here, we only have a contribution for each phrase. If we make the simplifying assumption that the alignment probability depends only log-linearly on the jump width (Eq. 6.21), we compute the minimum sum of all jump widths that is needed to complete the translation. This sum can be computed efficiently by the algorithm shown in Figure 6.6. This algorithm also obtains as parameter the previously covered position j.
Then, an admissible heuristic function for the jump width is obtained by:

$$H^J(\mathbf{c}, j) = {p_0}^{D(\mathbf{c},j)} \quad (6.48)$$

Combining all the heuristic functions for the various models, we obtain as final heuristic function for a search hypothesis n:

$$H^{ATLJ}(n) = H^J(c(n), j(n)) \cdot \prod_{j \notin c(n)} H^A(j) \cdot H^{TL}(j) \quad (6.49)$$

INPUT: coverage vector c_1^J, previously covered position j
ff $= \min(\{j' \mid c_{j'} = 0\})$
mj $= \lvert j - \text{ff} \rvert$
WHILE ff $\neq (J+1)$
fo $:= \min(\{j' \mid j' > \text{ff} \wedge c_{j'} = 1\})$
ff $:= \min(\{j' \mid j' > \text{fo} \wedge c_{j'} = 0 \vee j' = J+1\})$
mj $:=$ mj $+ \lvert \text{ff} - \text{fo} \rvert$
OUTPUT: mj

Figure 6.6: Algorithm `min-jumps` to compute the minimum number of needed jumps $D(c_1^J, j)$ to complete the translation.

6.5 Maximum Entropy Modeling of Alignment Templates

So far, the statistical translation models have been described in such a way that they can be used in a source–channel approach to statistical MT. Yet, as pointed out in Section 1.3.1, the source–channel approach only guarantees optimal results if we use the true probability distributions for $Pr(f_1^J|e_1^I)$ and $Pr(e_1^I)$. As we can only expect poor approximations of the true probability distributions, we perform the maximum entropy based combination of the available models as suggested in Section 1.3.2. In this section, we describe the used feature functions and the training of the model parameters.

6.5.1 Feature Functions

As suggested in Section 1.3.2, the simplest approach to define feature functions for the maximum entropy model would be the definition of the following two feature functions:

$$h_1(e_1^I, f_1^J) = \log p(e_1^I) \quad (6.50)$$
$$h_2(e_1^I, f_1^J) = \log p(f_1^J|e_1^I) \quad (6.51)$$

Here, $p(e_1^I)$ denotes the trained language model and $p(f_1^J|e_1^I)$ denotes the trained alignment template model. We obtain two maximum entropy model parameters λ_1 and λ_2 that can be trained using the GIS algorithm.

Yet, we use more refined feature functions for each component of the translation model instead of one feature function for the whole translation model $Pr(f_1^J|e_1^I)$. Therefore, the maximum entropy model can consider qualitative differences of the different component models. Taking as component models the different factors in Eq. 6.24, we obtain the following four feature

functions:

$$\begin{aligned}
h_1(e_1^I, f_1^J, K, \tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K, z_1^K) &= \log p(e_1^I) \\
h_2(e_1^I, f_1^J, K, \tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K, z_1^K) &= \log \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}, K) \\
h_3(e_1^I, f_1^J, K, \tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K, z_1^K) &= \log \prod_{k=1}^{K} p(z_k|\tilde{e}_{\tilde{a}_k}) \\
h_4(e_1^I, f_1^J, K, \tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K, z_1^K) &= \log \prod_{k=1}^{K} p(\tilde{f}_k|z_k, \tilde{e}_{\tilde{a}_k})
\end{aligned}$$

As we perform maximum approximation in search, these feature functions depend on the hidden variables of the alignment template model. To simplify the notation, we shall omit in the following the dependence on the hidden variables of the model.
So far, we use the logarithm of the components of a translation model as feature functions. This is a very convenient approach to improve the quality of a baseline system. Yet, we are not limited to train only model scaling factors, but we have many possibilities:

- We could add a sentence length feature:

  $$h(f_1^J, e_1^I) = I$$

  This corresponds to a word penalty for each produced target word.

- We could include additional language models by using features of the following form:

  $$h(f_1^J, e_1^I) = h(e_1^I)$$

- We could use a feature that counts how many entries of a conventional lexicon co-occur in the given sentence pair. Therefore, the weight for the provided conventional dictionary can be learned. The intuition is that the conventional dictionary is expected to be more reliable than the automatically trained lexicon and therefore should obtain a larger weight.

- We could use lexical features, which fire if a certain lexical relationship (f, e) occurs:

  $$h(f_1^J, e_1^I) = \left(\sum_{j=1}^{J} \delta(f, f_j)\right) \cdot \left(\sum_{i=1}^{I} \delta(e, e_i)\right)$$

- We could use grammatical features that relate certain grammatical dependencies of source and target language. For example, using a function $k(\cdot)$ that counts how many verb groups exist in the source or the target sentence, we can define the following feature, which is 1 if each of the two sentences contains the same number of verb groups:

  $$h(f_1^J, e_1^I) = \delta(k(f_1^J), k(e_1^I)) \tag{6.52}$$

  In the same way, we can introduce semantic features such as for example a dependence on the dialogue act of the French and English sentence or a dependence on a semantic classification.

We can use numerous additional features that deal with specific problems of the baseline statistical MT system. Here, we shall use the first three of these features. As additional language model, we use a class-based five-gram language model. This feature and the word penalty feature allow a straightforward integration into a dynamic programming search algorithm. As this is not possible for the conventional dictionary feature, we use n-best rescoring for this feature.

6.5.2 Training with GIS Algorithm

To train the model parameters λ_1^M of the direct translation model according to Eq. 1.11, we use the GIS (Generalized Iterative Scaling) algorithm [Darroch & Ratcliff 72]. It should be noted that, as was already shown by [Darroch & Ratcliff 72], by applying suitable transformations, the GIS algorithm is able to handle any type of real-valued features. To apply this algorithm, we have to solve various practical problems.
The renormalization needed in Eq. 1.8 requires a sum over many possible sentences, for which we do not know an efficient algorithm. Hence, we approximate this sum by sampling the space of all possible sentences by a large set of highly probable sentences. The set of considered sentences is computed by an appropriately extended version of the used search algorithm described in Section 6.3 computing an approximate n-best list of translations.
Using an n-best approximation, we might face the problem that the parameters trained with the GIS algorithm yield worse translation results even on the training corpus. This can happen because with the modified model scaling factors the n-best list can change significantly and can include sentences that have not been taken into account in training. Using these sentences, the new model parameters might perform worse than the old model parameters. To avoid this problem, we proceed as follows. In a first step, we perform search and compute an n-best list and use this n-best list to train the model parameters. Second, we use the new model parameters in a new search and compute a new n-best list, which is combined with the existing n-best list. Third, using this extended n-best list new model parameters are computed. This is iterated until the resulting n-best list does not change. In this algorithm, convergence is guaranteed as in the limit the n-best list will contain all possible translations. In practice, the algorithm converges after about five to seven iterations.
Unlike automatic speech recognition, we do not have one reference sentence, but there exists a number of reference sentences. Yet, the criterion in Eq. 1.11 allows for only one reference translation. Hence, we change the criterion to allow R_s reference translations $\mathbf{e}_{s,1}, \ldots, \mathbf{e}_{s,R_s}$ for the sentence $\mathbf{e}_s$:

$$\hat{\lambda}_1^M = \operatorname*{argmax}_{\lambda_1^M} \left\{ \sum_{s=1}^{S} \frac{1}{R_s} \sum_{r=1}^{R_s} \log p_{\lambda_1^M}(\mathbf{e}_{s,r}|\mathbf{f}_s) \right\}$$

We use this optimization criterion instead of the optimization criterion shown in Eq. 1.11.
In addition, we might have the problem that no single of the reference translations is part of the n-best list because the search algorithm performs pruning, which in principle limits the possible translations that can be produced given a certain input sentence. To solve this problem, we define for maximum entropy training each sentence as reference translation that has the minimal number of word errors with respect to any of the reference translations in the n-best list.

Chapter 7

Bilingual Word Classes

In this chapter, we describe the training of the bilingual word classes that is needed for the alignment templates. First, we review in Section 7.1 a well-known monolingual clustering approach. In Section 7.2, we describe an extension of this approach that is suited for bilingual word classes. Finally, the training algorithm and results are described.

7.1 Monolingual Word Clustering

The task of a statistical language model is to estimate the probability $Pr(w_1^I)$ of a sequence of words $w_1^I = w_1, \ldots, w_I$. A straightforward approximation of $Pr(w_1^I)$ is to model it as a product of bigram probabilities: $Pr(w_1^I) = \prod_{i=1}^{I} p(w_i|w_{i-1})$. If we want to estimate the bigram probabilities $p(w|w')$ using a realistic natural language corpus, we are faced with the problem that most of the bigrams are rarely seen. One method for solving this problem is to partition the set of all words into equivalence classes. The function C maps words w to their classes $C(w)$. Rewriting the corpus probability using classes, we arrive at the following probability model $p(w_1^I|C)$:

$$p(w_1^I|C) := \prod_{i=1}^{I} [p(C(w_i)|C(w_{i-1})) \cdot p(w_i|C(w_i))] \tag{7.1}$$

In this model, we have two types of probabilities: the transition probability $p(C|C')$ for class C given its predecessor class C' and the membership probability $p(w|C)$ for word w given class C.

We are given a training corpus w_1^I which we represent by the counts $N(C, C')$, $N(w, C)$, $N(C)$ and $N(w)$ of bigrams and unigrams. The set of all counts is denoted by $\{N\}$. To determine the

optimal classes $\hat{C}$, we perform a maximum likelihood approach:

$$\begin{aligned}
\hat{C} &= \operatorname*{argmax}_C \; p(w_1^I|C) && (7.2)\\
&= \operatorname*{argmax}_C \left\{ \prod_{i=1}^{I} [p(C(w_i)|C(w_{i-1})) \cdot p(w_i|C(w_i))] \right\} && (7.3)\\
&= \operatorname*{argmax}_C \left\{ \prod_{i=1}^{I} \left[\frac{N(C(w_i), C(w_{i-1}))}{N(C(w_{i-1}))} \cdot \frac{N(w_i)}{N(C(w_i))} \right] \right\} && (7.4)\\
&= \operatorname*{argmax}_C \left\{ \sum_{C,C'} N(C,C') \cdot \log N(C,C') - 2 \sum_{C} N(C) \cdot \log N(C) \right\} && (7.5)\\
&= \arg\max_C \; G_1(C, \{N\}) && (7.6)
\end{aligned}$$

We call the resulting optimization criterion $G_1(C, \{N\})$. In Eq. 7.3, we insert the bigram language model probability. In Eq. 7.4, we estimate the transition and the membership probability by relative frequencies, which directly follows from the used maximum likelihood principle. In Eq. 7.5, we apply the logarithm and change the summation order. Since the optimum is reached if each word is a class of its own, we have to fix the number of classes in C in advance. Therefore, an additional optimization process is necessary to determine the number of classes. The use of leaving-one-out in a modified optimization criterion as in [Kneser & Ney 93] could solve in principle this problem, but for the used corpora this method seems to overestimate the number of classes.

7.2 Bilingual Word Clustering

In bilingual word clustering, we are interested in classes C_f and C_e that form partitions of the vocabulary of two languages. To perform bilingual word clustering, we use a maximum likelihood approach as in the monolingual case. We maximize the joint probability of a bilingual training corpus (e_1^I, f_1^J):

$$\begin{aligned}
(\hat{C_e}, \hat{C_f}) &= \operatorname*{argmax}_{C_e, C_f} \; p(e_1^I, f_1^J|C_e, C_f) && (7.7)\\
&= \operatorname*{argmax}_{C_e, C_f} \left\{ p(e_1^I|C_e) \cdot p(f_1^J|e_1^I; C_e, C_f) \right\} && (7.8)
\end{aligned}$$

To perform the maximization of Eq. (7.8), we have to model the monolingual prior probability $p(e_1^I|C_e)$ and the translation probability $p(f_1^J|e_1^I; C_e, C_f)$. For the first, we use the class bigram probability from Eq. (7.1).
To model $p(f_1^J|e_1^I; C_e, C_f)$, we assume the existence of an alignment a_1^J. We assume that every word f_j is produced by the word e_{a_j} at position a_j in the training corpus with the probability $p(f_j|e_{a_j})$:

$$p(f_1^J|e_1^I) = \prod_{j=1}^{J} p(f_j|e_{a_j}) \qquad (7.9)$$

INPUT: Parallel corpus (e_1^I, f_1^J) and number of classes in C_e and C_f.
Determine the word alignment a_1^J.
Get some initial classes C_e and C_f.
UNTIL convergence criterion is met:
FOR EACH word e:
FOR EACH class E:
Determine the change of $G((C_e, C_f), N_g)$ if e is moved to E.
Move e to the class with the largest improvement.
FOR EACH word f:
FOR EACH class F:
Determine the change of $G((C_e, C_f), N_g)$ if f is moved to F.
Move f to the class with the largest improvement.
OUTPUT: Classes C_e and C_f.

Figure 7.1: Algorithm `bil-word-cluster` to compute bilingual word classes.

By rewriting the translation probability using word classes, we obtain (corresponding to Eq. (7.1)):

$$p(f_1^J|e_1^I; C_e, C_f) = \prod_{j=1}^{J} \left[p(C_f(f_j)|C_e(e_{a_j})) \cdot p(f_j|C_f(f_j))\right] \tag{7.10}$$

The variables F and E denote special classes in C_f and C_e. As in the monolingual case, the maximum likelihood estimate of $p(F|E)$ and $p(f|F)$ are relative frequencies:

$$\begin{aligned} p(F|E) &= N(E,F)/\left(N(E)\right) && (7.11)\\ p(f|F) &= N(f)/\left(N(F)\right) && (7.12)\\ &= N(f)/\left(\sum_E N(E,F)\right) && (7.13) \end{aligned}$$

If we insert these relative frequencies into Eq. (7.10) and apply the same transformations as in the monolingual case, we obtain a similar optimization criterion for the translation probability part of Eq. (7.8). Thus, the final optimization criterion for bilingual word classes is:

$$\begin{aligned} G_2(C_e, C_f, \{N\}) &= \sum_{E,E'} N(E',E)\log N(E',E) + \sum_{E,F} N(E,F)\log N(E,F)\\ &\quad -2\sum_E N(E)\log N(E)\\ &\quad -\sum_F N(F)\log N(F) - \sum_E N(E)\log N(E) && (7.14)\\ (\hat{C}_e, \hat{C}_f) &= \arg\min_{C_e, C_f} G_2((C_e, C_f), N_g) && (7.15) \end{aligned}$$

Another method for performing bilingual word clustering is to apply a two-step approach. First, we determine classes $\hat{C}_e$ optimizing only the monolingual part of Eq. (7.8). Second, we deter-

mine classes $\hat{C}_f$ optimizing the bilingual part (without changing $\hat{C}_e$):

$$\hat{C}_e = \arg\min_{C_e} G_1(C_e, N) \tag{7.16}$$

$$\hat{C}_f = \arg\min_{C_f} G_2((\hat{C}_e, C_f), N). \tag{7.17}$$

By using these two optimization processes, we enforce that the classes $\hat{C}_e$ are monolingually 'good' classes and that the classes $\hat{C}_f$ correspond to $\hat{C}_e$.

7.3 Implementation

An efficient optimization algorithm for the criterion G_1 is the exchange algorithm [Martin & Liermann$^+$ 98]. For the optimization of G_2, we can use the same algorithm with small modifications. The starting point is a random partition of the training corpus vocabulary. This initial partition is improved iteratively by moving a single word from one class to another. The algorithm `bil-word-cluster` to determine bilingual classes is shown in Figure 7.1.
If only one word w is moved between the partitions C and C' the change $G(C, N_g) - G(C', N_g)$ can be computed efficiently looking only at classes C for which $N_g(w, C) > 0$ or $N_g(C, w) > 0$. We define M_0 to be the average number of seen predecessor and successor word classes. With the notation I for the number of iterations needed for convergence, B for the number of word bigrams, M for the number of classes and V for the vocabulary size the computational complexity of this algorithm is roughly $I \cdot (B \cdot \log_2 (B/V) + V \cdot M \cdot M_0)$. A detailed analysis of the complexity can be found in [Martin & Liermann$^+$ 98].
The algorithm provides only a local optimum. The quality of the resulting local optima can be improved if we accept a short-term degradation of the optimization criterion during the optimization process. We do this in our implementation by applying the optimization method *threshold accepting* [Dueck & Scheuer 90], which is an efficient simplification of *simulated annealing*.

7.4 Results

Table 7.1 and Table 7.2 provide examples of bilingual word classes. We see that the resulting classes often contain words that are similar in their syntactic and semantic functions. The grouping of words with a different meaning such as *'today'* and *'tomorrow'* does not imply that these words should be translated by the same Spanish word, but it does imply that the translations of these words are likely to be in the same Spanish word class.

Table 7.1: Example of bilingual word classes (EUTRANS-I task, method BIL-2).

how it pardon what when where which who why
today tomorrow
ask call make
carrying changing giving looking moving putting sending showing waking
full half quarter
c'omo cu'al cu'ando cu'anta d'onde dice dicho hace qu'e qui'en tiene
ll'eveme mi mis nuestra nuestras nuestro nuestros s'ubanme
hoy ma nana mismo
hacerme ll'ameme ll'amenos llama llamar llamarme llamarnos llame p'idame p'idanos pedir pedirme pedirnos pida pide
cambiarme cambiarnos despertarme despertarnos llevar llevarme llevarnos subirme subirnos usted ustedes
completa cuarto media menos

Table 7.2: Example of bilingual word classes (VERBMOBIL task, method BIL-2).

Tag Tages tägig tägige tägigen tägiges
Gesamtheit ganze ganzen gesamte gesamten größte
einige etliche minimale paar wenige wenigen
gell klar ne nichtwahr richtig stimmt
bestimmt durchaus freilich sicher sicherer sicherlich
ach ah au hach och oh
früher früheren später spätere späteren vorne
auftauchen komme rüberkommen reinkommen vorbeikommen vorbeischauen
Bad Bar Garage Qual Sauna Solarium Video Werkstatt
ähnlich ähnlichem andernfalls ansonsten außerdem hoffentlich sonst
Galerien Laster Whirlpool
day
biggest entire shooting whole
earlier later thereafter
minimal sacred some
absolutely definitely usually
reluctantly sure
ah oh
earlier later thereafter
afford come jump step
bar bathroom fatter garage large pain sauna solarium video
besides effectively hopefully nevertheless otherwise since therefore

Chapter 8

Results of Alignment Template Approach

The alignment template approach to statistical MT has been applied to a large variety of tasks and language pairs. Various comparisons with other MT systems have been performed. The alignment template approach often obtains better results than other state-of-the-art approaches. In this chapter, we present some of the results obtained in the context of the VERBMOBIL project, the 2002 NIST MT evaluation and using the Canadian Hansards. Additional results for various European languages are presented in Chapter 9 and the results obtained in the EUTRANS project are summarized in the Appendix A.

8.1 VERBMOBIL Task

8.1.1 VERBMOBIL Training and Test Environment

The goal of the VERBMOBIL project is the translation of spoken dialogues in the domains of appointment scheduling and travel planning. Within the VERBMOBIL project, spoken dialogs were recorded. These dialogs were manually transcribed and later manually translated by VERBMOBIL project partners. Because different human translators were involved, there is great variability in the translations.
Each of these so-called dialog turns may consist of several sentences spoken by the same speaker. The dialog turns are split into shorter segments using punctuation marks as potential split points. A standard vocabulary had been defined for the various speech recognizers used in VERBMOBIL. However, not all words of this vocabulary were observed in the training corpus. Therefore, the translation vocabulary was extended semi-automatically by adding about 13 000 German–English word pairs from an online bilingual lexicon available on the web. The resulting lexicon contained not only word-word entries, but also multi-word translations, especially for the large number of German compound words. To counteract the sparseness of the training data, a couple of straightforward rule-based preprocessing steps were applied *before* any other type of processing:

- normalization of:
  – numbers,
  – time and date phrases,
  – spelling: *'don't'* $\rightarrow$ *'do not'*,...

Table 8.1: Statistics of VERBMOBIL task: training corpus (Train), conventional dictionary (Lex), development corpus (Dev), test corpus (Test), (Words*: words without punctuation marks).

		No Preprocessing		With Preprocessing	
		German	English	German	English
Train	Sentences	58 073			
	Words	519 523	549 921	522 933	548 874
	Words*	418 974	453 612	420 919	450 297
	Singletons	3 453	1 698	3 570	1 763
	Vocabulary	7 940	4 673	8 102	4 780
Lex	Entries	12 779			
	Extended Vocabulary	11 501	6 867	11 904	7 089
Dev	Sentences	276			
	Words	3 159	3 438	3 172	3 445
	Trigram PP	–	28.1	-	26.3
Test	Sentences	251			
	Words	2 628	2 871	2 640	2 862
	Trigram PP	–	30.5	-	29.9

- splitting of
  German compound words.

For the training of word classes (Chapter 7), we use as additional preprocessing of the corpus a categorization of proper names for persons and cities. Thereby, we guarantee that all these words are in pre-specified classes.

Table 8.1 shows the corpus statistics of this task. We use a training corpus to train the alignment template model and the language models, a development corpus, which is used to estimate the model scaling factors, and a test corpus. The 58 073 sentence pairs comprise about half a million running words for each language of the bilingual training corpus. The vocabulary size given is the number of full word forms seen in that corpus including the punctuation marks. Notice the large number of word types seen only once. The extended vocabulary is the vocabulary after adding the conventional dictionary.

Table 8.2 shows the number of alignment templates of different length found using the learning algorithm `phrase-extract` described in Section 5.2. We see that for long phrases the number of distinct phrases and the number of running phrases is very similar. Hence, most of the long phrases are seen only once. Comparing the results for a different number of word classes, we see that the use of word classes significantly reduces the number of distinct alignment templates.

An effect of word classes is the reduced memory consumption. Table 8.3 shows the effective amount of memory in megabytes that is needed to store the alignment templates. Using word classes, the number of different phrases reduces and as a result, memory consumption reduces.

Table 8.2: Statistics of bilingual phrases in training and test using `phrase-extract`.

Length of ATs	# Distinct ATs				# Occurrences of ATs
	100 classes	500 classes	1000 classes	no classes	
1	6967	9571	12086	16106	412814
2	34353	57299	63837	69593	272283
3	78306	101548	106151	108639	201416
4	102550	117566	119877	121337	162633
5	105477	114276	115230	115926	134928
6	98531	102928	103207	103591	113711
7	87417	89218	89290	89509	95855
8	75204	76044	76070	76223	80668
9	63512	63923	63934	64046	67330

Table 8.3: Memory consumption of alignment templates.

Maximal Length of ATs	Memory Consumption [MB]			
	100 classes	500 classes	1000 classes	no classes
1	0.2	0.3	0.3	0.5
2	1.1	1.9	2.2	2.6
3	3.7	5.6	6.1	6.6
4	8.1	11.0	11.7	12.3
5	13.8	17.5	18.3	18.9
6	20.2	24.4	25.2	25.9
7	26.9	31.4	32.2	32.9
8	33.5	38.1	39.0	39.7
9	39.8	44.5	45.4	46.2

Table 8.4: Effect of maximum entropy training for alignment template approach using a direct translation model (WP: word penalty feature, CLM: class-based language model (five-gram), MX: conventional dictionary).

	objective [%]					subjective [%]	
	SER	WER	PER	mWER	BLEU	SSER	IER
Baseline($\lambda_m = 1$)	70.9	40.5	30.8	34.2	47.9	38.0	42.1
ME	58.6	38.7	28.5	32.0	52.0	34.3	36.1
ME+WP	55.8	38.7	26.8	31.5	55.2	30.1	32.4
ME+WP+CLM	54.1	37.7	26.5	30.6	56.1	29.2	31.4
ME+WP+CLM+MX	53.4	37.3	26.5	30.3	56.4	29.3	30.9

Table 8.5: Effect of maximum entropy training for alignment template approach using an inverted translation model (conventional source–channel approach).

	objective [%]					subjective [%]	
	SER	WER	PER	mWER	BLEU	SSER	IER
Baseline($\lambda_m = 1$)	64.5	40.9	27.2	34.8	51.9	31.1	33.0
ME	58.6	39.9	26.9	33.7	52.9	30.4	30.9
ME+WP	59.0	40.1	27.1	33.8	52.4	30.0	30.9
ME+WP+CLM	57.4	39.5	26.8	32.7	54.0	28.4	29.9
ME+WP+CLM+MX	54.1	39.3	26.6	32.3	54.4	28.8	30.1

8.1.2 Effect of Various Model Parameters

In the following, we analzye the effect of various model parameters. As evaluation criteria are used the error rates described in Section 3.5. For all results, the objective error criteria are used. Since the subjective evaluation is very expensive, the subjective criteria are only used for selected experiments.

Effect of maximum entropy modeling of alignment templates

In the following, we present the results of maximum entropy training for the features described in Section 6.5. Table 8.4 shows the results if we use a direct translation model (Eq. 1.6). In addition to the normal error rates, we use the sentence error rate (SER), which is computed as the number of times that the generated string corresponds exactly to one of the reference translations used in maximum entropy training. On average, 3.32 reference translations for the development corpus and 5.14 reference translations for the test corpus are used.
As baseline features, we use a normal word trigram language model and the three component models of the alignment templates. The first row shows the results using only the four baseline features with $\lambda_1 = \cdots = \lambda_4 = 1$. The second row shows the result if we train the model scaling factors. We see a systematic improvement on all error rates. The mWER improves from 34.2% to 32.0%. If we add the word penalty feature (WP), an mWER of 31.5% is obtained and also the other error rates improve. Adding both features, the class-based five-gram language model (CLM) and the conventional dictionary (MX), we observe an additional improvement obtaining an mWER of 30.3%. Yet, the improvement on the other error rates is only small.

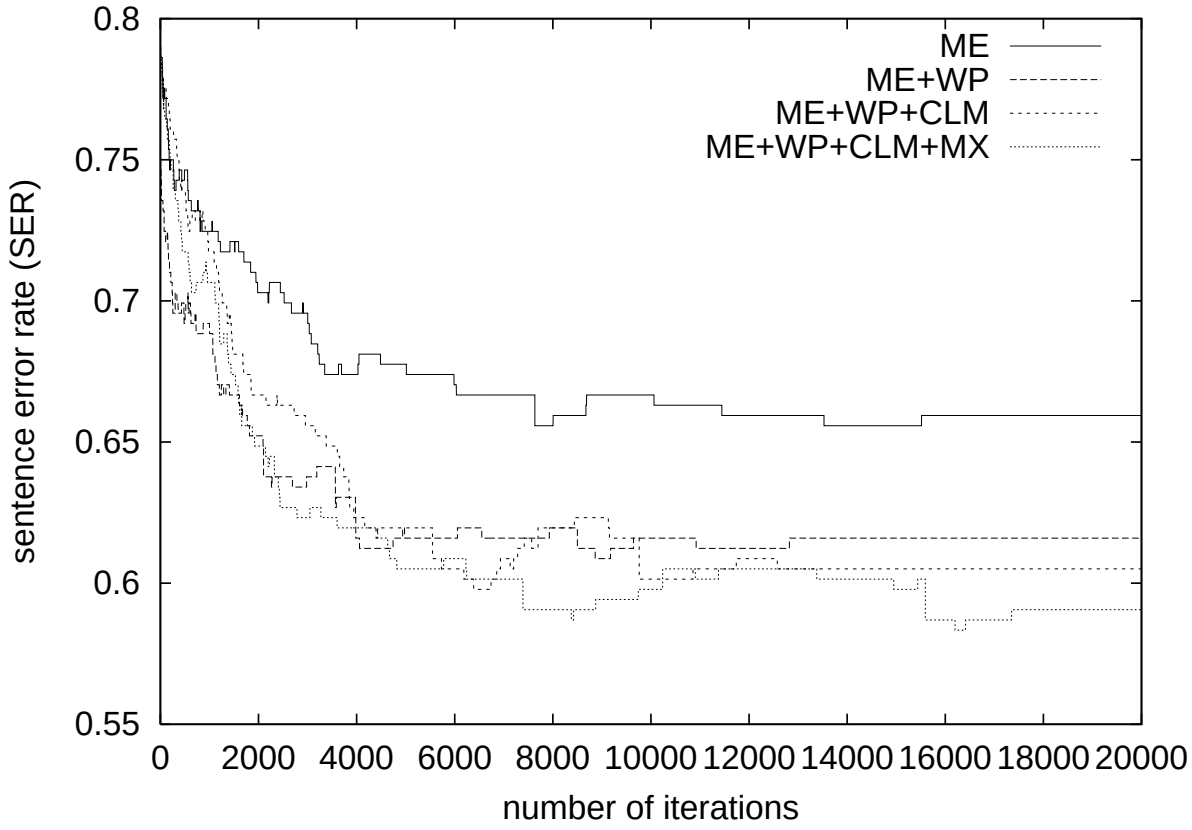


Figure 8.1: Training error rate over the iterations of the GIS algorithm for maximum entropy training of alignment templates.

Table 8.5 shows the results if we use a standard source–channel approach (Eq. 6.24). Also in this case, maximum entropy training using the additional features provides significant improvement. Comparing the results with Table 8.4, we see that the optimized direct translation approach obtains on some error rates a slightly better translation quality than the source–channel approach. Interestingly, we observe the maximum entropy training yields a smaller improvement in the case of the source–channel approach. In addition, the baseline quality of the source channel approach is significantly better than for the direct translation approach.

As we mentioned already in Section 6.3, the direct approach allows a better computation of the probability of a hypothesis, which leads to a more efficient pruning and therefore to a more efficient search (see Table 8.12 – Table 8.15). Hence, the following results will be made mainly using the direct translation approach.

Figure 8.1 and Figure 8.2 show how the sentence error rate (SER) on training and test improves during the iterations of the GIS algorithm. We see that the sentence error rates converges after about 4000 iterations. We do not observe significant overfitting.

Table 8.6 shows the resulting normalized model scaling factors. Multiplying each model scaling factor by a constant positive value does not affect the decision rule. We see that adding new features also has an effect on the other model scaling factors.

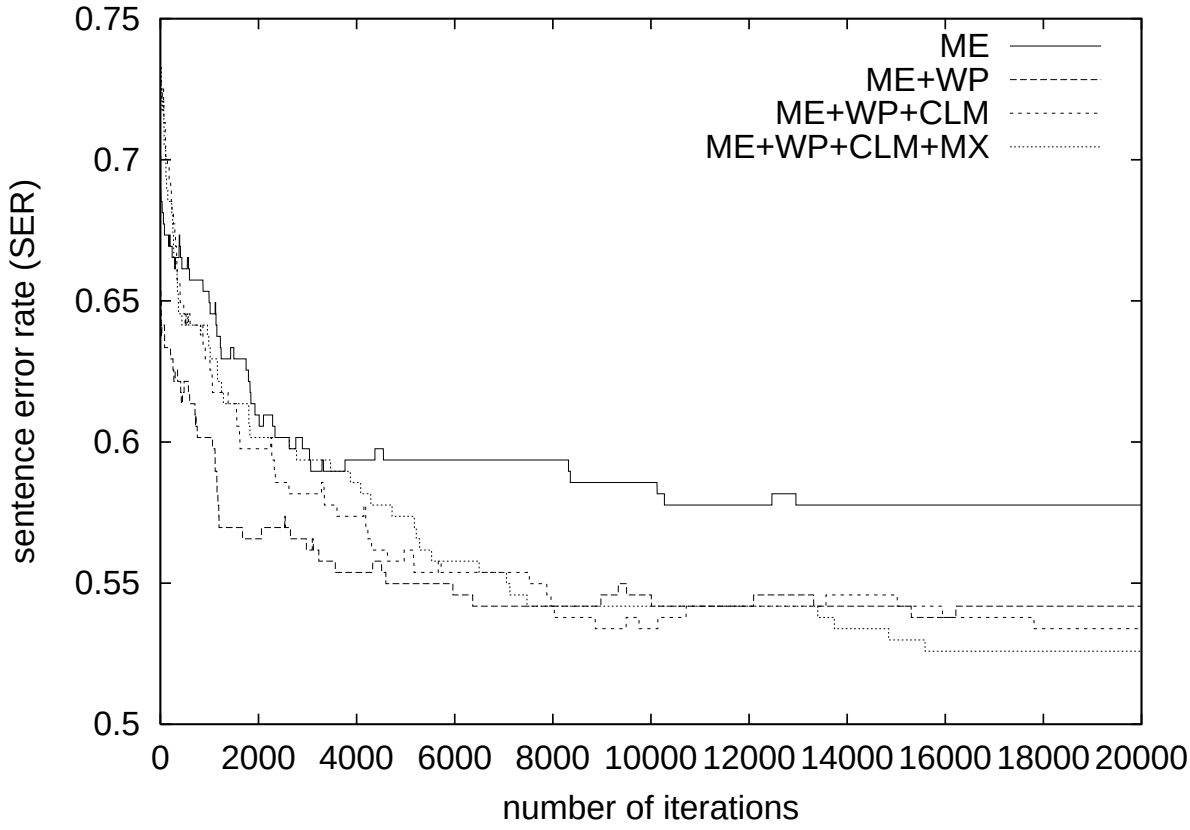


Figure 8.2: Test error rate over the iterations of the GIS algorithm for maximum entropy training of alignment templates.

Effect of lexicon model in the alignment templates

The lexicon model $p(\tilde{f}|z,\tilde{e})$ in Eq. 6.13 models the word choice. Without using word classes, the alignment template itself constrains the words. Hence, then the effect of this model is to obtain a smoothed version of the 'hard' phrase translation model designed in Chapter 5. The impact of this model can be adjusted using the corresponding model scaling factor. Table 8.7

Table 8.6: Resulting model scaling factors of maximum entropy training for alignment templates; λ_1: trigram language model; λ_2: alignment template model, λ_3: lexicon model, λ_4: alignment model (normalized such that $\sum_{m=1}^{4} \lambda_m = 4$).

	ME	+WP	+CLM	+MX
λ_1	0.86	0.98	0.75	0.77
λ_2	2.33	2.05	2.24	2.24
λ_3	0.58	0.72	0.79	0.75
λ_4	0.22	0.25	0.23	0.24
WP	·	2.60	3.03	2.78
CLM	·	·	0.33	0.34
MX	·	·	·	2.92

Table 8.7: Effect of model scaling factor for lexicon model $p(\tilde{f}|z,\tilde{e})$ (300 word classes).

	objective [%]			
smoothing parameter	WER	PER	mWER	BLEU
0.0	41.3	30.3	34.5	51.4
0.2	39.9	29.6	32.8	52.6
0.4	39.2	27.7	32.1	53.8
0.6	38.9	27.3	31.9	54.2
0.8	38.6	27.2	31.4	54.6
1.0	38.3	27.0	31.3	55.1
2.0	40.0	27.7	33.1	52.5
4.0	41.8	29.4	35.0	48.3

Table 8.8: Effect of model scaling factor for lexicon model $p(\tilde{f}|z,\tilde{e})$ (no word classes).

	objective [%]			
smoothing parameter	WER	PER	mWER	BLEU
0.0	40.6	28.8	33.9	53.3
0.2	39.4	27.4	32.6	54.7
0.4	39.0	26.8	32.1	55.6
0.6	39.0	26.6	32.1	55.2
0.8	39.2	26.6	32.2	54.9
1.0	39.1	26.9	32.1	54.7
2.0	39.6	27.3	32.7	53.5
4.0	42.0	29.3	35.0	48.8

shows the results obtained with different model scaling factors if we use 300 word classes. We see that completely avoiding this model by setting the model scaling factor to zero results in a poor translation quality. Table 8.8 shows the results obtained if each word is in its own class. We see, that also in this case the lexicon model has a positive effect. Yet, the improvement is smaller than in the case when we use word classes.

Effect of alignment model

Table 8.9 shows the effect of changing the model scaling factor for the alignment model. Setting the model scaling factor to 0.0 then only the language model distinguishes between different word orders. We see that the resulting translation quality is poor. If the model scaling factor is set to a very large value, then this effectively forbids any reordering, hence we obtain a monotone translation on the level of alignment templates. Optimal translation quality is obtained using an alignment model scaling factor between 0.2 and 0.4. The position-independent word error rate (PER) is almost not affected—it only changes between 28.1 and 29.0.

Table 8.9: Effect of model scaling factor for alignment model.

alignment model scaling factor	objective [%]			
	WER	PER	mWER	BLEU
0.0	43.6	29.0	36.8	50.7
0.2	39.6	28.1	32.6	53.3
0.4	39.1	28.3	32.1	53.3
0.6	39.7	28.4	33.0	52.7
0.8	40.1	28.6	33.5	51.7
1.0	40.8	28.7	34.5	50.3
2.0	42.3	28.8	36.4	47.2
4.0	42.6	28.8	36.6	46.8
1000.0	42.6	28.8	36.6	46.8

Table 8.10: Effect of alignment template length on translation quality.

maximum AT length	objective [%]			
	WER	PER	mWER	BLEU
1	45.4	29.8	39.6	44.6
2	39.2	27.0	32.6	53.6
3	37.5	26.5	30.5	56.1
4	38.3	26.9	31.2	55.8
5	38.3	26.8	31.2	55.8
6	37.8	26.5	30.6	56.1
7	37.7	26.5	30.6	56.1

Effect of alignment template length

Table 8.10 shows the effect of constraining the maximum length of the alignment templates in the source language. Typically, the alignment template length has to be restricted to keep the memory requirements low. We see that using only alignment templates with one or two words in the source languages results in a poor translation quality. Yet, already using alignment templates of length 3 yield very good results.

Effect of word alignment quality on translation quality

Table 8.11 shows the effect of the quality of the used alignment model and alignment symmetrization method on the resulting translation quality. We compare the alignment error rate obtained with various alignment models with the resulting translation quality. Therefore, we do not use preprocessing in these experiments as the reference alignment has been performed for the corpus without preprocessing.

We see that an improved word alignment typically yields an improved translation quality.

Table 8.11: Effect of alignment quality on translation quality (without preprocessing).

model	AER[%]	objective [%]				subjective [%]	
		WER	PER	mWER	BLEU	SSER	IER
Model 1	16.7	47.9	40.8	43.5	43.7	41.0	43.8
HMM	8.3	45.9	38.7	41.5	45.0	40.6	42.4
Model 4	5.4	44.0	35.4	39.1	48.9	36.1	39.8

Effect of pruning and heuristic function

In the following, we analyze the effect of beam search pruning and of the heuristic function. We use the following criteria:

- **Number of search errors:** A search error occurs, if the search algorithm produces a translation that is not the optimal with respect to the optimization criterion. As we typically cannot efficiently compute the probability of the optimal translation, we cannot efficiently compute the amount of search errors. Yet, we can compute a lower bound by comparing with the best translation that we have found using very conservative pruning thresholds. The best translation that is thus obtained is used to detect search errors.

- **Average translation time per sentence:** Pruning is used to adjust the optimal trade-off between efficiency and quality. Hence, we present the average time needed to translate one sentence of the test corpus.

- **Translation quality (mWER, BLEU):** Typically, a sentence can have many different correct translations. Therefore, a search error not necessarily yields worse translation quality. A search error might even improve translation quality. Hence, we analyze the effect of search on translation quality. We use the automatic evaluation criteria mWER and BLEU.

Table 8.12 shows the effect of the threshold pruning parameter t_p with the histogram pruning parameter $N_p = 50\ 000$ using the source–channel translation approach. Table 8.13 shows the obtained error rates. Table 8.14 and Table 8.15 show the corresponding results using the direct translation approach. Table 8.16 and Table 8.17 show the effect of the pruning parameter N_p with the pruning parameter $t_p = 10^{-12}$. In all six tables, we provide the results for using no heuristic functions and three variants of an increasingly informative heuristic function of using only an estimate of the alignment template and the lexicon probability (AT), in addition an estimate of the language model (ATL) probability and in addition also the alignment probability (ATLJ). These heuristic functions are described in Section 6.4.
We observe that the used search algorithm for the direct translation model is more efficient than for the source–channel approach. We need a significantly larger beam and significantly more search time to obtain low numbers of search errors in the source–channel model.
Without heuristic function, even more than a hundred seconds per sentence cannot guarantee search error free translation. We draw the conclusion that a good heuristic function is very important to obtain an efficient search algorithm.
In addition, the search errors have a more severe effect on the error rates if we do not use a heuristic function. If we compare the error rates in Table 8.17 which correspond to about 55 search errors in Table 8.16, we obtain an mWER of 36.4 % (53 search errors) using no

Table 8.12: Effect of pruning parameter t_p and heuristic function on search efficiency for source–channel translation model ($N_p = 50\ 000$).

	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
	time	search	time	search	time	search	time	search
t_p	[s]	errors	[s]	errors	[s]	errors	[s]	errors
10^{-1}	0.0	245	0.0	239	0.0	237	0.0	236
10^{-5}	0.2	218	0.2	203	0.2	180	0.1	157
10^{-9}	0.7	155	0.9	138	1.0	111	0.5	81
10^{-13}	11.5	111	14.8	89	18.1	69	7.0	43
10^{-17}	25.4	79	29.6	61	36.0	33	17.0	17
10^{-21}	113.0	64	125.6	35	146.2	20	83.1	6
10^{-25}	148.1	50	158.1	26	182.4	16	104.2	4
10^{-29}	387.3	45	402.0	23	465.5	16	288.2	3
10^{-33}	571.6	38	606.1	23	727.5	15	376.9	2

Table 8.13: Effect of pruning parameter t_p and heuristic function on error rate for source–channel translation model ($N_p = 50\ 000$).

	error rates [%]							
	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
t_p	mWER	BLEU	mWER	BLEU	mWER	BLEU	mWER	BLEU
10^{-1}	69.8	27.7	66.2	31.1	60.2	35.8	51.7	38.6
10^{-5}	58.1	38.1	54.3	39.7	49.6	43.1	41.6	47.2
10^{-9}	48.6	42.9	44.1	45.7	41.2	47.9	36.3	52.2
10^{-13}	42.8	46.4	40.0	49.0	37.1	51.7	34.2	53.0
10^{-17}	39.2	49.8	36.4	52.0	34.0	53.2	32.6	53.9
10^{-21}	37.9	50.5	34.7	52.9	33.2	53.9	32.2	54.3
10^{-25}	37.7	51.2	33.5	53.6	33.1	53.8	32.2	54.3
10^{-29}	36.7	52.0	33.9	53.5	33.3	53.8	32.1	54.3
10^{-33}	35.5	53.0	33.8	53.5	33.1	53.8	32.0	54.3

heuristic function and an mWER of 32.3 % (57 search errors) using the ATLJ heuristic function. The reason is that without heuristic function often first the ‘easy’ part of the input sentence is translated. This yields severe reordering errors.

Effect of the length of the language model history

In this thesis, we use only n-gram based language models. Ideally, we would like to take into account long-range dependencies. Yet, long n-grams are seen rarely and are therefore rarely used on unseen data. Therefore, we expect that extending the history length will at some point not improve further translation quality.
Table 8.18 shows the effect of the length of the language model history on translation quality. We see that the language model perplexity improves from 4781 for a unigram model to 29.9 for a trigram model. The corresponding translation quality improves from an mWER of 44.9% to

Table 8.14: Effect of pruning parameter t_p and heuristic function on search efficiency for direct translation model ($N_p = 50\ 000$).

	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
	time	search	time	search	time	search	time	search
t_p	[s]	errors	[s]	errors	[s]	errors	[s]	errors
10^{-1}	0.0	216	0.0	213	0.0	203	0.0	180
10^{-2}	0.0	194	0.0	174	0.0	150	0.0	91
10^{-3}	0.1	136	0.1	97	0.1	77	0.1	35
10^{-4}	0.2	97	0.2	57	0.3	40	0.2	13
10^{-5}	0.6	76	0.8	40	1.3	18	0.6	7
10^{-6}	2.0	61	2.8	21	4.1	11	1.8	3
10^{-7}	5.3	44	7.0	13	9.8	6	4.5	1
10^{-8}	11.9	41	15.0	7	19.9	5	9.5	1
10^{-9}	25.7	38	31.8	7	40.9	3	19.9	1
10^{-10}	45.6	38	50.9	6	65.2	3	32.0	1
10^{-11}	81.0	35	82.2	5	103.3	3	50.9	0
10^{-12}	114.6	34	119.2	5	146.2	2	75.2	0

Table 8.15: Effect of pruning parameter t_p and heuristic function on error rate for direct translation model ($N_p = 50\ 000$).

	error rates [%]							
	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
t_p	mWER	BLEU	mWER	BLEU	mWER	BLEU	mWER	BLEU
10^{-1}	53.9	41.9	52.3	43.1	50.4	44.0	40.1	47.7
10^{-2}	48.6	46.9	43.9	49.5	40.5	51.5	33.3	53.5
10^{-3}	41.8	50.6	37.9	52.2	35.1	53.2	31.3	55.0
10^{-4}	39.5	51.0	34.7	53.9	32.1	55.1	30.5	56.0
10^{-5}	36.9	51.7	32.2	54.8	30.9	55.6	30.1	56.2
10^{-6}	36.8	51.4	31.5	55.1	30.6	55.7	30.5	56.1
10^{-7}	35.7	53.2	31.2	55.5	30.6	55.9	30.6	56.1
10^{-8}	35.4	53.1	31.2	55.8	30.9	55.8	30.6	56.1
10^{-9}	35.7	52.9	31.1	55.8	30.8	56.0	30.5	56.1
10^{-10}	35.9	52.9	31.1	55.8	30.7	56.0	30.5	56.1
10^{-11}	35.6	53.1	30.9	55.9	30.7	56.0	30.6	56.1
10^{-12}	35.4	53.0	30.9	55.9	30.7	56.0	30.6	56.1

Table 8.16: Effect of pruning parameter N_p and heuristic function on search efficiency for direct translation model ($t_p = 10^{-12}$).

	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
	time	search	time	search	time	search	time	search
N_p	[s]	errors	[s]	errors	[s]	errors	[s]	errors
1	0.0	237	0.0	238	0.0	238	0.0	232
10	0.0	169	0.0	154	0.0	148	0.0	98
30	0.1	132	0.1	106	0.1	98	0.1	57
100	0.3	101	0.3	69	0.3	60	0.2	21
300	0.7	82	0.7	49	0.8	38	0.7	11
1000	2.2	65	2.3	33	2.4	27	2.0	5
3000	5.9	53	5.9	19	6.6	15	5.0	5
10000	18.3	40	18.3	10	21.1	5	14.3	1
30000	56.7	35	61.0	6	71.5	2	41.1	0
50000	114.6	34	119.2	5	146.2	2	75.2	0

Table 8.17: Effect of pruning parameter N_p and heuristic function on error rate for direct translation model ($t_p = 10^{-12}$).

	error rates [%]							
	no heur. f.		+altemp+lex (AT)		+lm (ATL)		+al (ATLJ)	
N_p	mWER	BLEU	mWER	BLEU	mWER	BLEU	mWER	BLEU
1	63.4	29.9	60.9	31.8	58.8	32.4	48.7	38.2
10	46.4	47.0	42.8	49.3	41.6	49.2	34.3	52.4
30	43.2	49.4	39.5	50.2	37.6	50.9	32.3	54.7
100	40.8	49.9	36.5	52.7	34.5	53.9	31.0	55.7
300	39.0	51.3	34.8	53.5	33.2	54.4	30.4	56.0
1000	37.5	51.6	32.7	54.6	32.0	55.3	30.4	56.0
3000	36.4	52.3	31.8	55.5	31.3	55.7	30.4	56.1
10000	35.1	53.2	31.1	55.7	30.7	55.6	30.6	56.1
30000	35.2	53.2	30.7	55.9	30.6	56.0	30.6	56.1
50000	35.4	53.0	30.9	55.9	30.7	56.0	30.6	56.1

Table 8.18: Effect of the length of the language model history (Unigram/Bigram/Trigram: word-based; CLM: class-based 5-gram).

Language Model Type	PP	objective [%]			
		WER	PER	mWER	BLEU
Zerogram	4781.0	50.1	38.1	44.9	29.1
Unigram	203.1	45.0	30.2	40.1	37.8
Bigram	38.3	39.3	26.9	32.6	53.1
Trigram	29.9	38.7	26.8	31.5	55.2
Trigram + CLM	-	37.7	26.5	30.6	56.1

an mWER of 31.5%. The largest effect seems to come from taking into account the bigram dependence which already achieves an mWER of 32.6%. If we perform log-linear interpolation of a trigram model with a class-based 5-gram model, we observe an additional small improvement in translation quality to an mWER of 30.6 %.

Effect of word classes

To increase the generalization capability of the translation model and to reduce the memory consumption of the alignment templates, word classes are used. Table 8.19 shows the results using different numbers of word classes. In addition, to the obtained error rates is shown also the average length of the alignment templates in German. For every different length of alignment templates the model scaling factors have been optimized using the development corpus. The best mWER of 30.6% is obtained using 300 word classes. Using only 100 word classes, the mWER increases to 32.9%. Not using word classes (every word is its own class), we obtain an mWER of 32.4%. Similar effects can be observed using the other error rates. We conclude that the word classes indeed help to yield a slight improvement of translation quality. We attribute this effect to the improved generalization capability of class-based alignment templates instead of word-based alignment templates. Yet, using too few word classes, we overgeneralize and obtain worse error rates.
Table 8.20 shows some example translation which change by the use of word classes. We observe that by using too few word classes, often wrong translations are formed. Using 300 word classes, the translations sound often more fluent than using no word classes.

8.1.3 Official VERBMOBIL Evaluation

While during the progress of the project many offline tests were carried out for the optimization and tuning of the MT system, the most important evaluation was the final evaluation of the VERBMOBIL prototype in spring 2000. This end-to-end evaluation of the VERBMOBIL system was performed at the University of Hamburg [Tessiore & v. Hahn 00].
In addition to the statistical approach, three other translation approaches had been integrated into the VERBMOBIL prototype system [Wahlster 00]:

- a transfer approach, which is based on a manually designed analysis grammar, a set of transfer rules, and a generation grammar [Uszkoreit & Flickinger$^+$ 00, Emele & Dorna$^+$ 00, Becker & Kilger$^+$ 00],

Table 8.19: Effect of the number of different word classes on translation quality (AATL: average alignment template length).

		objective [%]			
# word classes	AATL	WER	PER	mWER	BLEU
100	1.964	39.8	28.0	32.9	53.6
200	1.933	37.5	26.7	30.9	55.5
300	1.903	37.7	26.5	30.6	56.1
400	1.879	38.1	26.5	31.3	55.3
500	1.847	37.9	26.4	30.9	56.0
600	1.863	37.7	26.1	30.8	55.9
700	1.824	38.1	26.3	31.2	55.8
800	1.832	38.2	26.2	31.5	55.5
900	1.823	38.2	26.4	31.3	55.6
1000	1.819	38.6	26.2	31.8	55.5
2000	1.801	38.8	26.3	32.1	55.5
no	1.739	39.4	26.6	32.4	54.9

- a dialogue act based approach, which amounts to a sort of slot filling by classifying each sentence into one out of a small number of possible sentence patterns and filling in the slot values [Reithinger & Engel 00],

- an example-based approach, where a sort of nearest neighbor concept is applied to the set of bilingual training sentence pairs after suitable preprocessing [Auerswald 00].

- a so-called sub-string based approach, which is an example-based approach working not on a whole-sentence level but using also smaller units. This approach has also very many similarities to the here proposed alignment template approach as an almost identical training procedure is used to train word alignments and extract bilingual phrases [Block 00].

In the final end-to-end evaluation, human evaluators judged the translation quality for each of the four translation results using the following criterion:

Is the sentence approximately correct: yes/no?

The evaluators were asked to pay particular attention to the semantic information (e.g. date and place of meeting, participants etc.) contained in the translation. A missing translation which may happen for the transfer approach or other approaches was counted as wrong translation. The evaluation was based on 5069 dialogue turns for the translation from German to English and on 4136 dialogue turns for the translation from English to German. The speech recognizers used had a word error rate of about 25%. The overall sentence error rates, i.e. resulting from recognition *and* translation, are summarized in Table 8.22. In general, the empirical approaches perform better than the other approaches. As we can see, the error rates for the statistical approach are smaller by a factor of about 2 in comparison to the classical rule-based approach or the dialogue act based translation. Table 8.21 shows some translation examples.

Table 8.20: Example translations for the effect of word classes on translations (WC: word classes).

Source	das Einzelzimmer kostet hundert Mark . habe ich Sie richtig verstanden ?
Reference	the single room costs a hundred Deutsch-marks . did I understand you right ?
AlTemp (100 WC)	the single room costs one hundred marks . did I get you right ?
AlTemp (300 WC)	the single room costs one hundred marks . did I get you right ?
AlTemp (1000 WC)	the single room costs one hundred marks . did I get you right ?
AlTemp (no WC)	the single is one hundred marks . did I get you right ?
Source	haben Sie Lust anschließend noch essen zu gehen ?
Reference	would you like to go out for a meal afterwards ?
AlTemp (100 WC)	do you feel like to go out for dinner afterwards ?
AlTemp (300 WC)	do you feel like to go out for dinner afterwards ?
AlTemp (1000 WC)	do you feel like to go out for dinner afterwards ?
AlTemp (no WC)	do you feel like eat afterwards have to go ?
Source	ich habe immer noch nichts verstanden .
Reference	I still did not understand anything .
AlTemp (100 WC)	I have nothing always understood .
AlTemp (300 WC)	I still haven't understood .
AlTemp (1000 WC)	I still have nothing there .
AlTemp (no WC)	I still have nothing there .
Source	das ist gut . dann müssen wir noch die Rückreise klären . haben Sie einen Zeitplan für einen Rückflug ?
Reference	that is fine . then we still have to arrange the return journey . do you have a schedule for a return flight ?
AlTemp (100 WC)	that is good . then we still have to clear return trip . do you have a schedule for a return flight ?
AlTemp (300 WC)	that is good . then we still have to discuss the return trip . do you have a schedule for a return flight ?
AlTemp (1000 WC)	that is good . then we still have to clarify the return trip . do you have a schedule for a return flight ?
AlTemp (no WC)	that is good . then we still have to clarify the return trip . do you have a schedule for a return flight ?

Table 8.21: Translation examples from the official VERBMOBIL evaluation [Tessiore & v. Hahn 00].

spoken:	Wie kommen wir zum Bahnhof in Hannover?
recognized:	wie kommen wir zum Bahnhof in Hannover
transfer:	how are we getting to Hannover into the train station
dialog:	we have to get together to the station in Hanover
example:	arrive at Hanover.
statistical:	how we will get the train station, in Hanover,

spoken:	Ja, ich würde den Flug um Viertel nach sieben vorschlagen.
recognized:	ja ich würde den Flug um Viertel nach sieben vorschlagen
transfer:	well | I would suggest the flight in quarters after seven
dialog:	yeah | I fly at a quarter past seven
example:	yes I would suggest after seven
statistical:	well, I would suggest the flight at a quarter past seven,

spoken:	Okay. Dann treffen wir uns um acht Uhr sechs auf dem Hauptbahnhof in Hamburg. Ist das okay?
recognized:	okay dann treffen wir uns um acht Uhr sechs auf dem Hauptbahnhof in Hamburg ist das okay
transfer:	okay | that is okay
dialog:	yeah | how about at six past eight o'clock at the central station in Hamburg | what do you think
example:	and then we will meet in Hamburg at eight six is that okay.
statistical:	okay, then we will meet at eight oh six at the main train station, in Hamburg, is that okay,

spoken:	Dann muss ich Ihnen gleich mal sagen, die ersten zwei Oktoberwochen sieht's bei mir ganz schlecht aus.
recognized:	dann muss ich Ihnen gleich mal sagen die ersten zwei Oktoberwochen sieht bei mir ganz schlecht aus sind
transfer:	–
dialog:	that does not suit me first at two o'clock
example:	then I have to tell you right away it looks quite bad on my end in on the first two weeks of October I
statistical:	then I have to tell you right away, the first two weeks of October is very bad for me, is,

spoken:	Das tut mir leid, da habe ich –, muss ich auf eine Messe.
recognized:	das tut mir leid da ich muss ich eine Messe
transfer:	I'm sorry about that I must a fair there
dialog:	That does not suit me that is difficult at a fair
example:	–
statistical:	I am sorry, I have got a fair,

Table 8.22: Error rates of spoken sentence translation in the VERBMOBIL end-to-end evaluation. (*: The substring-based search has been evaluated using a different set of evaluators and also only on a selected subset of the test corpus. Therefore, the error rate of the substring-based search is not fully comparable to the other error rates.)

Translation Approach	Error [%]
Semantic Transfer	62
Dialogue Act Based	60
Example-Based	52
Substring-based	35^*
Statistical	29

Error analysis

In this section, we analyze the errors occurring in the developed statistical MT approach in comparison to the other translation approaches in VERBMOBIL. We selected 100 translation examples where the statistical translation approach was not judged as approximately correct. The results are summarized in Table 8.23.
We distinguish 7 categories of translation errors:

- **All four translations wrong**: There were 59 sentences for which none of the described approaches produced a correct translation. So the conclusion is that for these sentences either there is a speech recognition problem or the sentence is so difficult to translate that none of the four approaches worked. For the remaining 41 sentences, one of the following error categories must apply.

- **Word order**: Often, an error occurs because a wrong word order is chosen in the target language. A possible reason for this error is that the m-gram language model for the target language is poor because it allows wrong word sequences. Another possible reason is that the alignment model does not correctly predict changes in word order. Currently, there is a bias towards a monotone alignment.

- **Word sense disambiguation**: In this case, the problem is that a translation of a word is chosen that is wrong in the specific context. This effect occurs most often for frequent prepositions, which have typically many translations. We expect that by using better context-dependent models such as the maximum entropy lexicon models [Berger & Della Pietra$^+$ 96, García-Varea & Och$^+$ 01], we are able to deal with this problem.

- **No partial translation**: In comparison to each of the three other translation approaches, the statistical approach had been designed in such a way that a sentence must be translated as a whole and no part of a sentence can be omitted. So, there were five sentences in which the competing translation approaches were able to do better by omitting parts of the source sentence. These parts may have been corrupted by speech recognition errors or by spontaneous speech phenomena such as false starts.

- **Discontinuous units**: Some errors are caused by source words that are nonconsecutive but strongly interact, and need to be considered in translation as a single unit. This ef-

Table 8.23: Error analysis for 100 selected sentences of the official VERBMOBIL evaluation.

category	# sentences
All 4 approaches wrong	59
Word order	11
Word sense disambiguation	8
No partial translation	5
Discontinuous units	5
Prosodic boundary detection	2
Miscellaneous	10
Total	100

fect occurs very often with German separable verb prefixes. For example, in the sentence 'Wir fahren am nächsten Mittwoch ab' the verb 'abfahren' is splitted into the verb 'fahren' and the prefix 'ab', which is put at the end of the sentence. To deal with this problem, morphological preprocessing [Nießen & Ney 00, Nießen & Ney 01a] or using hierarchical translation models using various levels of morphological analysis [Nießen & Ney 01b] are very promising approaches.

- **Prosodic boundary detection**: Some errors occur because the heuristics used to detect prosodic boundaries made errors by choosing a sentence boundary within a sentence. Improvements can be expected by making the decision for a segmentation of the source sentence as part of the whole decision process.

- **Miscellaneous**: There are 10 sentences for which none of the error categories applies and a more detailed analysis would be required.

Although both text and speech input are translated with good quality on the average, there are examples where the syntactic structure of the produced sentence is not correct. Some of these syntactic errors are related to long-range dependencies and syntactic structures that are not captured by the m-gram language model used. Many fatal errors stem from the speech recognition engine.

8.1.4 Comparison with Baseline Algorithms

In this section, we compare the alignment template approach with the single-word based approach (Model 4) and the monotone phrase-based translation approach of Chapter 5. Here, all algorithms use model scaling factors that have been optimized on the development corpus. The single-word based approach uses the so-called GE reordering constraint. In addition, in a preprocessing step some sequences of English words are replaced by single vocabulary entries. These phrasal translation have been also automatically trained using a likelihood criterion [Tillmann 01].
The same training corpus and conventional dictionary has been used for all methods. The single-word based approach uses a specific preprocessing that is tuned for this search approach. For each method, the model scaling factors have been optimized on held-out data. The alignment template approach and example-based approach used identical corpus preprocessing. All

Table 8.24: Comparison of the monotone single-word based translation model and various variations of the phrase-based monotone translation models.

	objective [%]				subjective [%]	
	WER	PER	mWER	BLEU	SSER	IER
monotone single-word based	48.5	34.5	42.2	37.9	47.7	52.0
PBMonTrans	43.5	31.1	37.9	43.8	39.6	40.8
PBMonTrans/smoothed	42.2	29.7	36.3	45.8	36.6	39.0
single-word based	41.6	30.9	35.1	48.2	35.4	40.2
AlTemp	37.7	26.5	30.6	56.1	29.2	31.1

models use the same trigram language model. The alignment template approach uses 300 word classes.

Table 8.24 shows the results for comparing the single-word based monotone search with phrase-based monotone search. For the monotone phrase-based translation, we compare two variants: the unsmoothed approach described in Chapter 5 and using smoothing with the single-word based lexicon probability version as for the alignment template approach described in Section 6.1.2. We see that the method PBMonTrans produces significantly better results than the monotone single-word based translation. While the single-word based model achieves only an mWER of 42.2%, the monotone phrase-based translation achieves an mWER of 36.3%.

Table 8.24 also shows the results for comparing a single-word based approach with reordering with monotone phrase-based translation (PBMonTrans) and the alignment template approach. We see that PBMonTrans almost reaches the quality of the single-word based approach. Taking into account that PBMonTrans uses a much simpler model without any word reordering, we conclude that bilingual phrases are very important. The alignment template approach obtains the best results.

Table 8.25 shows some example translations. We often observe that the translations of PBMonTrans are more fluent than the translations with Model 4. Yet, the monotonicity constraint yields severe syntactic errors. This is for example the case if the verb group in German is separated and in the English translation it would be necessary to move the second part of the verb group together with the first part. Here, the translations obtained with the alignment template approach are often able to perform a correct word reordering.

8.2 Results on the HANSARDS task

The HANSARDS task contains the proceedings of the Canadian parliament, which are kept by law in both French and English. About 3 million parallel sentences of this bilingual data has been made available by the Linguistic Data Consortium (LDC). Here, we use a subset of the data containing only sentences of up to 30 words. Table 8.26 shows the training and test corpus statistics.

The results for French to English and for English to French are shown in Table 8.27. Due to memory limitations, the maximum alignment template length has been restricted to 4 words. For the single-word based search, no word joining has been carried out [Tillmann 01]. We see, that the alignment template approach obtains significantly better results than the single-word based search. Table 8.28 shows same example translations.

Table 8.25: Example translations of Model 4, PBMonTrans and alignment template approach for VERBMOBIL (German to English).

Source	genau . der Zug verläßt Hannover um sechs Minuten nach acht .
Reference	exactly . the train leaves Hanover at six past eight .
Model 4	exactly . the train leaves Hanover at six oh past eight .
PBMonTrans	exactly . the train leaves Hanover six minutes after eight .
AlTemp	exactly . Hanover at the train leaves six minutes after eight .
Source	wir treffen uns am besten um acht Uhr auf dem Bahnhof .
Reference	the best thing is we meet at the train station at eight o'clock .
Model 4	we meet best at eight o'clock at the train station .
PBMonTrans	we are meeting should at eight o'clock at the station .
AlTemp	we should meet at eight on the train station .
Source	wir fahren am fünften September wieder zurück mit der Bahn .
Reference	we will go back again by train on the fifth of September .
Model 4	we go on the fifth of September again with the train .
PBMonTrans	we go on the fifth September back by train .
AlTemp	we go back by train again on the fifth of September .
Source	das Einzelzimmer kostet hundert Mark . habe ich Sie richtig verstanden ?
Reference	the single room costs a hundred Deutsch-marks . did I understand you right ?
Model 4	the single room costs one hundred marks , did I understand you correctly ?
PBMonTrans	the single is one hundred marks . did I get you right ?
AlTemp	the single is hundred marks . did I get you right ?
Source	einverstanden . wann genau fliegt das Flugzeug ?
Reference	I agree . when exactly does the plane take off ?
Model 4	okay . when exactly the plane flies ?
PBMonTrans	okay . when exactly does the plane ?
AlTemp	okay . when exactly does the plane ?

Table 8.26: Corpus statistics of HANSARDS task (Words*: words without punctuation marks).

		French	English
Training	Sentences	1 470 473	
	Words	24 338 195	22 163 092
	Words*	22 175 069	20 063 378
	Vocabulary	100 269	78 332
	Singletons	40 199	31 319
Test	Sentences	5432	
	Words	97 646	88 773
	Trigram PP	–	179.8

Table 8.27: Translation results on the HANSARDS task.

	French→English		English→French	
Translation Approach	WER [%]	PER [%]	WER [%]	PER [%]
Alignment Templates	61.5	49.2	60.9	47.9
Single-Word Based: Monotone Search	65.5	53.0	66.6	56.3
Single-Word Based: IBM-Style Reordering	64.9	51.4	66.0	54.4

8.3 Results on Chinese–English

Various statistical or example-based MT systems for a Chinese–English news domain have been evaluated in the NIST 2002 MT evaluation[1]. With the alignment template approach described in this thesis, we participated in these evaluations. The problem domain is the translation of Chinese news text to English.
There have been defined three different resource categories in this evaluation:

- In the small data track a small pre-aligned parallel training corpus of about 100 thousand words and a small conventional lexicon with about 10 thousand entries has been made available.
- In the large data track all Chinese–English parallel resources were allowed that have been made available by LDC (Linguistic Data Consortium). These training corpora have been compiled from various sources and are partially aligned on sentence level and partially aligned on a story level.
- In the unlimited data track it has been possible to collect additional training material from other sources as long as this data has been obtained before March 15, 2002.

We participated in the large data track and the unlimited data track. Table 8.29 and Table 8.30 gives an overview on the used training and test data. The English vocabulary consists of full-form words that have been converted to lowercase letters. The number of sentences is artificially increased by adding certain parts of the original training material more than once to the used training corpus. This has been done to give larger weight to those parts of the training corpus

[1]Evaluation homepage: http://www.nist.gov/speech/tests/mt/mt2001/index.htm

Table 8.28: Example translations of Model 4 and alignment template approach for HANSARDS (SWB: single-word based approach, AlTemp: alignment template approach).

Source	je pourrais peut - être passer maintenant à le 23 octobre 1969 , date de le discours de le trône suivant .
Reference	perhaps I might turn now to October 23 , 1969 , the date of the next Speech from the Throne .
SWB	I might turn to the 1969 October 23 , standing in the Speech from the Throne follows .
AlTemp	I could be - can now go to the October 23 , 1969 date of the speech of the Throne as follows .
Source	je me demande , monsieur le Orateur , si ce est bien là le jeu que on veut jouer avec la justice sociale à le Canada .
Reference	I wonder , Mr. Speaker , if that is the type of game we want to play with social justice in Canada .
SWB	I wonder , Mr. Speaker , if it is clear that it wants play the game with social justice in Canada .
AlTemp	I ask the me , Mr. Speaker , if that well there is the game that it wants to play with the social justice to the country .
Source	ces installations furent construites à le cours de la Seconde Guerre mondiale pour servir de base militaire .
Reference	these facilities were created during the second world war as an army base .
SWB	these facilities were built in Canada during the Second World for use on Base .
AlTemp	these facilities were built in the course of the Second World Wars world for use of military basis .
Source	le coût de les vêtements est passé de $ 134.70 à $ 144.80 .
Reference	clothing which cost $ 134.70 last year now costs $ 144.80 .
SWB	the cost of clothing has increased from $ 134.70 to $ 144.80 .
AlTemp	the cost of clothing is the past of $ $ 134.70 to 144.80 .
Source	et que est - ce qui nous retient à le Canada ?
Reference	and what have we in Canada to contend with ?
SWB	and that is what we in Canada hold ?
AlTemp	and that is - that which we holds to the country ?

Table 8.29: Corpus statistics for Chinese–English corpora — large data track (Words*: words without puncutation marks).

		No Preprocessing		With Preprocessing	
		Chinese	English	Chinese	English
Train	Sentences	1 645 631			
	Unique Sentences	1 289 890			
	Words	31 175 023	33 044 374	30 849 149	32 511 418
	Words*	27 091 283	29 212 384	26 828 721	28 806 735
	Singletons	15 324	24 933	5 336	26 344
	Vocabulary	67 103	92 488	45 111	85 116
Lex	Entries	80 977			
	Extended Vocabulary	76 182	100 704	54 190	93 350
Dev	Sentences	993			
	Words	26 361	32 267	25 852	31 607
	Trigram PP	–	237.154	-	171.922
Test	Sentences	878			
	Words	24 540	–	24 144	–

that consist of high quality aligned Chinese news text and are therefore expected to be especially helpful for the translation of the test data.

As conventional dictionary in the unlimited data track, we use a compilation of various dictionaries available from LDC and publically available in the Internet. For the large data track, a compilation of various versions of the LDC dictionary has been used.

The Chinese language poses special problems because the word boundaries of Chinese words are not marked. Chinese text is provided as a sequence of characters without explicit annotation of word boundaries. For statistical MT, it would be possible to ignore this fact and treat the Chinese characters as elementary units and translate them into English. Yet, preliminary experiments showed that the existing alignment models produce better results if the Chinese characters are segmented in a preprocessing step into single words. We use the LDC segmentation tool[2].

For the English corpus, the following preprocessing steps are applied. First, the corpus is tokenized, segmented into sentences and all uppercase characters are converted to lowercase. Since the final evaluation criterion does not distinguish case, we can ignore case. In a postprocessing step of the system output, the case information is introduced by performing a monotone phrase-based translation using the approach described in Chapter 5.

Then, the preprocessed Chinese and English corpora are sentence aligned. From the resulting corpus, we automatically remove presumably wrong translations. In addition, only sentences with less than 60 words in English and Chinese are used.

To improve the translation of Chinese numbers, we use a categorization of Chinese number and date expressions. For the statistical learning, all number and date expressions are replaced by the generic symbols *'$number'* or *'$date'*. The number and date expressions are translated rule-based by simple lexicon lookup. The translation of the number and date expressions is inserted

[2]The LDC segmentation tool is available at
`http://morph.ldc.upenn.edu/Projects/Chinese/LDC_ch.htm#cseg`

Table 8.30: Corpus statistics for Chinese–English corpora — all data track (Words*: words without puncutation marks).

		No Preprocessing		With Preprocessing	
		Chinese	English	Chinese	English
Train	Sentences	2 234 738			
	Unique Sentences	1 448 875			
	Words	36 757 076	38 685 984	36 255 556	37 932 630
	Words*	32 009 776	34 087 664	31 598 259	33 515 866
	Singletons	19 210	26 871	6 590	28 311
	Vocabulary	77 033	102 200	48 601	91 649
Lex	Entries	430 686			
	Extended Vocabulary	82 031	169 045	56 315	154 348
Dev	Sentences	993			
	Words	26 361	32 267	25 852	31 607
	Trigram PP	–	229.881	-	169.986
Test	Sentences	878			
	Words	24 540	–	24 144	–

in the output using the alignment information. For Chinese and English, this categorization is implemented independently from the other language. As a result, only 57.3% of the bilingual sentence-pairs which include the category symbol *'$number'* have a corresponding number of occurrences of this symbol in source and target language. For the category symbol *'$date'* only 54.4% of the sentences correspond. The reason is the large number of problematic cases where a number in one language is not translated as a number in the other language. For example, the word *'one'* can be an indefinite article or a number and in addition the corresponding (correct) translation in the other language can completely ignore this word. As a result, various inconsistencies are introduced. We expect better results if the training corpus categorization of the English text depends on the categorization of the Chinese text. Doing that, it would be possible to solve many of the ambiguous cases.

In a first experiment, we evaluated the word alignment quality of our alignment models. Table 8.31 and Table 8.32 show the results of these experiments. As reference has been used a set of 272 manually aligned sentences. We observe that, corresponding to the results of Chapter 4, better models typically provide better results. Interestingly, the alignment error rate for English to Chinese does not improve after the HMM training.

If we compare the resulting error rates with alignment quality results on other tasks, we see that the alignment error rate is significantly higher. For example, the best error rate on the German–English VERBMOBIL task is less than 5% and the best error rate on the French–English HANSARDS task is less then 8%. We conclude that the word alignment of Chinese–English is harder than for the other language pairs. A possible reason is the word segmentation problem for Chinese. To improve these results, it might be promising to combine a statistical model for the segmentation of Chinese characters into words and the statistical alignment models into one model. As a result, in the training of such an alignment–segmentation model, it might be possible to learn a Chinese word segmentation that corresponds well to English words.

To evaluate MT quality on this task, NIST (U.S. National Institute of Standardization) made

Table 8.31: Word alignment quality for Hong Kong Hansards corpus for various statistical alignment models (training scheme $1^7H^63^54^36^4$).

	AER [%]	
Model	Chinese → English	English → Chinese
Model 1	33.5	40.5
HMM	29.9	32.9
Model 3	29.8	36.2
Model 4	26.7	33.6
Model 6	26.3	33.0

Table 8.32: Word alignment quality for Hong Kong Hansards corpus for various alignment symmetrization methods.

Model	Combination Method	AER [%]	precision	recall
Model 6	Intersection	26.6	88.3	61.7
	Union	31.5	57.8	88.6
	Refined	24.9	70.5	81.6

available the NIST-09 evaluation tool. This tool provides a modified BLEU score by computing a weighted precision of n-grams modified by a length penalty for short translations. Table 8.33 shows the results of the official evaluation performed by NIST in June 2002. The first row entitled "RWTH-late submission" corresponds to the results that have been submitted to NIST shortly after the official evaluation deadline. The second row entitled "RWTH-official submission" corresponds to the results using a smaller training corpus. These results have been obtained within the allowed deadline. All results have been obtained without knowledge of the reference translations and no optimization on the test data has been performed.

The training corpora for "late" and "official" submission differ with respect to the so-called FBIS data which consists of about 5 million running words in each language. This data has been allowed data for all participants in this evaluation for the large data track and the unlimited data track. Yet, due to copyright problems, this data had not been available for RWTH within the official evaluation deadline. Most of the competing translation approaches (at least in the large data track) used this training data.

The obtained results were with a score of 8.14 and 8.08 significantly better than any other competing approach. The fact that using the additional training material acquired from the Internet seems to deteriorate translation quality requires further investigation. Possible reasons might be that this additional data is very noisy and that a large part is from a different domain than the used test corpus.

In the large data track, the best competing system obtained a score of 7.34 which is 0.80 lower than the score obtained with the alignment template system. In the unlimited data track, the best competing system obtained a score of 7.58 which is 0.50 lower then the score obtained with the alignment template system. In addition to the competing research systems various commercial off-the-shelf-systems have been used which also perform significantly worse. Table 8.34 shows some of the resulting translations.

We conclude that the developed alignment template approach is also applicable to distant language pairs such as Chinese–English. We conclude that the developed statistical models indeed

Table 8.33: Results of Chinese–English NIST MT Evaluation, June 2002 (NIST-09 score: large values are better, *: inofficial contrastive submission).

	NIST-09 score	
System Name	Large Data	Unlimited Data
RWTH—late submission	8.14*	8.08*
RWTH—official submission	7.65	7.83
competing research systems	5.03–7.34	5.84–7.58
best of six commercial off-the-shelf-systems	-	6.08

seem to be largely language-independent.

Table 8.34: Example translations for Chinese–English MT.

Reference	Significant Accomplishment Achieved in the Economic Construction of the Fourteen Open Border Cities in China
Translation	The opening up of the economy of China's fourteen City made significant achievements in construction
Reference	Xinhua News Agency, Beijing, Feb.12 - Exciting accomplishment has been achieved in 1995 in the economic construction of China's fourteen border cities open to foreigners.
Translation	Xinhua News Agency, Beijing, February 12-China's opening up to the outside world of the 1995 in the fourteen border pleased to obtain the construction of the economy.
Reference	Foreign Investment in Jiangsu's Agriculture on the Increase
Translation	To increase the operation of foreign investment in Jiangsu agriculture
Reference	According to the data provided today by the Ministry of Foreign Trade and Economic Cooperation, as of November this year, China has actually utilized 46.959 billion US dollars of foreign capital, including 40.007 billion US dollars of direct investment from foreign businessmen.
Translation	The external economic and trade cooperation Department today provided that this year, the foreign capital actually utilized by China on November to US $ 46.959 billion, including of foreign company direct investment was US $ 40.007 billion.
Reference	According to officials from the Provincial Department of Agriculture and Forestry of Jiangsu, the "Three-Capital" ventures approved by agencies within the agricultural system of Jiangsu Province since 1994 have numbered more than 500 and have utilized over 700 million US dollars worth of foreign capital, respectively three times and seven times more than in 1993.
Translation	Jiangsu Province for the Secretaries said that, from the 1994 years, Jiangsu Province system the approval of the " three-funded " enterprises, there are more than 500, foreign investment utilization rate of more than US $ 700 million, 1993 years before three and seven.
Reference	The actual amount of foreign capital has also increased than 30% more as compared with the same period last year.
Translation	The actual amount of foreign investment has increased by more than 30 % compared with the same period last year .
Reference	Import and export in pudong new district exceeding 9 billion us dollars this year
Translation	Foreign trade imports and exports of this year to the Pudong new Region exceeds us $ 9 billion

Chapter 9

Statistical Multi-Source Translation

9.1 Introduction

In many applications for MT, documents have to be translated into multiple languages. For example, in international organizations such as the European Union or the United Nations, all relevant documents must be translated into all official languages. Often, the document is originally written in one language, and then translated into the other languages. If, for example, an English translation of a French document is first produced, then this translation should be used as additional knowledge source when producing a German translation. So far, existing MT technology is not able to use these additional knowledge sources.
In this section, we describe a new method that is able to use more than one source language to produce a better translation in a new language. Figure 9.1 shows the architecture of such an MT system.
From performing multi-source translation, we expect a better MT quality due to the following reasons:

- Better word sense disambiguation: Often ambiguities that need to be resolved between two languages do not exist between other languages.
- Better word reordering: A significant source of errors in statistical MT is the word reordering problem (Section 8.1.3). The word order between related languages is often very similar whereas the word order between distant languages might differ significantly. By using more source languages, we can expect that among the source languages there is one with a similar word order.

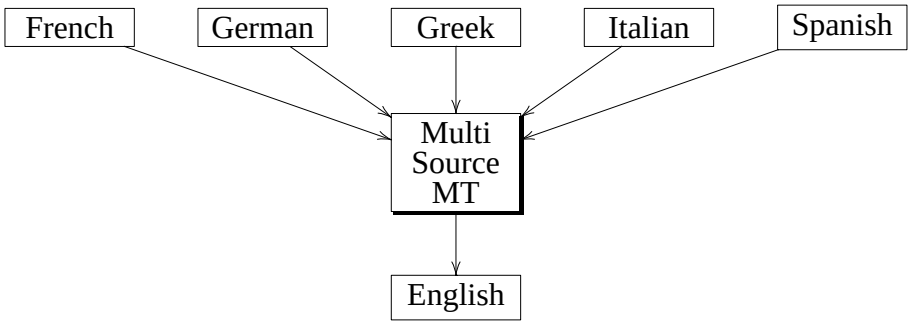


Figure 9.1: Architecture of an MT system using multiple source languages.

- Reduction of the need for explicit anaphora resolution: By having various translations of a pronoun in different languages, the probability increases that it can be translated correctly without performing a detailed anaphora resolution.

In the following, we are able to deduce a general statistical approach to multi-source translation. The described method is very general and independent of specific models, languages or application domains. It fits nicely into the statistical approach and is relatively easy to implement. Ultimately, the approach boils down to a multiplicative combination of various statistical translation models.
In principle, multi-source translation is not restricted to a statistical approach and it would be possible to pursue it also in a transfer-based approach. Yet, we believe that this would be significantly more complicated as already the development of transfer rules for single-source translation is a complex task which requires experts.

9.2 Statistical Modeling

The goal in multi-source translation is the translation of a text given in N source languages into a single target language. We are given N source sentences $\mathbf{f}_1^N = \mathbf{f}_1, \ldots, \mathbf{f}_N$, which are to be translated into a target sentence $\mathbf{e}$. Among all possible target sentences, we choose the sentence with the highest probability:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \{Pr(\mathbf{e}|\mathbf{f}_1^N)\} \tag{9.1}$$

$$= \underset{\mathbf{e}}{\operatorname{argmax}} \{Pr(\mathbf{e}) \cdot Pr(\mathbf{f}_1^N|\mathbf{e})\} \tag{9.2}$$

As in single-source translation $Pr(\mathbf{e})$ is the language model of the target language, whereas $Pr(\mathbf{f}_1^N|\mathbf{e})$ is the multi-source translation model.

Combination method Prod

We make the following assumption: Given the hypothesized target sentence $\mathbf{e}$, the source sentences $\mathbf{f}_n$ are considered statistically independent. Thus, we obtain:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ p(\mathbf{e}) \cdot \prod_{n=1}^{N} p(\mathbf{f}_n|\mathbf{e}) \right\} \tag{9.3}$$

In principle, we have to hypothesize all possible target sentences to perform this maximization. As a first step, we use the approximation that for each language n the best translation $\mathbf{e}_n$ is computed by taking into account only the translation model for this language:

$$\mathbf{e}_n = \underset{\mathbf{e}}{\operatorname{argmax}} \{p(\mathbf{e}) \cdot p(\mathbf{f}_n|\mathbf{e})\}, \quad n = 1, \ldots, N \tag{9.4}$$

To this purpose, we can use a standard search algorithm (Section 6.3) for single-source translation. In the search process for multi-source translation, we hypothesize only these N different target sentences $\mathbf{e}_1, \ldots, \mathbf{e}_N$. Obviously, this is a severe restriction of the search space resulting in search errors. Hence, we expect that better results can be obtained using a general search algorithm.

Model scaling factors

To consider differences in the quality of various models, we can introduce scaling factors α_n for each source language: $p(\mathbf{f}_n|\mathbf{e}) \rightarrow p(\mathbf{f}_n|\mathbf{e})^{\alpha_n}$. Hence, from the viewpoint of direct maximum entropy translation models of Section 6.5, we define $n = 1, \ldots, N$ feature functions for the different source languages:

$$h_n(\mathbf{f}_1^N, \mathbf{e}) = \log p(\mathbf{f}_n|\mathbf{e}) \tag{9.5}$$

Informal experiments have shown that the optimal scaling factors do not deviate much from 1. Therefore, in the experiments, we do not use scaling factors.

Combination method Max

We obtain an even easier decision rule if we perform an additional approximation by replacing the product over all languages by a maximum operation over these languages:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \{p(\mathbf{e}) \cdot \max_n p(\mathbf{f}_n|\mathbf{e})\} \tag{9.6}$$

$$= \operatorname*{argmax}_{\mathbf{e},n} \{p(\mathbf{e}) \cdot p(\mathbf{f}_n|\mathbf{e})\} \tag{9.7}$$

In other words, we translate N times using any of the N source languages. Finally, we choose the translation that obtains the best probability.

9.3 Results

We evaluated the method for performing multi-source translation on the 11-lingual EU Bulletin corpus described in Section 3.2. In all experiments, we use WER (word error rate) and PER (position-independent word error rate) as evaluation criteria (Section 3.5). Both error rates are nicely related to the post-editing effort that a human needs to invest to correct the MT output.

Single-Source Translation Results

For each bilingual corpus, we trained a single-word based alignment model, computed a word alignment and trained the alignment template model. Hence, we obtained ten translation systems from some language to English. Table 9.1 shows the training corpus perplexity (PP), the word error rate (WER) and the position-independent word error rate (PER) of every translation system.
Looking at Table 9.1, we make the following interesting observations:

- The error rates differ significantly for the different languages. The best translation quality is obtained with French (WER: 55.3%) and Portuguese (58.9%) and the worst translation quality is obtained with German (66.9%), Greek (72.4%) and Finnish (83.3%). Obviously, the languages with a very large vocabulary size, due to the rich morphology in these languages, result in a poor translation quality, which shows the necessity of morphologic processing for these languages.

- The error rates correspond to training corpus perplexity. Often, language pairs with a high translation model perplexity also result in a high WER (exception: Dutch).

Table 9.1: Training corpus perplexity of Hidden Markov alignment model and translation results for translating into English from ten different source languages.

Language		PP	WER	PER
French	fr	19.1	55.3	45.3
Portuguese	pt	21.3	58.9	48.2
Spanish	es	18.4	59.2	47.6
Italian	it	24.3	59.5	48.8
Swedish	sv	24.1	60.3	49.9
Danish	da	24.3	62.7	52.9
Dutch	nl	17.6	64.3	51.7
German	de	31.7	66.9	54.2
Greek	el	31.7	72.4	53.0
Finnish	fi	44.2	83.3	66.3

Table 9.2: Absolute improvements in WER combining two languages using method **Max** compared with the best WER obtained by any of the two languages.

	fr	pt	es	it	sv	da	nl
fr	0.0	1.5	1.2	0.5	2.7	1.9	0.8
pt		0.0	2.2	2.1	4.0	3.4	1.3
es			0.0	2.4	3.9	2.6	1.7
it				0.0	3.5	3.2	1.6
sv					0.0	2.7	1.7
da						0.0	4.3
nl							0.0

Multi-source translation results

Table 9.2 and Table 9.3 show the quality improvement in WER when combining the languages French, Spanish, Portuguese, Swedish, Danish, and Dutch using method **Max** and using method **Prod**. These tables show the absolute improvement to the best word error rate obtained by any of the two languages. Using **Max**, we observe an improvement in word error rate between 0.5 and 4.3 percent. Using **Prod**, the improvement is typically lower. Interestingly, the error rates almost never increase. This shows the robustness of the approach.

Table 9.4 shows the translation quality when combining even more languages. We chose always the next language pair that yields the largest improvement. For the combination method **Max**, the additional improvement by using a third language is quite small. Translation quality does not improve when more than three languages are used. For the combination method **Prod**, we observe that the additional improvement by using more languages is still large. Using more than two languages, the combination method **Prod** yields better results than **Max**. In the end, we obtain a WER improvement of 6.5% using six source languages instead of French alone.

Table 9.5 shows some of the examples where a combination of French and Spanish yields an improvement.

Table 9.3: Absolute improvements in WER combining two languages using method **Prod** compared with the best WER obtained by any of the two languages.

	fr	pt	es	it	sv	da	nl
fr	0.0	0.8	0.1	0.4	1.0	0.8	-0.2
pt		0.0	2.6	2.1	2.6	2.8	-0.1
es			0.0	2.4	3.4	3.7	1.1
it				0.0	1.9	3.0	0.3
sv					0.0	1.8	0.5
da						0.0	1.5
nl							0.0

Table 9.4: Combination of more than two languages.

Method	**Max**		**Prod**	
languages	WER	PER	WER	PER
fr	55.3	45.3	55.3	45.3
fr+sv	52.6	43.7	54.3	44.5
fr+sv+es	**52.0**	**43.2**	51.0	41.4
fr+sv+es+pt	52.3	43.6	50.2	40.2
fr+sv+es+pt+it	52.7	44.0	49.8	39.8
fr+sv+es+pt+it+da	52.5	43.9	**48.8**	**39.1**

Table 9.5: Translation examples for multi-source translation ('+': chosen translation).

Source: fr	L'existence de limites financières et sa justification;
Source: es	La existencia de límites financieros y su justificación;
Translation: fr	The existence of limit financial and its justification;
Translation: es +	The existence of financial limits and their justification;
Source: fr	Présentation des perspectives financières dans le cadre de l'élargissement.
Source: es	Presentación de las perspectivas financieras en el contexto de la ampliación.
Translation: fr	Presentation of the financial perspective in the framework of enlargement.
Translation: es +	Presentation of the financial perspective in the context of enlargement.
Source: fr	La Bosnie-et-Herzégovine est désormais acceptée comme une nation.
Source: es	Se reconoce a Bosnia y Herzegovina como un Estado nacional.
Translation: fr +	Bosnia and Herzegovina is now accepted as a nation.
Translation: es	Welcomed to Bosnia and Herzegovina as a State national.

9.4 Conclusions

We have described methods for translating a text given in multiple source languages into a single target language. We have described the general statistical approach to this problem and have developed two specific statistical models: **Prod** and **Max**. We have evaluated the approach on a multilingual corpus collected automatically from the Internet.
For many language combinations, we have been able to obtain significant improvements. The combination method **Max** seems to be better suited for the combination of two languages whereas **Prod** yields better results if three or more languages are combined. Using **Prod**, we have been able to improve word error rate when translating into English from 55.3 percent using French as source language to 48.8 percent using five additional source languages.
The large discrepancies between the translation quality obtained with various languages seem to be mainly due to the sparse data problem resulting from the rich morphology in these languages. Therefore, we expect that a systematic handling of morphology using preprocessing and postprocessing [Nießen & Ney 00, Nießen & Ney 01a] in these languages would result in a comparable translation quality in all 10 source languages. A combination should lead to an additional significant improvement. Further improvements are expected by performing a finer combination of different languages on a phrase level rather than on a complete sentence level.

Chapter 10

Interactive MT

10.1 Motivation

Current MT technology is not able to guarantee high quality translations for large domains. Hence, in many applications, post-editing of the MT output is necessary. In such an environment, the main scope of the MT system is not to produce translations that are understandable for an inexperienced recipient but to support a professional human post-editor.

We can expect that typically a better quality of the produced MT text yields a reduced post-editing effort. Yet, from an application viewpoint, many additional aspects have to be considered: the user interface, the used formats and the additional support tools such as lexicons, terminological databases or translation memories. Many of these influencing factors are not directly related to statistical MT and are therefore outside the scope of this thesis.

Yet, the concept of *interactive MT*, first suggested by [Foster & Isabelle$^+$ 96], finds a very natural implementation in the framework of statistical MT. In interactive MT, the basic idea is to provide an environment to a human translator, which is interactively reacting to user input as the user writes or corrects the translation. In the simplest environment, the system suggests an extension of a sentence that the human can accept or ignore. An implementation of such a tool has been performed in the TransType project [Foster & Isabelle$^+$ 96, Foster & Isabelle$^+$ 97, Langlais & Foster$^+$ 00].

The user interface of the TransType system combines an MT system and a text editor into one application. The human translator types the translation of a given source text. For each prefix of a word, the MT system computes the most probable extension of this word, which is presented to the user. The translator can accept this translation by pressing a certain key or he can ignore the suggestion and continue writing.

A major bottleneck of the TransType approach is that only single-word completions are suggested. It would be preferable that the suggested extension consists of more words or whole phrases. Ideally, the whole sentence should be suggested completely and the translator should have the freedom to accept any prefix of the suggested translation.

In the following, we first describe the problem from a statistical viewpoint. For the resulting decision rule, we describe efficient approximations based on word graphs. Afterwards, we describe the architecture of the implemented interactive MT system. Finally, we present some results.

10.2 Statistical Approach

In a statistical approach, the problem of finding an extension e_{i+1}^I of a given prefix e_1^i can be described by constraining the search to those sentences that contain e_1^i as prefix:

$$\hat{e}_{i+1}^I = \operatorname*{argmax}_{e_{i+1}^I} \{Pr(e_1^I) \cdot Pr(f_1^J | e_1^I)\} \tag{10.1}$$

For simplicity, we do not include in this equation the case where the prefix contains a prefix of the word e_i. In that case, we have to optimize over all target language words e_i that have the same prefix.

In an interactive MT environment, we have to evaluate this quantity after every key-stroke of the translator and present the corresponding extension to the user. For the practicability of this approach, an efficient maximization in Eq. 10.1 is very important. For the human user, a response time larger than a fraction of a second is not acceptable. The search algorithms developed so far are not able to achieve this efficiency without an unacceptable amount of search errors. Hence, we have to perform certain simplifications making the search problem feasible.

Our solution is to precompute a subset of possible word sequences. The search in Eq. 10.1 is then constrained to this set of hypotheses. As data structure for efficiently representing the set of possible word sequences, we use the data structure of word graphs [Ney & Aubert 94, Ortmanns & Ney$^+$ 97].

A translation word graph is a directed acyclic graph $G = (V, E)$, which is a subset of the search graph expanded during normal search (Section 6.3). The word graph is computed as a byproduct of the search algorithm if we maintain for each search node not only the single-best backpointer, but also the back-pointers to the recombined hypotheses. The nodes $n \in V$ correspond to search hypotheses. The edges $(n, n') \in E$ are annotated with target language words $e(n, n')$. The edges are also annotated with the extension probability $p(n, n')$ stemming from language and translation model. For simplicity, we assume that there exists exactly one goal node n_{GOAL} and one start node n_{START}.

For each node in the word graph, the maximal probability path to reach the goal node n_{GOAL} is computed. This probability can be decomposed into the so-called forward probability $p(n)$, which is the maximal probability to reach the node n from the start node n_{START} and the so-called backward probability $h(n)$, which is the maximal probability to reach the node n backwards from the goal node n_{GOAL}.

The backward probability $h(n)$ is an optimal heuristic function. Having this information, we can compute efficiently for each node in the graph the best successor node:

$$BS(n) = \operatorname*{argmax}_{n':(n,n')\in E} \{p(n) \cdot p(n, n') \cdot h(n')\} \tag{10.2}$$

Hence, if the optimal extension of a given translation prefix is given as a node in the word graph, the function BS provides the optimal word sequence in a time complexity linear to the number of words in the extension. If the node n corresponds to the translation prefix e_1^i, the optimal extension is obtained by:

$$\hat{e}_{i+1} = e(n, BS(n)) \tag{10.3}$$

$$\hat{e}_{i+2} = e(BS(n), BS^2(n)) \tag{10.4}$$

$$\hat{e}_{i+k} = e(BS^{k-1}(n), BS^k(n)) \tag{10.5}$$

Yet, as the word graph contains only a subset of the possible word sequences, we might face the problem that the prefix path is not part of the word graph. To avoid that problem, we perform a fuzzy search in the word graph. We find the set of nodes that correspond to word sequences with minimal Levenshtein distance to the given prefix. This can be computed by a straightforward extension of the normal Levenshtein algorithm for word graphs. From this set of nodes, we choose the one with maximal probability and compute the extension according to Eq. 10.2. Because of this approximation, the suggested translation extension might contain words that are already part of the translation prefix.

10.3 Implementation

In the following, we describe the implemented interactive translation system, which consists of an MT environment that allows an effective interaction between the human translator and the MT system using the concept of auto-typing. It has the following key properties:

- The MT system is able to translate text in XML format, while maintaining the XML structure. As MT engine, we use the alignment template approach described in Chapter 6.
- The system allows post-editing the MT output, by an auto-typing facility suggesting a completion of the sentence. The user can accept the complete translation, a single word or a certain prefix using one key-stroke.
- The translator is able to obtain a list of alternative words at a specific position in the sentence. This helps the translator to find alternative formulations.
- Since the system is based on the statistical approach, it can learn from existing sample translations. Therefore, it adapts to very specific domains without much human intervention. Unlike translation memory systems, the system is able to provide suggestions to the user also for sentences that have not been seen in the bilingual translation examples.
- The system can also learn interactively on request from those sentences that have been corrected by the user. The user can request that all the sentences that he corrected are added to the knowledge base. A major aim of this feature is an improved user acceptability as the MT environment is able to adapt rapidly and easily to new vocabulary.

The developed system has various advantages over currently used MT or translation memory environments which combines important concepts from these areas in a unique form into one application. The two major advantages over existing systems are the auto-typing facility, which suggests full-sentence extensions, and the ability to learn interactively from user corrections.
The system is implemented as a client–server application. The server performs the actual translations and performs all time-consuming operations such as computation extensions or the list of alternatives. The client includes only the user interface. Therefore, the client can run on a small computer. Client and server are connected via Internet or Intranet.

10.4 Results

In the following, we present some results using this approach for interactive MT. As evaluation criterion, we use the key-stroke ratio (KSR), which is the ratio of the number of key-strokes

Table 10.1: Key-stroke ratio (KSR) and average extension time for various pruning thresholds ($N_p = 50\ 000$, Verbmobil task).

pruning parameter t_p	single-word extension		sentence extension	
	time [s]	KSR [%]	time [s]	KSR [%]
10^{-1}	0.005	61.2	0.005	50.9
10^{-2}	0.008	57.9	0.008	46.0
10^{-3}	0.007	54.8	0.014	41.6
10^{-4}	0.027	52.5	0.029	38.5
10^{-5}	0.053	51.1	0.059	36.6
10^{-6}	0.088	50.8	0.099	36.2
10^{-7}	0.077	50.3	0.151	35.7
10^{-8}	0.182	49.9	0.207	35.3
10^{-9}	0.165	49.6	0.301	34.9
10^{-10}	0.216	49.5	0.298	34.8
10^{-11}	0.318	49.5	0.483	34.3
10^{-12}	0.409	49.3	0.485	34.4

needed to type the reference translation using the auto-typing facility divided by the number of key-strokes needed to type the reference translation. For using the auto-typing facility, we make the assumption that the user can accept an arbitrary length of the presented extension using a single key-stroke. Hence, a key-stroke ratio of 1 means that the system was never able to suggest a correct extension. A very small key-stroke ratio means that the suggested extension is often correct. This value gives an indication about the possible effective gain that can be achieved if this is used in a real translation task. While the key-stroke ratio is overly optimistic with respect to the efficiency gain of the user, it has the advantage of being a well-defined objective criterion. We expect it to be well correlated to a more user-centered evaluation criterion.

Table 10.1 shows the resulting key-stroke ratio and the average extension time for various pruning thresholds which lead to word graphs of different density for single-word extension and whole-sentence extension.

We see that by using a smaller pruning threshold, a significantly larger time is needed to search in the resulting word graph. Yet, also the KSR improves significantly. In the case of single-word extension, the KSR improves from 61.2% and 0.005 seconds per extension to 49.3% and 0.409 seconds per extension. Significantly better results are obtained, if we perform whole-sentence extension. Here, the KSR improves from 50.9% and 0.005 seconds per extension to 34.4% and 0.485 seconds per extension. Hence, we conclude that using the word graph we can indeed efficiently represent the relevant search space. In addition, we conclude that whole-sentence extension gives significantly better results than only single-word extension.

Chapter 11

Conclusion

11.1 Summary

The aim of this work has been to extend the state-of-the-art in MT by developing new statistical translation models, efficient training and search algorithms. In addition, new innovative applications for statistical MT have been developed. Especially, the following scientific contributions have been achieved:

- We have described in detail the development of a statistical MT system in all its aspects ranging from data collection, preprocessing, modeling, training and search. Using the *evolutionary rapid prototyping paradigm*, various successful statistical MT systems have been developed.

- We have provided a quantitative comparison of various word alignment models. In addition, new models and new efficient training algorithms have been developed. A new statistical alignment model—Model 6—has been suggested, specific training algorithms and new methods for using a conventional dictionary in training have been developed. These methods have led to significantly increased alignment quality. Using heuristic symmetrization algorithms that combine the alignments in both translation directions, it has been possible to overcome the limitations of baseline alignment models, which do not allow for one-to-many alignments.

- We have suggested to develop statistical machine translation systems based on a direct model for the posterior probability using maximum entropy models. This approach contains the conventional source–channel approach as a special case. This approach allows not only a better exploitation of conventional translation models, but also allows extending statistical MT systems easily by adding new feature functions. Using this approach on the VERBMOBIL task, we have achieved significant improvements.

- The alignment template approach, a new approach to phrase-based statistical MT, has been developed. This approach combines the advantages of statistical alignment models and the use of whole phrases as in an example-based approach. In various evaluations, this statistical translation model produces significantly better results than other state-of-the-art statistical translation models. In the speech translation evaluations of the VERBMOBIL project, the developed alignment template approach yielded significantly better results than four competing systems.

- In the literature, various search algorithms have been proposed to deal with the search problem in statistical MT. In left-to-right search algorithms, the hypotheses are formed in increasing length. Typically, the scoring of the search hypotheses takes into account only the current translation prefix probability. We have proposed to improve search efficiency by including an admissible heuristic function, which estimates the contribution of the remaining probability that are needed to produce a complete translation. We have developed refined admissible and almost admissible heuristic functions for statistical MT. The developed heuristic function has a strong effect on search efficiency.

- We have suggested a new method for using multiple source languages to produce a better translation in a new language. The framework of statistical MT allows a very concise formulation of this problem. We have shown that translation quality can be significantly improved by using more source languages.

- We have suggested an interactive MT environment, which supports the human translator by interactively reacting to user input providing an *auto-typing* facility that suggests the human translator a complete extension of a sentence. Here, word graphs are used to allow an efficient search for the optimal extension. Using this method, the amount of key-strokes needed to produce the reference translation reduces significantly.

11.2 Outlook

Automated collection of training data

A key element in the empirical approach to MT is the collection of large amounts of useful training data. A very interesting approach is the idea of automatically collecting large amounts of training data from the Internet [Resnik 99]. The EU Bulletin Corpus used in this thesis has been collected automatically from the Internet. In the near future, we might think of systems that are completely bootstrapped automatically from the Internet for any language pair that exists in the Internet in sufficient amounts.
One of the problems of the data collected in this way is that there are frequently contained wrong translations, omissions and other noise. To use these data, we have to perform robust sentence alignment, automatic detection and filtering of wrong translation examples.

Integrated speech translation using maximum entropy framework

Typically, speech-to-speech translation is performed using a serial coupling of a speech recognizer and a translation system. Yet, analyzing the problem from the viewpoint of statistical decision theory, an integrated approach would be desirable [Ney 99]. The probability distributions of translation and recognition can interact and the errors in the best hypothesis of the speech recognizer not necessarily lead to errors in the final translation.
The experiments in the EUTRANS project [Vidal et al. 00] have shown that integrated speech translation yields improvements on simple tasks, but for large tasks, the translation quality significantly deteriorates with respect to a serial coupling. A possible reason is that, due to the pursued approach, different models have to be used for the serial coupling and for the integrated search.

Therefore, a very promising approach would be the application of the here suggested maximum entropy framework. In this framework, we can use in addition to the standard language and translation model features also features that come from the acoustic model and from the source language model.

Refined statistical models

The alignment template approach developed in this thesis has outperformed other translation approaches in various evaluations in VERBMOBIL, EUTRANS and other projects. Yet, the obtained translation quality still leaves much to be desired. We expect that better translation can be achieved by using refined statistical language and translation models:

- There is a need for statistical models that are better suited for the recursive structure of natural languages. Current statistical MT systems have problems with nonlocal phenomena, i.e. dependencies between nonconsecutive words. As a result, the target language sentence often contains syntactic errors. There are only a few approaches that try to deal with this problem [Wu & Wong 98, Alshawi & Bangalore$^+$ 98, Wang & Waibel 98, Yamada & Knight 01]. So far, the success of more linguistic oriented models is limited.

  Yet, the recent success of lexicalized grammars in language modeling for speech recognition [Chelba 00, Roark 01, Charniak 01] gives new hope. First, it would be interesting to analyze the effect of these grammar-based language models on translation quality. We expect that these models have a more significant effect in translation than they have in speech recognition because of the reordering problem in translation. Second, the availability of high-quality grammars in source and target language allows for grammar-based translation models, which can be used in addition to existing translation models.

- The knowledge bases obtained for rule-based MT systems or other linguistic resources should be exploited as part of the statistical approach. This means that in addition to the bilingual corpus used to train the translation model, additional knowledge sources (e.g. parallel tree banks, WordNet or annotated dictionaries) are used in the training of refined translation models. This is more promising than the standard approach where different translation approaches are used in parallel and are combined in a postprocessing step [Nirenburg & Frederking 94, Cavar & Küssner$^+$ 00]. We suggest integrating the additional knowledge sources by specifying appropriate features using the maximum entropy framework described in this thesis.

- Statistical translation systems typically ignore the context in which a sentence appears. This means that for example anaphora are normally translated by their most probable translation. This is a source of systematic errors. In addition, there is no dependence on the text structure or the dialogue act in speech translation.

  The rhetorical structure theory (RST) [Mann & Thompson 87] is the field in linguistics that tries to describe the structure of texts. For MT purposes, it would be interesting to develop a practical variant of a rhetorical structure theory model, which deals with the specific problems relevant for statistical MT.

Standard evaluation environment and training corpora

The current situation in MT evaluation leaves much to be desired. Even though there is many literature on MT evaluation, there is no measure available that is generally accepted. As a result, comparing research results is very hard. An ideal evaluation criterion would produce a one-dimensional score that can be easily interpreted and would be computed automatically. In addition, it should be strongly related to human subjective evaluation scores. There have been suggested various evaluation criteria that seem to meet these criteria (Section 3.5). For example, the multi-reference word error rate (mWER) or the BLEU score seem to be well suited. Yet, so far no standard evaluation criterion exists, which is used by the whole research community.
An additional reason that research results are incomparable is the lack of standard training and test corpora. More efforts are needed to produce such corpora and to make these corpora freely available to interested research groups. An extremely useful knowledge source would be the parallel texts in the eleven official European Union languages that are available at the European institutions such as the European Commission, the European Parliament or the European Court. The availability of parallel texts with billions of words in these languages would be an enormous stimulus for the research community.

Appendix A

Additional Results

In this appendix, we present some additional results that have been obtained using the alignment template approach described in this thesis. As reference system, we typically use the single-word based approach based on Model 4 described in [Tillmann & Ney 00, Tillmann 01].

A.1 EUTRANS-I task

The EUTRANS-I task is a subtask of the "Traveler Task" [Vidal 97] for which semi-automatically generated Spanish–English corpus is available. The domain of the corpus consists of human-to-human communication situation at a reception desk of a hotel. For this task a categorization procedure exists, which replaces numbers, dates and names by category labels. A summary of the corpus used in the experiments with and without categorization is given in Table A.1.

Table A.2 shows the results of the alignment template approach compared to the single-word based approach.

Table A.1: Corpus statistics of EUTRANS-I task(Spanish $\rightarrow$ English, Words*: words without puncutation marks).

		Without Categorization		With Categorization	
		Spanish	English	Spanish	English
Train:	Sentences	10 000			
	Words	97 131	99 292	92 240	94 664
	Words*	78 783	85 797	73 892	81 169
	Vocabulary	686	513	410	220
	Singletons	3	0	3	0
Test:	Sentences	2 996			
	Words	35 023	35 590	32 356	33 078
	Trigram PP	–	3.3	–	2.5

Table A.2: Translation results on the EUTRANS-I task.

	Categorization	WER[%]	PER[%]
Alignment Templates	N	4.4	2.9
	Y	2.5	1.9
Single-Word Based Approach	N	10.8	10.0
	Y	6.7	6.0

Table A.3: Corpus statistics of EUTRANS-II speech task (Italian $\rightarrow$ English, Words*: words without puncutation marks).

		Italian	English
Train:	Sentences	3 038	
	Words	55 302	65 446
	Words*	47 464	57 446
	Vocabulary Size	2 459	1 701
Test(Text):	Sentences	300	
	Words	6 121	7 243
	Trigram PP	–	17.8
Test(Speech):	Words	6 121	7 243
	WER [%]	33.4	–

A.2 EUTRANS-II speech task

The EUTRANS-II speech task is an Italian–English corpus collected in the EUTRANS-project. It consists of transcriptions of spoken dialogues in the framework of hotel reception desk person-to-person communication. A summary of the corpus used in the experiments is given in Table A.3.
Table A.4 show the results on this corpus.

Table A.4: Translation results on the EUTRANS-II speech task.

Language pair	Method	WER[%]	PER[%]
Text	Alignment Templates	25.1	19.0
Speech	Alignment Templates	41.5	32.8

Appendix B

Free Software

During the development of this thesis, various software tools have been developed. In the hope that they are useful for other researchers, I have made some of this software publically available under the GNU Public License (GPL). All software is written in C++ and is extensively tested under the Linux operating system.
The following tools can be downloaded from my web page in the Internet[1]:

- `mkcls` [Och 00b]: This is a program that has originally been developed as part of my pre-diploma thesis at the University of Erlangen-Nuremberg [Och 95]. This tool allows to train word classes by using a maximum likelihood criterion [Kneser & Ney 91, Kneser & Ney 93]. The resulting word classes are especially suited for language models. This program is used in this thesis to train bilingual word classes (Chapter 7).

- `GIZA++` [Och 00a]: This software package is an extension of the `GIZA` program, which is part of the `EGYPT` statistical MT toolkit [Al-Onaizan & Curin$^+$ 99]. `GIZA` has been developed by a group of eleven researchers including myself at the 1999 Johns Hopkins University summer workshop on statistical MT. As part of an NSF-sponsored follow-up research project, I have performed significant extensions to the program. `GIZA++` includes the following extensions to `GIZA`:

  - Model 4 and Model 5 training
  - alignment models depending on word classes
  - Hidden Markov alignment model: Baum–Welch training, Forward-Backward algorithm, empty word, dependence on word classes, transfer to fertility-based alignment models
  - variants of Model 3 and Model 4
  - various smoothing methods for fertility and alignment parameters;
  - significant more efficient training of the fertility-based alignment models;
  - correct implementation of pegging as described in [Brown & Della Pietra$^+$ 93b], a series of heuristics in order to make pegging sufficiently efficient

[1] My web page: `http://www-i6.informatik.rwth-aachen.de/~och`

- YASMET [Och 01]: This is a tiny toolkit for performing training of conditional maximum entropy models. It includes training of model parameters, evaluation, perplexity and error rate computation, count-based feature reduction, smoothing with Gaussian priors and feature count normalization. In addition, YASMET is efficient enough to deal with millions of features.

Appendix C

Efficient Training of Fertility Models

In this appendix, we describe some methods for efficient training of fertility-based alignment models. The core idea is to enumerate only a small subset of good alignments in the E-step of the EM algorithm instead of enumerating all $(I+1)^J$ alignments. This small subset of alignments is the set of neighboring alignments of the best alignment that can be found by a greedy-search algorithm. We use the following operators that transform alignments: the move operator $m_{[i,j]}(\mathbf{a})$ changes $a_j := i$ and the swap operator $s_{[j_1,j_2]}(\mathbf{a})$ exchanges a_{j_1} and a_{j_2}. The neighborhood $\mathcal{N}(\mathbf{a})$ of an alignment $\mathbf{a}$ is then defined as the set of all alignments that differ by one move or one swap from alignment $\mathbf{a}$:

$$\mathcal{N}(\mathbf{a}) = \{\mathbf{a}' : \exists_{i,j} : \mathbf{a}' = m_{[i,j]}(\mathbf{a}) \vee \exists_{j_1,j_2} : \mathbf{a}' = s_{[j_1,j_2]}(\mathbf{a})\} \tag{C.1}$$

The hill climbing operator (of Model 3) is then defined as follows:

$$b(\mathbf{a}) = \underset{\mathbf{a}' \in \mathcal{N}(\mathbf{a})}{\operatorname{argmax}}\ p_3(\mathbf{a}'|\mathbf{e},\mathbf{f}) \tag{C.2}$$

Similarly, we define a hill climbing operator for the other alignment models.

Straightforward implementation

A straightforward count collection procedure for a sentence pair (**f**,**e**) following the description in [Brown & Della Pietra$^+$ 93b] is:[1]

1. Calculate the Viterbi alignment of Model 2: $\mathbf{a}_0 := \operatorname{argmax}_{\mathbf{a}}\ p_2(\mathbf{f},\mathbf{a}|\mathbf{e})$, $n := 0$
2. While in the neighborhood $\mathcal{N}(\mathbf{a}_n)$ an alignment $\mathbf{a}'$ exists with $p_3(\mathbf{a}'|\mathbf{e},\mathbf{f}) > p_3(\mathbf{a}_n|\mathbf{e},\mathbf{f})$:
   (a) Set $\mathbf{a_{n+1}}$ to the best alignment in the neighborhood.
   (b) $n := n+1$.
3. Calculate

$$s := \sum_{\mathbf{a} \in \mathcal{N}(\mathbf{a_n})} Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) \tag{C.3}$$

[1] To simplify the description, we ignore the process called "pegging" that generates a bigger number of alignments considered in training.

4. For each alignment $\mathbf{a}$ in the neighborhood $\mathcal{N}(\mathbf{a}_n)$

   (a) Calculate

$$p := Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \tag{C.4}$$

$$= \frac{Pr(\mathbf{f},\mathbf{a}|\mathbf{e})}{s} \tag{C.5}$$

   (b) For each $j := 1$ to J: Increase alignment counts

$$c(j|a_j,m,l;\mathbf{e},\mathbf{f}) := c(j|a_j,m,l;\mathbf{e},\mathbf{f}) + p \tag{C.6}$$

   (c) For each $i := 1$ to I: Increase the fertility counts with p:

$$c(\phi_i|e_i;\mathbf{e},\mathbf{f}) := c(\phi_i|e_i;\mathbf{e},\mathbf{f}) + p \tag{C.7}$$

   (d) Increase the counts for p_1:

$$c(1;\mathbf{e},\mathbf{f}) := c(1;\mathbf{e},\mathbf{f}) + p \cdot \phi_0 \tag{C.8}$$

A major part of the time in this procedure is spent on calculating the probability $Pr(\mathbf{a}'|\mathbf{e},\mathbf{f})$ of an alignment $\mathbf{a}'$. In general, this takes about $(I+J)$ operations. [Brown & Della Pietra$^+$ 93b] describe a method for obtaining $Pr(\mathbf{a}'|\mathbf{e},\mathbf{f})$ incrementally from $Pr(\mathbf{a}|\mathbf{e},\mathbf{f})$ if alignment $\mathbf{a}$ differs only by moves or swaps from alignment $\mathbf{a}'$. This trick results in a constant number of operations that are sufficient to calculate the score of a move or the score of a swap.

Refined implementation: fast hill climbing

Analyzing the training program reveals that most of the time is spent on the computation of the moves and swaps. To reduce the number of operations, these values are cached in two matrices. We use one matrix for the scores of a move $a_j := i$:

$$M_{i,j} = \frac{Pr(m_{[i,j]}(\mathbf{a})|\mathbf{e},\mathbf{f})}{Pr(\mathbf{a}|\mathbf{e},\mathbf{f})} \cdot (1 - \delta(a_j,i)) \tag{C.9}$$

and an additional matrix for the scores of a swap of a_j and $a_{j'}$:

$$S_{j,j'} = \begin{cases} \dfrac{Pr(s_{[j,j']}(\mathbf{a})|\mathbf{e},\mathbf{f})}{Pr(\mathbf{a}|\mathbf{e},\mathbf{f})} \cdot (1 - \delta(a_j,a_{j'})) & \text{if } j < j' \\ 0 & \text{otherwise} \end{cases} \tag{C.10}$$

During the hill climbing, it is sufficient, after doing a move or a swap, to update only those rows or columns in the matrix that are affected by the move or swap. In such a way, the number of operations in hill climbing can be reduced by about one order of magnitude. For example, when performing a move $a_j := i$, it is necessary to:

- update in matrix M the columns j' with $a_{j'} = a_j$ or $a_{j'} = i$,
- update in matrix M the rows a_j and i,
- update in matrix S the rows and columns j' with $a_{j'} = a_j$ or $a_{j'} = i$.

Similar updates have to be performed after a swap. In count collection (step 3) can be used the matrices obtained in the final hill climbing step.

Refined implementation: fast count collection

The given straightforward algorithm for performing the count collection has the disadvantage that all alignments in the neighborhood of alignment $\mathbf{a}$ have to be enumerated explicitly. In addition, a loop over all target and a loop over all source positions to update the lexicon, alignment and fertility counts is necessary. To perform the count collection in an efficient way, we use that the alignments in the neighborhood $\mathcal{N}(\mathbf{a})$ are very similar. This allows the sharing of many operations in the count collection process.
To efficiently obtain the alignment and lexicon probability counts, we introduce the following auxiliary quantities that use the move and swap matrices that are available after performing hill climbing:

- probability of all alignments in the neighborhood $\mathcal{N}(\mathbf{a})$:

$$Pr(\mathcal{N}(\mathbf{a})|\mathbf{e},\mathbf{f}) = \sum_{\mathbf{a}' \in \mathcal{N}(\mathbf{a})} Pr(\mathbf{a}'|\mathbf{e},\mathbf{f}) \tag{C.11}$$

$$= Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \cdot \left(1 + \sum_{i,j} M_{i,j} + \sum_{j,j'} S_{j,j'}\right) \tag{C.12}$$

- probability of all alignments in the neighborhood $\mathcal{N}(\mathbf{a})$ that differ in position j from alignment $\mathbf{a}$:

$$Pr(\mathcal{N}_j(\mathbf{a})|\mathbf{e},\mathbf{f}) = \sum_{\mathbf{a}' \in \mathcal{N}(\mathbf{a})} Pr(\mathbf{a}'|\mathbf{e},\mathbf{f})(1 - \delta(a_j, a'_j)) \tag{C.13}$$

$$= Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \left(\sum_{i} M_{i,j} + \sum_{j'} (S_{j,j'} + S_{j',j})\right) \tag{C.14}$$

For the alignment counts $c(j|i;\mathbf{e},\mathbf{f})$ and the lexicon counts $c(f|e;\mathbf{e},\mathbf{f})$, we have:

$$c(j|i;\mathbf{e},\mathbf{f}) = \begin{cases} Pr(\mathcal{N}(\mathbf{a})|\mathbf{e},\mathbf{f}) - Pr(\mathcal{N}_j(\mathbf{a})|\mathbf{e},\mathbf{f}) & \text{if } i = a_j \\ Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \left(M_{i,j} + \sum_{j'} \delta(a_{j'}, i) \cdot (S_{j,j'} + S_{j',j})\right) & \text{if } i \neq a_j \end{cases} \tag{C.15}$$

$$c(f|e;\mathbf{e},\mathbf{f}) = \sum_{i} \sum_{j} c(j|i;\mathbf{e},\mathbf{f}) \cdot \delta(f, f_j) \cdot \delta(e, e_i) \tag{C.16}$$

To efficiently obtain the fertility probability counts and the count for p_1, we introduce the following auxiliary quantities:

- probability of all alignments that have an increased fertility for position i:

$$Pr(\mathcal{N}_i^{+1}(\mathbf{a})|\mathbf{e},\mathbf{f}) = Pr(\mathbf{a}|\mathbf{f},\mathbf{e}) \left(\sum_{j} (1 - \delta(a_j, i)) \cdot M_{i,j}\right) \tag{C.17}$$

- probability of all alignments that have a decreased fertility for position i:

$$Pr(\mathcal{N}_i^{-1}(\mathbf{a})|\mathbf{e},\mathbf{f}) = Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \left(\sum_{j} \delta(a_j, i) \sum_{i'} M_{i',j}\right) \tag{C.18}$$

- probability of all alignments that have an unchanged fertility for position i:

$$Pr(\mathcal{N}_i^{+0}(\mathbf{a})|\mathbf{e},\mathbf{f}) = Pr(\mathcal{N}(\mathbf{a})|\mathbf{e},\mathbf{f}) - Pr(\mathcal{N}_i^{+1}(\mathbf{a})|\mathbf{e},\mathbf{f}) - Pr(\mathcal{N}_i^{-1}(\mathbf{a})|\mathbf{e},\mathbf{f}) \quad \text{(C.19)}$$

These quantities do not depend on swaps, because a swap does not change the fertilities of an alignment. For the fertility counts, we have:

$$c(\phi|e;\mathbf{e},\mathbf{f}) = \sum_i \delta(e,e_i) \sum_k Pr(\mathcal{N}_i^{+k}(\mathbf{a})|\mathbf{e},\mathbf{f})\delta(\phi_i + k, \phi) \quad \text{(C.20)}$$

For p_1, we have:

$$c(1;\mathbf{e},\mathbf{f}) \quad = \quad \sum_k Pr(\mathcal{N}_0^{+k}(\mathbf{a})|\mathbf{e},\mathbf{f})(\phi_0 + k) \quad \text{(C.21)}$$

Using the auxiliary quantities, a count collection algorithm can be formulated that requires about $O(\max(I,J)^2)$ operations. This is one order of magnitude faster than the described straightforward algorithm. In practice, we observe that the resulting training is about 10-20 times faster.

Bibliography

[Al-Onaizan & Curin$^+$ 99] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J.D. Lafferty, I.D. Melamed, D. Purdy, F.J. Och, N.A. Smith, D. Yarowsky. Statistical machine translation, final report, JHU workshop, 42 pages, 1999. `http://www.clsp.jhu.edu/ws99/projects/mt/final_report/mt-final-report.ps`.

[Alshawi & Bangalore$^+$ 98] H. Alshawi, S. Bangalore, S. Douglas: Automatic acquisition of hierarchical transduction models for machine translation. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, Vol. 1, pp. 41–47, Montreal, Canada, Aug. 1998.

[Alshawi & Bangalore$^+$ 00] H. Alshawi, S. Bangalore, S. Douglas: Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, Vol. 26, No. 1, pp. 45–60, 2000.

[Arnold & Balkan$^+$ 94] D. Arnold, L. Balkan, S. Meijer, L.L. Humphreys, L. Sadler: *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, Great Britain, 1994.

[Auerswald 00] M. Auerswald. Example-based machine translation with templates. In [Wahlster 00], pp. 418–427.

[Baum 72] L.E. Baum: An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, Vol. 3, pp. 1–8, 1972.

[Becker & Kilger$^+$ 00] T. Becker, A. Kilger, P. Lopez, P. Poller. The Verbmobil generation component VM-GECO. In [Wahlster 00], pp. 481–496.

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

[Berger & Brown$^+$ 94] A.L. Berger, P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.R. Gillett, J.D. Lafferty, H. Printz, L. Ureš: The Candide system for machine translation. In *Proc. ARPA Workshop on Human Language Technology*, pp. 157–162, Plainsboro, NJ, March 1994.

[Berger & Brown$^+$ 96] A.L. Berger, P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.R. Gillett, A.S. Kehler, R.L. Mercer. Language translation apparatus and method of using context-based translation models. United States Patent, No. 5510981, 75 pages, April 1996.

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

[Block 00] H.U. Block. Incremental synchronous interpretation. In [Wahlster 00], pp. 411–417.

[Brown 97] R.D. Brown: Automated dictionary extraction for "knowledge-free" example-based translation. In *Seventh Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI-97)*, pp. 111–118, Santa Fe, NM, July 1997.

[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+ 92] P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, R.L. Mercer: Class-based n-gram models of natural language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, 1992.

[Brown & Della Pietra+ 93a] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, M.J. Goldsmith, J. Hajic, R.L. Mercer, S. Mohanty: But dictionaries are data too. In *Proc. ARPA Workshop on Human Language Technology*, pp. 202–205, Plainsboro, NJ, March 1993.

[Brown & Della Pietra+ 93b] 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, 1993.

[Castellanos & Galiano+ 94] A. Castellanos, I. Galiano, E. Vidal. Application of OSTIA to machine translation tasks. In R.C. Carrasco, J. Oncina, editors, *Grammatical Inference and Applications, Proc. of 2nd ICGI*, Vol. 862 of *Lecture Notes in Computer Science*, pp. 93–105. Springer-Verlag, Alicante, Spain, 1994.

[Cavar & Küssner+ 00] D. Cavar, U. Küssner, D. Tidhar. From off–line evaluation to on–line selection. In [Wahlster 00], pp. 599–612.

[Chandioux & Grimaila 96] J. Chandioux, A. Grimaila: Specialized machine translation. In *2nd Conf. of the Association for Machine Translation in the Americas (AMTA 96)*, pp. 206–212, Montreal, Canada, Oct. 1996.

[Charniak 01] E. Charniak: Immediate-head parsing for language models. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 116–123, Toulouse, France, July 2001.

[Chelba 00] C. Chelba. *Exploiting Syntactic Structure for Natural Language Modeling*. Ph.D. thesis, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, 2000.

[Cole & Mariani+ 95] R.A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, V. Zue: *Survey of the State of the Art in Human Language Technology*. Carnegie Mellon University, Pittsburgh, PA, Nov. 1995.

[Connell & Shafer 94] J. Connell, L. Shafer: *Object-Oriented Rapid Prototyping*. Prentice Hall, Englewood Cliffs, NJ, Oct. 1994.

[Dagan & Church$^+$ 93] I. Dagan, K.W. Church, W.A. Gale: Robust bilingual word alignment for machine aided translation. In *Proc. of the Workshop on Very Large Corpora*, pp. 1–8, Columbus, OH, June 1993.

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

[Della Pietra & Epstein$^+$ 97] S.A. Della Pietra, M. Epstein, S. Roukos, T. Ward: Fertility models for statistical natural language understanding. In *Proc. 35th Annual Conf. of the Association for Computational Linguistics*, pp. 168–173, Madrid, Spain, July 1997.

[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.

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

[Dice 45] L.R. Dice: Measures of the amount of ecologic association between species. *Journal of Ecology*, Vol. 26, pp. 297–302, 1945.

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

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

[Duda & Hart$^+$ 00] R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*. John Wiley and Sons, New York, NY, 2nd edition, 2000.

[Dueck & Scheuer 90] G. Dueck, T. Scheuer: Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, Vol. 90, No. 1, pp. 161–175, 1990.

[Emele & Dorna$^+$ 00] M.C. Emele, M. Dorna, A. Lüdeling, H. Zinsmeister, C. Rohrer. Semantic-based transfer. In [Wahlster 00], pp. 359–376.

[Epstein & Papineni$^+$ 96] M. Epstein, K. Papineni, S. Roukos, T. Ward, S.A. Della Pietra: Statistical natural language understanding using hidden clumpings. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 176–179, Atlanta, GA, May 1996.

[Foster 00a] G. Foster: Incorporating position information into a maximum entropy/minimum divergence translation model. In *Fourth Conf. on Computational Language Learning (CoNLL)*, pp. 37–52, Lisbon, Portugal, Sept. 2000.

[Foster 00b] G. Foster: A maximum entropy/minimum divergence translation model. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 37–44, Hong Kong, Oct. 2000.

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

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

[Fukunaga 90] K. Fukunaga: *Introduction to Statistical Pattern Recognition*. Academic Press, New York, NY, 1990.

[Gale & Church 93] W.A. Gale, K.W. Church: A program for aligning sentences in bilingual corpora. *Computational Linguistics*, Vol. 19, No. 1, pp. 75–90, 1993.

[García-Varea & Casacuberta$^+$ 98] I. García-Varea, F. Casacuberta, H. Ney: An iterative, DP-based search algorithm for statistical machine translation. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP'98)*, pp. 1235–1238, Sydney, Australia, Nov. 1998.

[García-Varea & Casacuberta 01] I. García-Varea, F. Casacuberta: Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proc. of Machine Translation Summit VIII*, pp. 115–120, Santiago de Compostela, Spain, Sept. 2001.

[García-Varea & Och$^+$ 01] I. García-Varea, F.J. Och, H. Ney, F. Casacuberta: Refined lexicon models for statistical machine translation using a maximum entropy approach. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 204–211, Toulouse, France, July 2001.

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

[Haas & Hornegger$^+$ 97] J. Haas, J. Hornegger, R. Huber, H. Niemann: Probabilistic semantic analysis of speech. In *19. Symposium der Deutschen Arbeitsgemeinschaft für Mustererkennung*, pp. 270–277, Braunschweig, Germany, Sept. 1997.

[Hovy 99] E. Hovy: Toward finely differentiated evaluation metrics for machine translation. In *Proc. of the EAGLES Workshop on Standards and Evaluation*, pp. 127–133, Pisa, Italy, 1999.

[Huang & Choi 00] J.X. Huang, K.S. Choi: Chinese–Korean word alignment based on linguistic comparison. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 392–399, Hong Kong, Oct. 2000.

[Hutchins 95] J. Hutchins: Reflections on the history and present state of machine translation. In *Proc. of Machine Translation Summit V*, pp. 89–96, Luxembourg, July 1995.

[Hutchins & Somers 92] W.J. Hutchins, H.L. Somers: *An Introduction to Machine Translation*. Academic Press, Cambridge, MA, 1992.

[Jelinek 69] F. Jelinek: A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, Vol. 13, pp. 675–685, 1969.

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

[Jones & Rusk 00] D.A. Jones, G.M. Rusk: Toward a scoring function for quality-driven machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 376–382, Saarbrücken, Germany, Aug. 2000.

[Jurafsky & Martin 00] D. Jurafsky, J.H. Martin: *Speech and Language Processing*. Prentice Hall, Englewood Cliffs, NJ, 2000.

[Ker & Chang 97] S.J. Ker, J.S. Chang: A class-based approach to word alignment. *Computational Linguistics*, Vol. 23, No. 2, pp. 313–343, 1997.

[Kneser & Ney 91] R. Kneser, H. Ney: Forming word classes by statistical clustering for statistical language modelling. In *1. Quantitative Linguistics Conf.*, pp. 221–226, Trier, Germany, Sept. 1991.

[Kneser & Ney 93] R. Kneser, H. Ney: Improved clustering techniques for class-based statistical language modelling. In *European Conf. on Speech Communication and Technology*, pp. 973–976, Berlin, Germany, Sept. 1993.

[Knight 99a] K. Knight: Decoding complexity in word-replacement translation models. *Computational Linguistics*, Vol. 25, No. 4, pp. 607–615, 1999.

[Knight 99b] K. Knight. A statistical machine translation tutorial workbook, 35 pages, Aug. 1999. `http://www.isi.edu/natural-language/mt/wkbk.rtf`.

[Koenig & Moo 97] A. Koenig, B. Moo: *Ruminations on C++*. Addison-Wesley Publishing Company, Reading, MA, 1997.

[Langenscheidt-Redaktion 96] Langenscheidt-Redaktion: *Langenscheidts Großes Schulwörterbuch Englisch-Deutsch*. Langenscheidt KG, Berlin, Germany, 1996.

[Langlais & Foster$^+$ 00] P. Langlais, G. Foster, G. Lapalme: TransType: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems*, pp. 46–51, Seattle, WA, May 2000.

[Macherey & Och$^+$ 01] K. Macherey, F.J. Och, H. Ney: Natural language understanding using statistical machine translation. In *European Conf. on Speech Communication and Technology*, pp. 2205–2208, Aalborg, Denmark, Sept. 2001.

[Manber & Myers 90] U. Manber, E. Myers: Suffix arrays: A new method for on-line string searches. In *Proc. of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 319–327, San Francisco, CA, Jan. 1990.

[Mann & Thompson 87] W.C. Mann, S.A. Thompson. Rhetorical structure theory: A theory of text organization. Technical Report RS-87-190, Information Sciences Institute, Los Angeles, CA, 82 pages, 1987.

[Martin & Liermann$^+$ 98] S. Martin, J. Liermann, H. Ney: Algorithms for bigram and trigram word clustering. *Speech Communication*, Vol. 24, No. 1, pp. 19–37, 1998.

[Melamed 98] I.D. Melamed. Manual annotation of translational equivalence: The blinker project. Technical Report 98-07, Institute for Research in Cognitive Science, Philadelphia, PE, 13 pages, 1998.

[Melamed 00] I.D. Melamed: Models of translational equivalence among words. *Computational Linguistics*, Vol. 26, No. 2, pp. 221–249, 2000.

[Miller & Bobrow$^+$ 94] S. Miller, R. Bobrow, R. Ingria, R. Schwartz: Hidden understanding models of natural language. In *Proc. of the 32nd Annual Conf. of the Association for Computational Linguistics*, pp. 25–32, Las Cruces, NM, June 1994.

[Ney 95] H. Ney: On the probabilistic-interpretation of neural-network classifiers and discriminative training criteria. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, pp. 107–119, Feb. 1995.

[Ney 99] H. Ney: Speech translation: Coupling of recognition and translation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 517–520, Phoenix, AR, March 1999.

[Ney & Aubert 94] H. Ney, X. Aubert: A word graph algorithm for large vocabulary continuous speech recognition. In *Proc. Int. Conf. on Spoken Language Processing*, pp. 1355–1358, Yokohama, Japan, Sept. 1994.

[Ney & Generet$^+$ 95] H. Ney, M. Generet, F. Wessel: Extensions of absolute discounting for language modeling. In *Proc. of the Fourth European Conf. on Speech Communication and Technology*, pp. 1245–1248, Madrid, Spain, Sept. 1995.

[Ney & Mergel$^+$ 87] H. Ney, D. Mergel, A. Noll, A. Paeseler: A data-driven organization of the dynamic programming beam search for continuous speech recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 833–836, Dallas, TX, April 1987.

[Ney & Nießen$^+$ 00] H. Ney, S. Nießen, F.J. Och, H. Sawaf, C. Tillmann, S. Vogel: Algorithms for statistical translation of spoken language. *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 1, pp. 24–36, Jan. 2000.

[Ney & Och$^+$ 00] H. Ney, F.J. Och, S. Vogel: Statistical translation of spoken dialogues in the verbmobil system. In *Workshop on Multi-Lingual Speech Communication*, pp. 69–74, Kyoto, Japan, Oct. 2000.

[Ney & Och$^+$ 01] H. Ney, F.J. Och, S. Vogel: The RWTH system for statistical translation of spoken dialogues. In *Proc. ARPA Workshop on Human Language Technology*, San Diego, CA, March 2001.

[Niemann 90] H. Niemann: *Pattern Analysis and Understanding*. Springer Verlag, Berlin, Germany, 1990.

[Nießen & Ney 00] S. Nießen, H. Ney: Improving SMT quality with morpho-syntactic analysis. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 1081–1085, Saarbrücken, Germany, July 2000.

[Nießen & Ney 01a] S. Nießen, H. Ney: Morpho-syntactic analysis for reordering in statistical machine translation. In *Proceedings of the Machine Translation Summit VIII*, pp. 247–252, Santiago de Compostela, Spain, Sept. 2001.

[Nießen & Ney 01b] S. Nießen, H. Ney: Toward hierarchical models for statistical machine translation of inflected languages. In *Data-Driven Machine Translation Workshop*, pp. 47–54, Toulouse, France, July 2001.

[Nießen & Och$^+$ 00] S. Nießen, F.J. Och, G. Leusch, H. Ney: An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 39–45, Athens, Greece, May 2000.

[Nießen & Vogel$^+$ 98] S. Nießen, S. Vogel, H. Ney, C. Tillmann: A DP-based search algorithm for statistical machine translation. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 960–967, Montreal, Canada, Aug. 1998.

[Nilsson 82] N.J. Nilsson: *Principles of Artificial Intelligence*. Springer Verlag, Berlin, Germany, 1982.

[Nirenburg & Frederking 94] S. Nirenburg, R. Frederking: Toward multi-engine machine translation. In *Proc. ARPA Workshop on Human Language Technology*, pp. 147–151, Plainsboro, NJ, March 1994.

[Och 95] F.J. Och. Maximum-Likelihood-Schätzung von Wortkategorien mit Verfahren der kombinatorischen Optimierung. Studienarbeit, Friedrich-Alexander-Universität, Erlangen-Nürnberg, Germany, July 1995.

[Och 98] F.J. Och. Ein beispielsbasierter und statistischer Ansatz zum maschinellen Lernen von natürlichsprachlicher Übersetzung. Diploma thesis, Friedrich-Alexander-Universität, Erlangen-Nürnberg, Germany, March 1998.

[Och 99] F.J. Och: An efficient method for determining bilingual word classes. In *EACL '99: Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics*, pp. 71–76, Bergen, Norway, June 1999.

[Och 00a] F.J. Och. Giza++: Training of statistical translation models, 2000. `http://www-i6.informatik.rwth-aachen.de/~och/software/GIZA++.html`.

[Och 00b] F.J. Och. mkcls: Training of word classes for language modeling, 2000. `http://www-i6.informatik.rwth-aachen.de/~och/software/mkcls.html`.

[Och 01] F.J. Och. YASMET: Toolkit for conditional maximum entropy models, 2001. `http://www-i6.informatik.rwth-aachen.de/~och/software/YASMET.html`.

[Och & Ney 00a] F.J. Och, H. Ney: A comparison of alignment models for statistical machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 1086–1090, Saarbrücken, Germany, Aug. 2000.

[Och & Ney 00b] F.J. Och, H. Ney: Improved statistical alignment models. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 440–447, Hong Kong, Oct. 2000.

[Och & Ney 00c] F.J. Och, H. Ney: Statistical machine translation. In *Proc. of Workshop of the European Association for Machine Translation*, pp. 39–46, Ljubljana, Slovenia, May 2000.

[Och & Ney 01a] F.J. Och, H. Ney: Statistical multi-source translation. In *Proc. of Machine Translation Summit VIII*, pp. 253–258, Santiago de Compostela, Spain, Sept. 2001.

[Och & Ney 01b] F.J. Och, H. Ney: What can machine translation learn from speech recognition? In *Workshop: MT 2010 - Towards a Road Map for Machine Translation*, pp. 26–31, Santiago de Compostela, Spain, Sept. 2001.

[Och & Ney 02a] F.J. Och, H. Ney: Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, PA, July 2002.

[Och & Ney 02b] F.J. Och, H. Ney: A systematic comparison of various statistical alignment models. *Computational Linguistics*, Vol. 28, 2002. To appear.

[Och & Tillmann$^+$ 99] F.J. Och, C. Tillmann, H. Ney: Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, June 1999.

[Och & Ueffing$^+$ 01] F.J. Och, N. Ueffing, H. Ney: An efficient A* search algorithm for statistical machine translation. In *Data-Driven Machine Translation Workshop*, pp. 55–62, Toulouse, France, July 2001.

[Och & Weber 98] F.J. Och, H. Weber: Improving statistical natural language translation with categories and rules. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 985–989, Montreal, Canada, Aug. 1998.

[Ortmanns & Ney$^+$ 97] S. Ortmanns, H. Ney, X. Aubert: A word graph algorithm for large vocabulary continuous speech recognition. *Computer, Speech and Language*, Vol. 11, No. 1, pp. 43–72, Jan. 1997.

[Papineni & Roukos$^+$ 97] K.A. Papineni, S. Roukos, R.T. Ward: Feature-based language understanding. In *European Conf. on Speech Communication and Technology*, pp. 1435–1438, Rhodes, Greece, Sept. 1997.

[Papineni & Roukos$^+$ 98] K.A. Papineni, S. Roukos, R.T. Ward: Maximum likelihood and discriminative training of direct translation models. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 189–192, Seattle, WA, May 1998.

[Papineni & Roukos$^+$ 01] K.A. Papineni, S. Roukos, T. Ward, W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY, 10 pages, Sept. 2001.

[Price 90] P. Price: Evaluation of spoken language systems: The ATIS domain. In *Proc. of the Speech and Natural Language Workshop*, pp. 91–95, Hidden Valley, PA, June 1990.

[Rabiner & Juang 93] L. Rabiner, B.H. Juang: *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[Reithinger & Engel 00] N. Reithinger, R. Engel. Robust content extraction for translation and dialog processing. In [Wahlster 00], pp. 428–437.

[Resnik 99] P. Resnik: Mining the web for bilingual text. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 527–534, University of Maryland, College Park, MD, June 1999.

[Roark 01] B. Roark: Probabilistic top-down parsing and language modeling. *Computational Linguistics*, Vol. 27, No. 2, pp. 249–285, 2001.

[Sawaf & Schütz$^+$ 00] H. Sawaf, K. Schütz, H. Ney: On the use of grammar based language models for statistical machine translation. In *6th Int. Workshop on Parsing Technologies*, pp. 231–241, Trento, Italy, Feb. 2000.

[Simard & Foster$^+$ 92] M. Simard, G. Foster, P. Isabelle: Using cognates to align sentences in bilingual corpora. In *Fourth Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI-92)*, pp. 67–81, Montreal, Canada, June 1992.

[Smadja & McKeown$^+$ 96] F. Smadja, K.R. McKeown, V. Hatzivassiloglou: Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, Vol. 22, No. 1, pp. 1–38, March 1996.

[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, Yokohama, Japan, Sept. 1994.

[Tessiore & v. Hahn 00] L. Tessiore, W. v. Hahn. Functional validation of a machine interpretation system: Verbmobil. In [Wahlster 00], pp. 611–631.

[Tillmann 01] C. Tillmann. *Word Re-Ordering and Dynamic Programming based Search Algorithms for Statistical Machine Translation*. Ph.D. thesis, Computer Science Department, RWTH Aachen, Germany, May 2001.

[Tillmann & Ney 00] C. Tillmann, H. Ney: Word re-ordering and DP-based search in statistical machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Tillmann & Vogel$^+$ 97a] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga: A DP-based search using monotone alignments in statistical translation. In *Proc. 35th Annual Conf. of the Association for Computational Linguistics*, pp. 289–296, Madrid, Spain, July 1997.

[Tillmann & Vogel$^+$ 97b] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, H. Sawaf: Accelerated DP based search for statistical translation. In *European Conf. on Speech Communication and Technology*, pp. 2667–2670, Rhodes, Greece, Sept. 1997.

[Tillmann & Vogel+ 00] C. Tillmann, S. Vogel, H. Ney, H. Sawaf: Statistical translation of text and speech: First results with the RWTH system. *Machine Translation*, Vol. 15, No. 1/2, pp. 43–73, June 2000.

[Uszkoreit & Flickinger+ 00] H. Uszkoreit, D. Flickinger, W. Kasper, I.A. Sag. Deep linguistic analysis with HPSG. In [Wahlster 00], pp. 216–237.

[Vidal 97] E. Vidal: Finite-state speech-to-speech translation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 111–114, Munich, Germany, April 1997.

[Vidal et al. 00] E. Vidal et al. Final report of Esprit research project 30268 (EuTrans): Example-based language translation systems, deliverable D0.1c, 53 pages, Sept. 2000.

[Vilar & Vidal+ 96] J.M. Vilar, E. Vidal, J.C. Amengual: Learning extended finite state models for language translation. In *Proc. of the 12th European Conf. on Artificial Intelligence*, pp. 92–96, Budapest, Hungary, Aug. 1996.

[Vogel & Ney+ 96] S. Vogel, H. Ney, C. Tillmann: HMM-based word alignment in statistical translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pp. 836–841, Copenhagen, Denmark, Aug. 1996.

[Vogel & Nießen+ 00] S. Vogel, S. Nießen, H. Ney: Automatic extrapolation of human assessment of translation quality. In *2nd Int. Conf. on Language Resources and Evaluation (LREC 2000): Proc. of the Workshop on Evaluation of Machine Translation*, pp. 35–39, Athens, Greece, May 2000.

[Vogel & Och+ 00] S. Vogel, F.J. Och, C. Tillmann, S. Nießen, H. Sawaf, H. Ney. Statistical methods for machine translation. In [Wahlster 00], pp. 377–393.

[Wahlster 00] W. Wahlster, editor: *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, 2000.

[Wang 98] Y.Y. Wang. *Grammar Inference and Statistical Machine Translation*. Ph.D. thesis, School of Computer Science, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, 1998.

[Wang & Waibel 97] Y.Y. Wang, A. Waibel: Decoding algorithm in statistical translation. In *Proc. 35th Annual Conf. of the Association for Computational Linguistics*, pp. 366–372, Madrid, Spain, July 1997.

[Wang & Waibel 98] Y.Y. Wang, A. Waibel: Modeling with structures in statistical machine translation. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, Vol. 2, pp. 1357–1363, Montreal, Canada, June 1998.

[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.

[Wu 96] D. Wu: A polynomial-time algorithm for statistical machine translation. In *Proc. of the 34th Annual Conf. of the Association for Computational Linguistics (ACL '96)*, pp. 152–158, Santa Cruz, CA, June 1996.

[Wu & Wong 98] D. Wu, H. Wong: Machine translation with a stochastic grammatical channel. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics*, pp. 1408–1414, Montreal, Canada, Aug. 1998.

[Yamada & Knight 01] K. Yamada, K. Knight: A syntax-based statistical translation model. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 523–530, Toulouse, France, July 2001.

[Yarowsky & Ngai$^+$ 01] D. Yarowsky, G. Ngai, R. Wicentowski: Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Human Language Technology Conference*, pp. 109–116, San Diego, CA, March 2001.

[Yarowsky & Wicentowski 00] D. Yarowsky, R. Wicentowski: Minimally supervised morphological analysis by multimodal alignment. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 207–216, Hong Kong, Oct. 2000.

[Zens 02] R. Zens. Kontextabhängige Statistische Übersetzungsmodelle. Diploma thesis, Computer Science Department, RWTH Aachen, Aachen, Germany, June 2002.

Lebenslauf - Curriculum Vitae

Name:	Franz Josef Och
Adresse:	Trattach 5 91362 Pretzfeld
E-Mail:	och@cs.rwth-aachen.de
Geburtstag:	2. November 1971
Geburtsort:	Ebermannstadt
Staatsangehörigkeit:	Deutsch

Schulbildung	1978 – 1984	Grundschule Pretzfeld
	1984 – 1988	Realschule Ebermannstadt
	1988 – 1990	Fachoberschule Erlangen
Studium der Informatik	1990 – 1991	Fachhochschule Regensburg
	1991 – 1995	Universität Erlangen–Nürnberg
	1995 – 1996	Universität Bologna, Italien
	1996 – 1998	Universität Erlangen–Nürnberg
	März 1998	Diplom in Informatik mit Auszeichnung
Arbeitstätigkeiten	1998 – Sept. 2002	Wissenschaftlicher Angestellter am Lehrstuhl für Informatik VI an der RWTH Aachen
	Okt. 2002 –	Mitarbeiter am Information Science Institute der University of Southern California