Diplomarbeit im Fach Informatik
Rheinisch-Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik 6
Prof. Dr.-Ing. H. Ney

# Training Phrase Models for Statistical Machine Translation

24 April 2009

vorgelegt von:
Jörn Wübker
Matrikelnummer 243912

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. G. Lakemeyer, Ph.D.

Betreuer:
Dipl.-Inform. A. Mauser

# Erklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, 23. April 2009

Jörn Wübker

# Abstract

In this work we present and compare different methods of estimating phrase translation probabilities for statistical machine translation (SMT).

Given a source sentence, a phrase-based SMT system produces a translation by segmenting the sentence into phrases and translating those phrases separately. The phrase translation table, which contains the bilingual phrase pairs and the corresponding probabilities, is an essential part of a phrase-based SMT system. We examine different methods for training phrase translation probabilities. In contrast to state-of-the-art heuristics, the proposed training procedure is consistent with the translation decoder. By combination of the current state of the art with novel models we develop a method which performs equal or better than the baseline in all tested setups. Current state of the art is to use phrases heuristically extracted from word-aligned bilingual corpora. This makes translation quality strongly dependent on the quality of the underlying word alignment algorithms. We propose several novel phrase models and develop a procedure to train them. Different from previous similar approaches, we make use of the leaving-one-out method to counteract overfitting effects. The models we introduce consider both bilingual phrase translation probabilities and monolingual phrase prior probabilities. Further, different strategies of incorporating the information gained from training into the heuristic model are considered. In additional experiments we tested the impact of initialization on the translation results and also utilized the training process to re-estimate word alignments.

The performance of the different approaches is measured and compared on two standard data sets, a small Chinese-English corpus and a medium sized German-English corpus. We show that on the German-English corpus our proposed phrase models lead to improvements of translation quality over the baseline system which represents the current state of the art. A method is developed, which consistently performs better or equal to the baseline system in all setups we experimented on. This method is based on a phrase table interpolation.

# Kurzfassung

In dieser Arbeit werden verschiedene Methoden zur Schätzung der Übersetzungswahrscheinlichkeiten von Phrasen in statistischer maschineller Übersetzung (SMT) präsentiert und verglichen.

Ein phrasenbasiertes SMT-System zerlegt einen Quellsatz in kontinuierliche Segmente und übersetzt die Segmente (Phrasen) einzeln. Die Phrasentabelle, welche die bilingualen Phrasenpaare und die dazugehörigen Wahrscheinlichkeiten enthält, ist ein wichtiger Bestandteil eines solchen Systems. Für das Training der Übersetzungswahrscheinlichkeiten werden verschiedene Techniken untersucht, die im Gegensatz zu den Heuristiken konventioneller Systeme konsistent mit dem Suchverfahren für die freie Übersetzung sind. Durch Kombination eines konventionellen Systems mit neuartigen Phrasenmodellen wird eine Methode entwickelt, die in allen durchgeführten Versuchen vergleichbare oder bessere Übersetzungen liefert als die Baseline. Konventionelle Systeme basieren auf Phrasen, die heuristisch aus Wort-alignierten bilingualen Korpora extrahiert werden. Die Qualität der Übersetzungen hängt deswegen stark von der Qualität der zugrunde liegenden Wort-Alignments ab. In dieser Arbeit werden mehrere neuartige Phrasenmodelle eingeführt und eine Technik für deren Training entwickelt. Anders als in bisherigen Ansätzen wird von der Leaving-One-Out-Methode Gebrauch gemacht, um Überanpassung zu vermeiden. Sowohl für bilinguale Übersetzungswahrscheinlichkeiten als auch für monolinguale A-Priori-Verteilungen werden Modelle eingeführt. Darüber hinaus werden verschiedene Strategien untersucht, die im Training gewonnenen Informationen in das heuristische Modell einzubinden. In weiteren Experimenten wurde der Einfluss der Initialisierung auf die Übersetzungen geprüft, sowie mithilfe des Trainingsverfahrens neue Wort-Alignments generiert.

Die verschiedenen Ansätze werden auf zwei Datensätzen verglichen, einem kleinen Chinesisch-Englischen Korpus und einem Deutsch-Englischen Korpus mittlerer Grösse. Auf dem Deutsch-Englischen Datensatz führen Die eingeführten Phrasenmodelle zu einer Verbesserung der Übersetzungsqualität gegenüber der Baseline. Weiterhin wird eine Methode entwickelt, die in allen Versuchen vergleichbare oder bessere Übersetzungen produziert als die Baseline und auf der Interpolation von Phrasentabellen basiert.

# Acknowledgements

I would like to thank all the people who have supported me in the production of this work.

In particular, I express my gratitude towards Prof. Dr. Ing. H. Ney for giving me the opportunity to write my diploma thesis at the Chair for Computer Science 6 at the RWTH Aachen University.

Special thanks go to my supervisor Arne Mauser for his unrelenting support and for always being available for questions, discussions, advice and constructive criticism.

I would also like to thank Prof. G. Lakemeyer for co-evaluating my thesis.

Further, I would like to thank Cristian Wente, Christian Kohlschein, Christian Buck, Stephan Jonas and Pavel Golik for proofreading this work.

My thanks also go to all student workers at the Chair for Computer Science 6 for the friendly atmosphere and the regular much needed cheering up.

I thank the administrators of the RBI cluster for their immediate help in case of technical problems and for keeping the computers up and running reliably for the great number of time consuming and computationally demanding experiments I ran to evaluate my work.

Finally, and most of all, I would like to thank my parents for their infinite compassion and support throughout the past years, making possible not only this work and the preceding university studies, but also the many non-academic activities I have been following during this time. For this and your never ceasing, judgement-free encouragement I am deeply grateful.

# Contents

# 1 Introduction

## 1.1 Statistical Machine Translation

Machine Translation (MT) is the task of automatically producing a translation in one natural language from a given text written in another natural language. In spite of the research devoted to this topic in the past decades it is still considered to be an unsolved problem. But even though with current technology it is not possible to build a system that produces high quality translations regardless of domain, we can make good use of automation in a number of translation related tasks.

For information retrieval for example, a precise translation of a document may not be necessary. To get a rough idea of the contents of a text in a foreign language, a sketchy and potentially flawed translation can be sufficient for a human reader.

Another possible application is to aid human translators in their work. Providing them with one or several suggested preliminary translations can speed up their efforts.

Some translation tasks require understandable translations, but only on a limited domain. This restricts the vocabulary and thus reduces the complexity of the process. Exchanging travel information and making appointments are two examples of such tasks.

We distinguish between two conceptually different approaches to machine translation.

- **Rule-based Approach**:
  For rule-based systems human experts devise a set of fixed rules which are used to transform a text into an intermediary representation from which the translation is produced. To condense the semantics of a sentence, these rules can exploit syntactical structures and morphological dependencies. However, in order to capture the complex interdependencies within a natural language, a large number of rules is required. Producing this set of rules is a time consuming process and ensuring consistency gets more and more difficult as the number of rules increases.

- **Data-driven Approach:**
  In the data-driven approach we use bilingual and monolingual corpora as a main knowledge source. Here, MT is treated as a statistical decision problem. Given a source language sentence, the system has to decide for the best translation in the target language. We model the underlying probability distributions, tune the models on the data from our knowledge source and make use of statistical decision theory to address the problem.

In this work the statistical approach is taken. Statistical decision theory is a well understood field, providing us with sound ways of combining knowledge sources to construct a global decision criterion and growing in reliability as more and bigger training corpora become available.

One important method for statistical machine translation (SMT) provided by decision theory is the Bayes decision rule. Given a source language sentence $f_1^J = f_1 \dots f_J$ which is to be translated into a target language sentence $e_1^I = e_1 \dots e_I$, we choose the hypothesis $\hat{e}_1^{\hat{I}}$ which maximizes the posterior probability $Pr(e_1^I|f_1^J)$ [Och & Ney 02]:

$$\hat{e}_1^{\hat{I}} = \operatorname*{arg\,max}_{I,e_1^I} \left\{ Pr(e_1^I|f_1^J) \right\} \tag{1.1}$$

With the decision rule specified, three problems have to be addressed in SMT [Ney 01]:

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

In this work we will concentrate on a specific aspect of the training problem, namely the training of the phrase translation table which will be introduced in Section 2.1.

## 1.2 Related work

Our research focuses on two central points: Generative phrase models and the use of forced alignment to train them. In this section we will review some previous work on these topics.

[DeNero & Gillick$^+$ 06] give a detailed analysis of the difficulties with training a generative phrase model. They introduce a model similar to one we propose in Section 4 and train it with the Expectation-Maximization (EM) algorithm (cf. [Dempster & Laird$^+$ 77]). Their results show that it can not reach a performance competitive to extracting a phrase table from word alignment by simple surface heuristics [Zens & Och$^+$ 02]. Several reasons are revealed in [DeNero & Gillick$^+$ 06]. When given a correct phrase segmentation and alignment for a bilingual sentence pair we can not assume that competing segmentations are wrong. This stands in contrast to word-based translation models, where we can assume only one word alignment to be correct. As a result, different segmentations are recruited for different examples during training. That in turn leads to overfitting which shows in overly determinized estimates of the phrase translation probabilities. Furthermore they found that the trained phrase table shows a highly peaked distribution in opposition to the flat distribution given by the surface heuristic, which leads to undesired effects at decoding time. Our work differs from [DeNero & Gillick$^+$ 06] in a number of ways, addressing those problems. To limit the effects of overfitting, we apply the leaving-one-out method in training. In addition to that we do not restrict the training to phrases consistent with the word alignment, as was done in [DeNero & Gillick$^+$ 06]. Thus we allow recovery from flawed word alignments. Thirdly, our models address the problem of competition between equally correct phrase segmentations by integrating into our estimates a number of different segmentations for each training sentence.

In [Liang & Buchard-Côté$^+$ 06] a discriminative translation system is described. For training of the parameters for the discriminative features they propose a strategy they call *bold updating*. It is similar to our training procedure, the forced alignment method which we describe in Section 3.1.

Forced alignment can also be utilized to train a phrase segmentation model, as is shown in [Shen & Delaney$^+$ 08]. They report small but consistent improvements by incorporating this segmentation model, which works as an additional prior probability on the monolingual target phrase. We will consider this method to refine our model in Section 4.3.3.

## 1.3 Outline of this work

This work is structured as follows. In Chapter 2 we will review the principles of phrase-based statistical machine translation, give a detailed account of the RWTH system used for experimenting and describe the state-of-the-art method for estimating phrase translation probabilities, which will serve as our baseline. Chapter 3 concentrates on the topic of training our phrase models, and the forced alignment method which is applied for this purpose. The novel feature functions and generative phrase models we propose are specified in Chapter 4. We compare the performance of our novel models experimentally on two standard data sets in Chapter 5. After reviewing the applied evaluation metrics, we will give an account of the data sets and the experimental setup and discuss the results. In Chapter 6 we give a summary and an outlook on possible future work.

# 2 Phrase-based SMT

## 2.1 Phrases

The first approaches to SMT were single-word based (SWB). In SWB machine translation each word is translated by itself and contextual information is provided by the language model only.

To better capture local context, recent SMT systems make use of a phrase translation table and consider whole phrases to be translated as a unit. Here, a phrase is understood as a contiguous sequence of words. When given a source sentence the system will first segment it into phrases, look up each of the phrases in the phrase translation table and produce the target sentence by piecing together the translation phrases. Figure 2.1 shows an example of a phrase-based translation [Zens & Och$^+$ 02].

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

$$k \rightarrow s_k := (i_k; b_k, j_k), \text{ for } k = 1, \dots, K \tag{2.1}$$

SOURCE: abends würde ich gerne entspannen und vielleicht in die Sauna gehen .

| **source segmentation** | **translation** |
|---|---|
| abends | in the evening |
| würde ich gerne entspannen | I would like to relax |
| und | and |
| vielleicht in die Sauna gehen | maybe go to the sauna |
| . | . |

TARGET: in the evening I would like to relax and maybe go to the sauna .

**Figure 2.1.** Example for phrase-based translation.

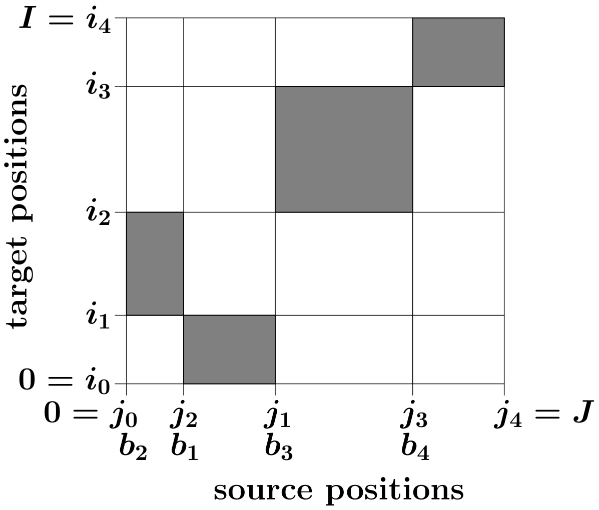

**Figure 2.2.** Illustration of phrase segmentation.

Here $i_k$ denotes the last position of the $k$th target phrase and the pair $(b_k, j_k)$ denotes the start and end positions of the source phrase which is aligned to the $k$th target phrase. We set $i_0 := 0$ and $j_0 := 0$. In our definition all words in source and target sentence have to be covered by exactly one phrase.

Given the sentence pair $(f_1^J, e_1^I)$ and the segmentation $s_1^K$ we define the bilingual phrase pairs as:

$$\tilde{e}_k := e_{i_{k-1}+1} \dots e_{i_k} \tag{2.2}$$

$$\tilde{f}_k := f_{b_k} \dots f_{j_k} \tag{2.3}$$

Figure 2.2 illustrates this notion of phrase segmentation. Note that our definition of the segmentation $s_1^K$ explicitly contains the information on phrase-level reordering.

## 2.2 The RWTH System

### 2.2.1 Source-channel approach

When given a source sentence $f_1^J$ we apply the Bayes decision rule and choose the target sentence $e_1^I$ which maximizes the posterior probability as specified in Equation (1.1). In the source-channel approach to SMT the posterior probability is decomposed into two knowledge sources [Brown & Cocke$^+$ 90]:

$$\hat{e}_1^{\hat{I}} = \operatorname*{arg\,max}_{I,e_1^I} \left\{ Pr(e_1^I|f_1^J) \right\} \tag{2.4}$$

$$= \operatorname*{arg\,max}_{I,e_1^I} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} \tag{2.5}$$

In this decomposition the target language model $Pr(f_1^J)$ and the translation model $Pr(f_1^J|e_1^I)$ are considered separately. The target language model caters for the target language sentence to be well formed, while the translation model describes the dependencies between source language sentence and target language sentence.

### 2.2.2 Log-linear modeling

The log-linear model is a generalization of the source-channel approach. We can incorporate an arbitrary number $M$ of models $h_m(\cdot,\cdot,\cdot), m = 1 \dots M$, and use scaling factors $\lambda_m$ to assign them different weights:

$$Pr(e_1^I|f_1^J) = \sum_{K,s_1^K} \frac{\exp\left( \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right)}{\sum_{I',{e'}_1^{I'},K',{s'}_1^{K'}} \exp\left( \sum_{m=1}^{M} \lambda_m h_m({e'}_1^{I'}, {s'}_1^{K'}, f_1^J) \right)} \tag{2.6}$$

$$\approx \max_{K,s_1^K} \frac{\exp\left( \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right)}{\sum_{I',{e'}_1^{I'},K',{s'}_1^{K'}} \exp\left( \sum_{m=1}^{M} \lambda_m h_m({e'}_1^{I'}, {s'}_1^{K'}, f_1^J) \right)} \tag{2.7}$$

In practice, instead of carrying out the sum over all segmentations $s_1^K$ we apply the maximum approximation. The denominator can be omitted during search, as it is a normalization factor that depends only on the source language sentence $f_1^J$. As a decision rule we obtain:

$$\hat{e}_1^{\hat{I}} = \operatorname*{arg\,max}_{I, e_1^I, K, s_1^K} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \tag{2.8}$$

The model scaling factors $\lambda_1^M$ are trained by minimum error rate training (MERT) as described in Section 2.2.5.

### 2.2.3 Models

#### Phrase translation model

The phrase translation model is the main focus of this work. It assigns an estimated probability $p(\tilde{f}|\tilde{e})$ to each pair of target and source phrase $(\tilde{f}, \tilde{e})$. We will examine different ways of estimating these probabilities in Sections 2.3 and 4.

The corresponding feature function $h_{Phr}(\cdot, \cdot, \cdot)$ is composed of the translation probabilities of the phrases given by the segmentation $s_1^K$ [Mauser & Zens$^+$ 06]:

$$h_{Phr}(e_1^I, s_1^K, f_1^J) = log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \tag{2.9}$$

This model is used in both translation directions $p(\tilde{f}|\tilde{e})$ and $p(\tilde{e}|\tilde{f})$ to achieve symmetry. The inverse model is:

$$h_{iPhr}(e_1^I, s_1^K, f_1^J) = log \prod_{k=1}^{K} p(\tilde{e}_k|\tilde{f}_k) \tag{2.10}$$

#### Word-based lexicon model

In the word-based model, the score for a phrase pair $(\tilde{f}, \tilde{e})$ is computed similar to the IBM model 1 [Brown & Pietra$^+$ 93], but the sum is carried out within the phrase pair only instead of over the whole target sentence:

$$h_{Lex}(e_1^I, s_1^K, f_1^J) = log \prod_{k=1}^{K} \prod_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \tag{2.11}$$

To estimate the word translation probabilities $p(f|e)$ we use relative frequencies from the word-aligned training corpus. This model is also used in both translation directions $p(f|e)$ and $p(e|f)$.

### Word and phrase penalty model

In order to be able to adjust average phrase and sentence length, we introduce two simple heuristics, the word penalty and the phrase penalty:

$$h_{WP}(e_1^I, s_1^K, f_1^J) = I \tag{2.12}$$

$$h_{PP}(e_1^I, s_1^K, f_1^J) = K \tag{2.13}$$

### Target language model

A standard $n$-gram language model is used, trained by the SRI language modeling toolkit [Stolcke 02]. This is the resulting feature function:

$$h_{LM}(e_1^I, s_1^K, f_1^J) = log \prod_{i=1}^{I} p(e_i|e_{i-n+1}^{i-1}) \tag{2.14}$$

As a smoothing technique the modified Kneser-Ney discounting with interpolation is applied [Kneser & Ney 95]. Depending on the size of the data set we use either a 4-gram or a 6-gram language model for our experiments, which in our experience provide a good compromise between translation quality and computational complexity.

### Reordering model

The reordering model assigns costs based on the distance from the end position of a phrase to the start position of the next phrase. There is an upper limit $D$ on the jump width:

$$h_{RM}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} q_{Dist}(b_k, j_{k-1}) \tag{2.15}$$

with

$$q_{Dist}(j, j') := \begin{cases} |j - j' + 1| & \text{if } |j - j' + 1| < D \\ \infty & \text{else} \end{cases} \tag{2.16}$$

We define $b_{K+1} := J + 1$, so the sum includes a jump from the last position of the final phrase to the sentence end.

### 2.2.4 Search

In search, our goal is to find the maximizing argument of the Bayes decision rule (cf. Equation (2.8)).

We have to decide on [Zens 08]:

- the number $K$ of phrases
- the segmentation of the source sentence into phrases
- the permutation of the phrases
- the phrase translation $\tilde{e}$ for each source phrase $\tilde{f}$

To find the best hypothesis, we make use of dynamic programming [Bellman 57]. As enumeration of all target language sentences is infeasible, approximations have to be made which are realized by the beam search technique [Jelinek 97].

We can interpret the search as a sequence of decisions $(\tilde{e}_k, b_k, j_k)$, $k = 1 \ldots K$. The hypotheses are generated step by step by choosing a source phrase $\tilde{f}_k$ with start position $b_k$ and end position $j_k$ and its translation $\tilde{e}_k$. To ensure that there are no gaps or overlaps in the produced target sentence, we keep track of the source positions we have already visited in the coverage set $C \subseteq \{1, \ldots, J\}$.

The search space can be represented as a graph, where the states are labeled with coverage sets $C$ and the arcs are labeled with the decisions $(\tilde{e}_k, b_k, j_k)$. The hypothesis translations are paths through the graph, starting in the initial state $C = \emptyset$ and terminating in the goal state $C = \{1, \ldots, K\}$.

In our log-linear framework, we can compute the score of the decision sequence by summing the scores of the individual decisions. For the phrase model, the word-based model and the word and phrase penalties, these scores are solely dependent

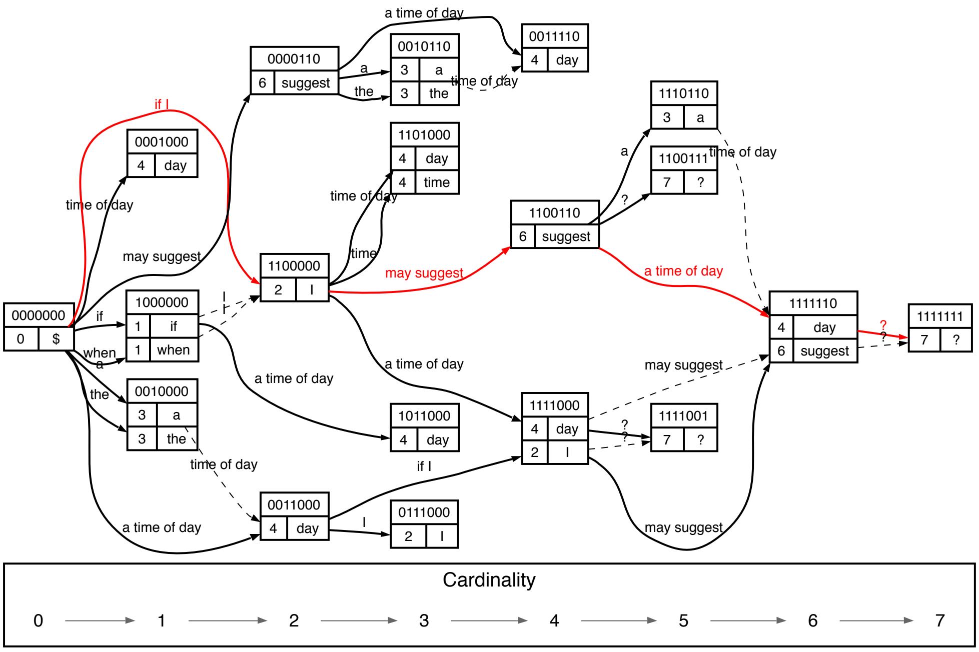

**Figure 2.3.** Illustration of the search. German input sentence: 'Wenn ich eine Uhrzeit vorschlagen darf?'. English translation: 'If I may suggest a time of day?' In each node, we store the coverage (as a bitvector), the end position of the current phrase and the language model history (here: bigram). Dashed edges are recombined. The best path is marked in red. Scores are omitted. Taken from [Zens 08].

on the chosen phrase pair $(\tilde{f}_k, \tilde{e}_k)$. For the language model and reordering model score we require some information on the decisions taken previously. Therefore we introduce additional labeling for the states, namely the end position of the previous source phrase and the language model history. If we use an $n$-gram language model, the language model history is defined as the last $(n-1)$ words of the target sentence generated up to the current state. Thus, each state is identified by a triple $(C, \tilde{e}, j)$, where $C$ denotes the coverage set, $\tilde{e}$ the language model history and $j$ the end position of the previous source phrase. We call the computation of the successor states of a given state $(C, \tilde{e}, j)$ hypothesis expansion. An example for this search graph is shown in Figure 2.3.

The search problem is equivalent to finding the optimum path within the described search graph. As each state is assigned a score which can be computed from its predecessor states, this allows us to apply dynamic programming to find the solution. The size of the search graph, however, is exponential in the source sentence length and it has been shown in [Knight 99] that the search problem is NP-hard.

Beam search is a feasible method to find a good approximation for the solution. The idea is to expand only the most promising hypotheses at each point during search. The process of discarding the other hypotheses is called pruning. This requires us to be able to estimate the rest costs for a given hypothesis, for which purpose we use the heuristics described in [Zens 08], pp. 57-59.

The search is being carried out synchronous to the cardinality of the coverage set $C$. This means we first produce all hypotheses which translate one source position, then the hypotheses which translate two source positions etc. At each stage we apply pruning on several levels.

We distinguish two kinds of hypotheses:

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

Figure 2.4 illustrates the source cardinality synchronous search strategy.

Two pruning variants are used: histogram pruning [Steinbiss & Tran$^+$ 94] and threshold pruning. In histogram pruning, we limit the total number of hypotheses, keeping only the $N$ best ones. Threshold pruning will keep only those hypotheses scoring close enough to the one with the highest score, according to some threshold parameter. Let $Q_{max}$ be the maximum score of a hypothesis on a given level and $\tau$ the threshold parameter. Then we will keep a hypothesis with score $Q$ iff:

$$Q + \tau \geq Q_{max} \tag{2.17}$$

Both $Q$ and $Q_{max}$ include a rest cost estimate if applicable. We apply four different pruning strategies:

1. **Observation pruning.** Before the actual search starts, we limit the number of translation options per source phrase. The scores used for pruning include a within-phrase estimate of the language model score, which restricts the language model history to the length of the phrase. This is the only pruning strategy where no rest cost estimate is included in the scores.

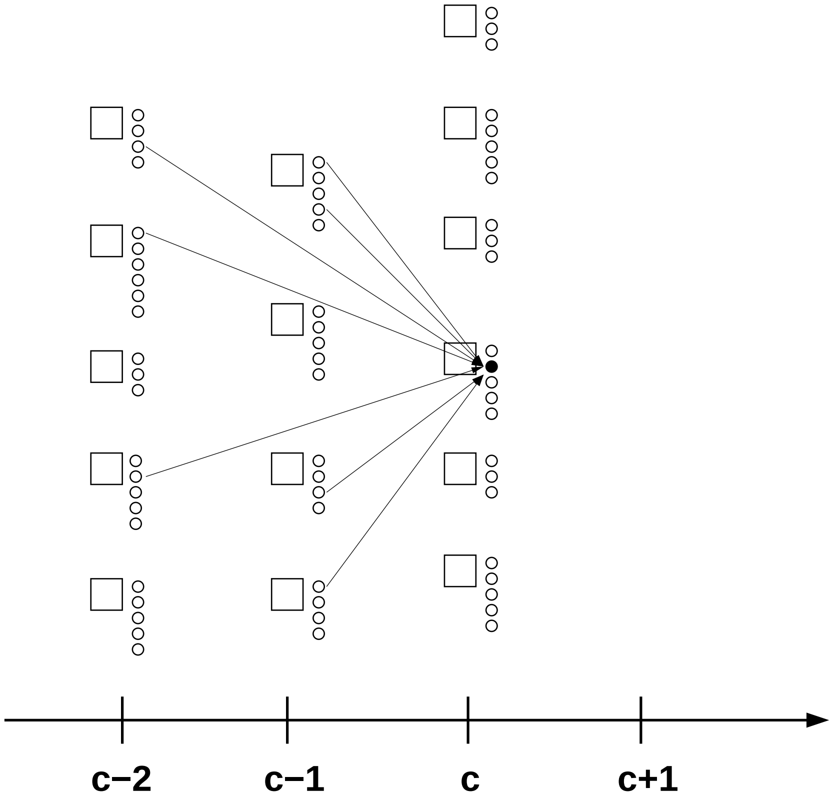

**Figure 2.4.** Illustration of the search. For each cardinality, we have a list of coverage hypotheses (boxes). For each coverage hypothesis, we have a list of lexical hypotheses (circles). A hypothesis with cardinality $c$ can be generated by expanding a hypothesis of cardinality $c - 1$ with a one-word phrase, by expanding a hypothesis of cardinality $c - 2$ with a two-word phrase etc. Taken from [Zens 08].

2. **Lexical pruning per coverage.** At this level, we consider all lexical hypotheses with the same coverage $C$. They may differ in their language model history $\tilde{e}$ or the end position $j$ of the previous phrase.
3. **Lexical pruning per cardinality.** Here, all lexical hypotheses with the same cardinality are taken into account, which therefore may differ in their coverage $C$.
4. **Coverage pruning per cardinality.** In this case, all coverage hypotheses with the same cardinality are considered. The score of a coverage hypothesis $C$ is the maximum score of any lexical hypothesis with coverage $C$.

To further reduce the search space, we adapted the word-based IBM reordering

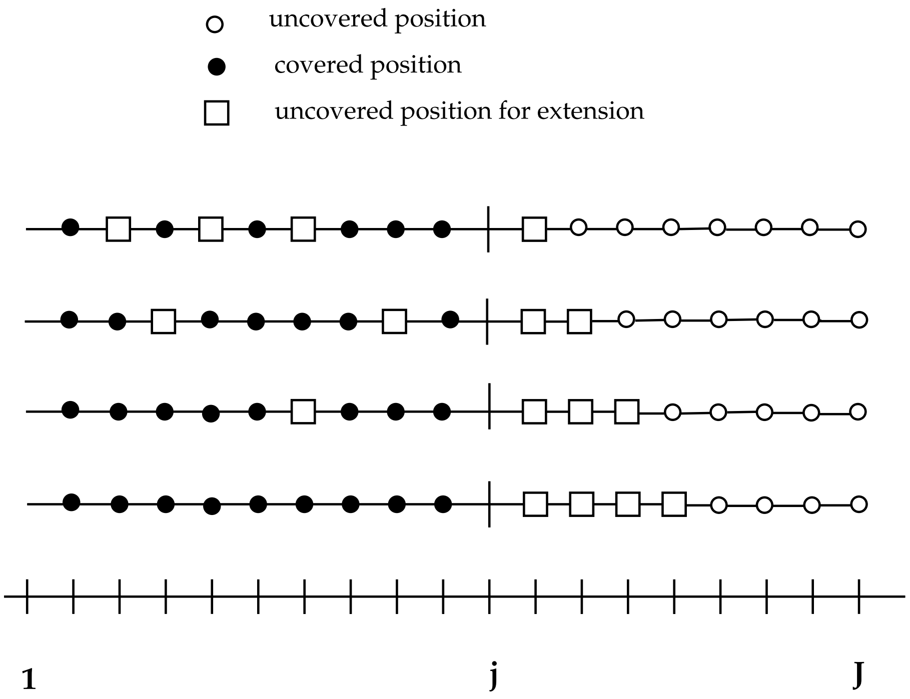

**Figure 2.5.** Illustration of the IBM reordering constraints. Taken from [Tillmann & Ney 00].

constraints [Berger & Brown+ 96] for our phrase-based framework. We will start by describing the original IBM constraints.

In the beginning each position in the source sentence is uncovered. The source positions are processed from left to right and it is allowed to skip a position and return to it at a later point. The next position always has to be one of the $k$ first uncovered positions in the sentence so that there are never more than $k-1$ skipped positions. Figure 2.5 illustrates this. The x-axis represents the source sentence positions, uncovered positions are marked with unfilled circles and covered positions with filled circles. Candidates for the next extension are shown as squares.

For the phrase-based framework we adapt the IBM constraints. It is permitted to skip $k-1$ blocks rather than single positions. This means we allow for up to $k-1$ gaps in the coverage. For a coverage $C$ we have to check the following condition during search:

$$\big|\{j > 1 | j \in C \wedge j-1 \notin C\}\big| < k \tag{2.18}$$

Note that there is no constraint on the number of phrases for filling the gaps. A detailed description of the complete search procedure, including all dynamic programming recursion equations, can be found in [Zens 08], pp. 47-80.

### 2.2.5 Minimum error rate training (MERT)

The goal of MERT is to find scaling factors $\lambda_1^M$, such that the system will produce good translations on a bilingual corpus with respect to some evaluation measure. In our experiments, we chose BLEU score [Papineni & Roukos$^+$ 02] and the development set as the bilingual corpus. We will go into more detail on the subject of evaluation metrics in Section 5.1.

We are given a bilingual data set $(\mathcal{F}, \mathcal{R})$, consisting of a sequence of source sentences $\mathcal{F}$ and a sequence of corresponding reference sentences $\mathcal{R}$, and a function $E(\cdot, \cdot)$ serving as an error measure. The error of a hypothesis translation $\mathcal{E}$ for the source sentences $\mathcal{F}$ is given by $E(\mathcal{R}, \mathcal{E})$.

Thus we want to optimize the following criterion [Och 03]:

$$\hat{\lambda}_1^M = \operatorname*{arg\,min}_{\lambda_1^M} \left\{ E(\mathcal{R}, \hat{\mathcal{E}}(\mathcal{F}, \lambda_1^M)) \right\} \tag{2.19}$$

where $\hat{\mathcal{E}}(\mathcal{F}, \lambda_1^M)$ is the hypothesis translation for $\mathcal{F}$ produced by the system with the parameters $\lambda_1^M$.

The algorithm we used for this optimization is the downhill simplex method proposed in [Nelder & Mead 65]. It guarantees local convergence, which is shown in [Press & Teukolsky$^+$ 02].

In our experiments, we restricted the translation system to monotonic phrasal alignment for the optimization. The scaling factor for the reordering model was therefore not included in MERT, but adjusted by hand afterwards.

## 2.3 Phrase extraction from word alignment

### 2.3.1 Word alignment

Current state of the art to estimate the phrase translation probabilities $p(\tilde{f}|\tilde{e})$ described in Section 2.2.3 is to extract the phrases from a word alignment.

The word alignment describes, which words within a bilingual sentence pair correspond to each other. Formally, given a sentence pair $(e_1^I, f_1^J)$, we define a word alignment $A$ as a relation over the word indices:

$$A \subseteq I \times J \tag{2.20}$$

We say $e_i$ is aligned to $f_j$ iff $(i, j) \in A$. Figure 2.6 shows an example for a word alignment.

As a starting point for the phrase extraction method we are going to describe next, we need a word alignment on the bilingual training corpus. We produce this word alignment by training statistical alignment models using GIZA++ [Och & Ney 03] for both translation directions and computing the Viterbi word alignments. Then we merge the two alignments by applying the refined symmetrization method proposed in [Och & Ney 03].

### 2.3.2 Phrase extraction

The set of bilingual phrases $\mathcal{BP}$ the system considers to be translations of each other are learned from the given word alignment $A$. For this we apply the alignment template criterion from [Och & Ney 04]. Informally, we consider a phrase pair to be consistent with the word alignment, if both phrases are contiguous and none of the words within the phrase pair are aligned to words outside the phrase pair. A phrase pair is extracted if it is consistent with the word alignment. Formally, we get the following criterion for a given sentence pair $(f_1^J, e_1^I)$ with alignment $A$:

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, A) = \Big\{ (f_{j_1}^{j_2}, e_{i_1}^{i_2}) : \forall (j, i) \in A : j_1 \leq j \leq j_2 \leftrightarrow i_1 \leq i \leq i_2 \\ \wedge \exists (j, i) \in A : j_1 \leq j \leq j_2 \wedge i_1 \leq i \leq i_2 \Big\} \end{aligned} \tag{2.21}$$

Figure 2.7 shows the complete list of phrases that can be extracted from the word alignment given in Figure 2.6.

In practice, the length of the extracted phrases is restricted to constrain the size of the phrase table. We define the length of a phrase $\tilde{f}$ to be the number of its words and denote it with $|\tilde{f}|$. A phrase pair $(\tilde{f}, \tilde{e})$ is only stored in the phrase table if $|\tilde{f}| \leq f_{max}$ and $|\tilde{e}| \leq e_{max}$.

The parameters $f_{max}$ and $e_{max}$ are hand-adjusted to fit the language pair. Typical values are $f_{max} = 5$ and $e_{max} = 10$ for German-English and $f_{max} = 6$ and $e_{max} = 12$ for Chinese-English.

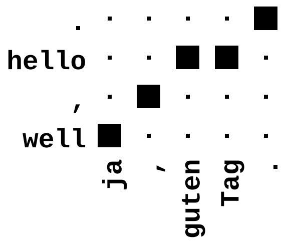

**Figure 2.6.** Word aligned sentence pair.

| source phrase | target phrase |
|---|---|
| ja | well |
| ja, | well, |
| ja, guten Tag | well, hello |
| ja, guten Tag. | well, hello. |
| , | , |
| , guten Tag | , hello |
| , guten Tag. | , hello. |
| guten Tag | hello |
| guten Tag. | hello. |
| . | . |

**Figure 2.7.** List of extracted phrases.

### 2.3.3 Heuristic phrase count model

The next step is to assign translation probabilities $p(\tilde{f},\tilde{e})$ to the extracted phrase pairs. In our baseline system, relative frequencies are used for the estimation. For a word-aligned training data set the phrase translation probabilities are given by:

$$p_H(\tilde{f}|\tilde{e}) = \frac{N_H(\tilde{f},\tilde{e})}{N(\tilde{e})} \qquad (2.22)$$

Here, $N_H(\tilde{f},\tilde{e})$ denotes the number of co-occurrences of the phrase pair $(\tilde{f},\tilde{e})$ in the bilingual training data that are consistent with the word alignment. If there are non-aligned words in the sentence, one occurrence of a target phrase $\tilde{e}$ can have $N > 1$ possible translations. In this case each of them contributes to $N(\tilde{f},\tilde{e})$ with $1/N$. Note that therefore the count $N(\tilde{f},\tilde{e})$ may differ from the corresponding count in the inverse direction $N(\tilde{e},\tilde{f})$. Figure 2.8 illustrates an example of these effects. There are three different target phrases $\tilde{e}$ that can be extracted as possible translations for the source phrase $\tilde{f}$ = *'eine Uhrzeit'*. In the inverse translation direction, each of them contributes to $N(\tilde{e},\tilde{f})$ with 1/3. In the original translation direction, however, each of them contributes to $N(\tilde{f},\tilde{e})$ with 1, as *'eine Uhrzeit'* is the only source phrase consistent with the word alignment for the target phrases in question.

The marginal count $N(\tilde{e}) = \sum_{e_1^I \in \mathcal{R}} |\{(i_1, i_2) : \tilde{e} = e_{i_1}^{i_2}\}|$ is the number of occurrences of the phrase $\tilde{e}$ on the target side $\mathcal{R}$ of the training corpus. Also note that $N(\tilde{e}) \geq \sum_{\tilde{f}} N(\tilde{f},\tilde{e})$, as there may be occurrences of the target phrase with no consistent source phrase, which contribute to $N(\tilde{e})$ but not to the joint count $N(\tilde{f},\tilde{e})$.

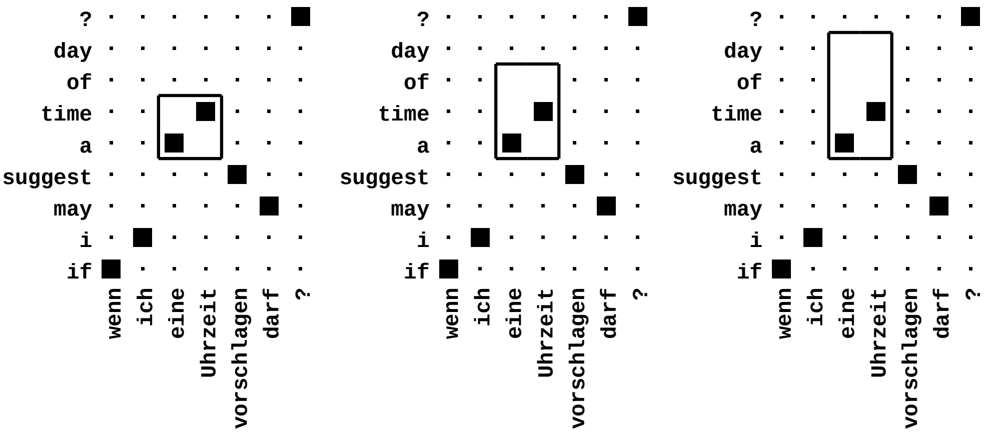

**Figure 2.8.** Illustration of the different target phrases $\tilde{e}$ that can be extracted as possible translations for the source phrase $\tilde{f}$ = *'eine Uhrzeit'*.

Due to this fact the heuristic phrase model does not yield an actual probability distribution over the set of source phrases $\tilde{f}$, as the summation to unity constraint is violated.

# 3 Training

In Section 2 we have described a statistical machine translation system which reflects the current state of the art. However, one of its main components, the phrase translation model (cf. Section 2.2.3), is based on a heuristic estimation of phrase counts from word alignment data. Our goal is to introduce a phrase translation model which is independent of this kind of previous determination. For that purpose we adapt the translation decoder to produce a phrase level alignment on the training data, from which we can compute real phrase counts rather than having to estimate them from lower-level information. Figure 3.1 illustrates the training procedure.

## 3.1 Forced alignment

To train new phrase models we apply a method we call forced alignment (FA), which is based on the translation decoder described in Section 2. We are given a bilingual training corpus. For each sentence pair $(f_1^J, e_1^I)$ our goal is to find a corresponding phrase segmentation $\hat{s}_1^{\hat{K}}$.

Two problems have to be solved: the segmentation problem and the phrase alignment problem. The phrase alignment problem corresponds to the word alignment problem which has been widely studied in the context of training word translation models [Brown & Pietra+ 93]. The difference is that we have two sequences of

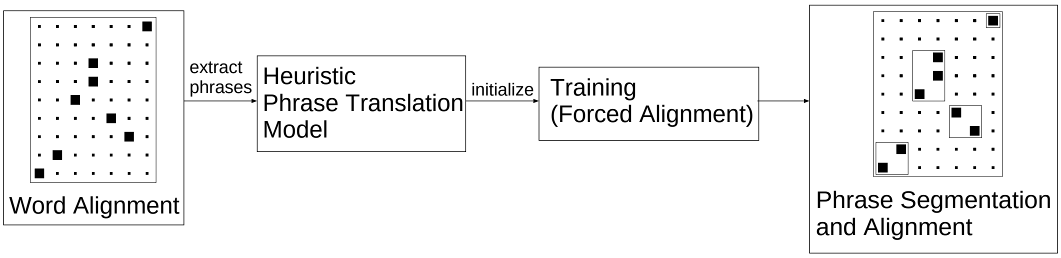
Word Alignment
extract phrases
Heuristic Phrase Translation Model
initialize
Training (Forced Alignment)
Phrase Segmentation and Alignment

**Figure 3.1.** Illustration of the training procedure.

SOURCE: 明早 去 巴黎 然后 再 转 瑞士 航空 公司 下午 的 飞机 怎么样 ?

REFERENCE:
how about leaving for paris tomorrow morning and transferring to the swissair afternoon flight ?

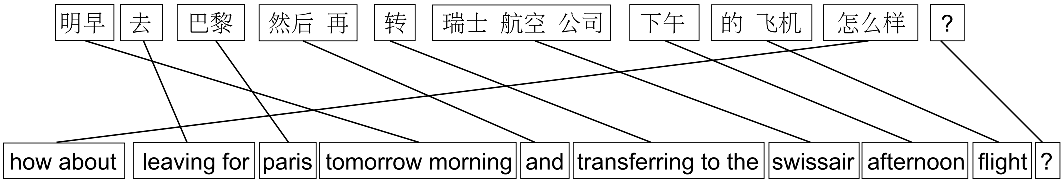

**Figure 3.2.** Example segmentation for a sentence from the IWSLT training data set.

phrases rather than words that need to be aligned to each other. The segmentation problem is specific to phrase-based translation. Source and target sentence have to be segmented into phrases. Figure 3.2 shows an example of a phrase segmentation and alignment for a sentence pair from the IWSLT Chinese-English data (cf. Chapter 5.2).

The implementation of forced alignment is straightforward. We use the same system as for translation, but constrain the search to the reference sentence $e_1^I$. Analogous to $Pr(e_1^I|f_1^J)$ in Equation 2.6 we model the probability distribution over the segmentations in the following way:

$$Pr(s_1^K|e_1^I, f_1^J) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J)\right)}{\sum_{K', {s'}_1^{K'}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, {s'}_1^{K'}, f_1^J)\right)} \tag{3.1}$$

The decision rule for the best phrase segmentation is analogous to Equation (2.8):

$$\hat{s}_1^{\hat{K}} = \operatorname*{arg\,max}_{K, s_1^K} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \tag{3.2}$$

To distinguish it from the true distribution $Pr(s_1^K|e_1^I, f_1^J)$ we will denote the model distribution estimated by forced alignment with $p_{FA}(s_1^K|e_1^I, f_1^J)$.

**Table 3.1.** Average number of candidate phrases that match the given source sentence $f_1^J$ in translation compared to the number of phrases that match the given sentence pair $(f_1^J, e_1^I)$ in FA for the IWSLT training data set (cf. Section 5.2).

| | # candidate phrases |
|---|---:|
| translation | 6473 |
| forced alignment | 133 |

Although search remains a hard problem, if implemented in the right way the restriction on only a single target sentence can reduce the search space and thus computational load and memory usage considerably in comparison with translation. In the phrase matching phase for unconstrained search the decoder filters all candidate phrase pairs $(\tilde{f}, \tilde{e})$ from the phrase table, for which the source phrase $\tilde{f}$ can be found in the source sentence $f_1^J$. In FA the phrase matching can be applied to both source and target side. Table 3.1 compares the average number of candidate phrases per sentence for unconstrained search and forced alignment for the IWSLT data set.

In addition to that, we can ignore the target language model $h_{LM}(\cdot, \cdot, \cdot)$ as it is constant if the target sentence remains fixed. The scaling factors $\lambda_1^M$ for FA are the same as used for translation.

## 3.2 Leaving-one-out

### 3.2.1 Motivation

The training data set is used for both the initialization of the translation model $p(\tilde{f}|\tilde{e})$ as for the phrase model training. While in this way we can make full use of the available data and avoid unknown words during training, it has the drawback that it can lead to overfitting. If the initialization is done with the heuristic described in Section 2.3, all phrases extracted from a specific sentence pair $(f_1^J, e_1^I)$ can be used for the segmentation of $(f_1^J, e_1^I)$. This includes longer phrases, which only match a few sentences in the data. Therefore those long phrases are trained to fit the few corresponding sentence pairs, strongly overestimating their translation probabilities and failing to generalize. The length of the used phrases is an indicator of this kind of overfitting, as the number of matching training sentences decreases with increasing phrase length. We can see an example in Figure 3.3 where the sentence is segmented into only two phrases.

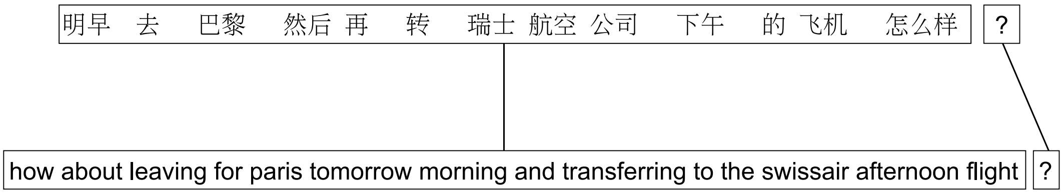
明早 去 巴黎 然后 再 转 瑞士 航空 公司 下午 的 飞机 怎么样
?
how about leaving for paris tomorrow morning and transferring to the swissair afternoon flight
?

**Figure 3.3.** Top scoring segmentation in FA without leaving-one-out.

A possible way to reduce the described overfitting effects would be to simply restrict the phrase length. However, [DeNero & Gillick+ 06] have presented results that indicate this might not be enough. They experienced the same kind of overfitting with short phrases due to the fact that the same word sequence can be segmented in different ways, leading to specific segmentations being learned for specific training sentence pairs. Therefore we propose a different approach to deal with this problem. We will first take a closer look at the effects leading to overfitting in training on the example of initialization with the heuristic described in Section 2.3.

For a sentence pair $(f_1^J, e_1^I)$ with word alignment $A$ the segmentation $s_1^K$ which segments the whole sentence as a single phrase pair $(\tilde{f}, \tilde{e}) = (f_1^J, e_1^I)$ is always consistent with $A$ and will therefore be extracted by the heuristic. If the same target sentence $e_1^I$ does not appear in the data set a second time, the relevant counts are

$$N_H(\tilde{f}, \tilde{e}) = N(\tilde{e}) = 1, \tag{3.3}$$

where $N_H(\tilde{f}, \tilde{e})$ and $N(\tilde{e})$ are the heuristic phrase count and the monolingual phrase count we defined in Section 2.3. Therefore, the heuristic estimates the translation probability to be

$$p_H(\tilde{f}|\tilde{e}) = \frac{N_H(\tilde{f}, \tilde{e})}{N(\tilde{e})} = 1, \tag{3.4}$$

where $p_H(\tilde{f}|\tilde{e})$ is the heuristic estimation of the phrase translation probability as defined in Equation (2.22). As a result, in training there is a strong bias towards the segmentation as a single phrase, and more generally towards segmentations with long phrases, leaving us with a skewed model distribution $p_{FA}(s_1^K|e_1^I, f_1^J)$. To overcome this problem, we apply the leaving-one-out method (l1o).

**Table 3.2.** Avg. target phrase lengths in FA with and without standard leaving-one-out on the IWSLT training data.

| | avg. phrase length |
|---|---:|
| without l1o | 5.2 |
| with l1o | 2.0 |

The idea of leaving-one-out is the following. When we compute the forced alignment, the current sentence pair $(f_1^J, e_1^I)$ is removed from the training data from which the phrase translation probabilities $p(\tilde{f}|\tilde{e})$ are estimated. This means we have to compute a different phrase table for each sentence pair in the training data. We do this by appropriately reducing the overall phrase counts of the phrases that can be extracted from the current sentence pair $(f_1^J, e_1^I)$. The heuristic phrase count model $p_H(\tilde{f}, \tilde{e})$ is then re-estimated from those counts. Table 3.2 shows the average phrase length used in FA on the IWSLT training data with and without leaving-one-out. We can clearly see the reduction of phrase lengths. Our results in Section 5.3.2 show that leaving-one-out is superior to a simple restriction of phrase length.

### 3.2.2 Implementation

Before we start the training, the heuristic monolingual and bilingual phrase counts $N_H(\tilde{f}, \tilde{e})$ and $N(\tilde{e})$ are initialized from the whole training data. Considering a single sentence pair $(f_1^J, e_1^I)$ in training, we define $\bar{N}_H(\tilde{f}, \tilde{e})$ and $\bar{N}(\tilde{e})$ to be the local counts which $(f_1^J, e_1^I)$ contributed to the overall counts $N_H(\tilde{f}, \tilde{e})$ and $N(\tilde{e})$. These local counts can easily be computed from the current sentence pair $(f_1^J, e_1^I)$ and the corresponding word alignment $A$. Now we only have to subtract the local counts $\bar{N}_H(\tilde{f}, \tilde{e})$ and $\bar{N}(\tilde{e})$ from the initial counts $N_H(\tilde{f}, \tilde{e})$ and $N(\tilde{e})$ and we can re-estimate the phrase translation probabilities with:

$$\bar{p}_H(\tilde{f}|\tilde{e}) := \frac{N_H(\tilde{f}, \tilde{e}) - \bar{N}_H(\tilde{f}, \tilde{e})}{N(\tilde{e}) - \bar{N}(\tilde{e})} \tag{3.5}$$

An additional effect of this re-estimation is the loss of singleton phrases. Whenever we encounter a phrase or phrase pair which does not appear in the data set a second time, the re-estimated phrase counts $(N_H(\tilde{f}, \tilde{e}) - \bar{N}_H(\tilde{f}, \tilde{e}))$ and $(N(\tilde{e}) - \bar{N}(\tilde{e}))$ can both become zero, forcing us to remove those phrase pairs from the phrase table. This can render it impossible for the decoder to produce a segmentation $s_1^K$ for a sentence pair $(f_1^J, e_1^I)$ containing singleton phrases and thus deprive us of a part of

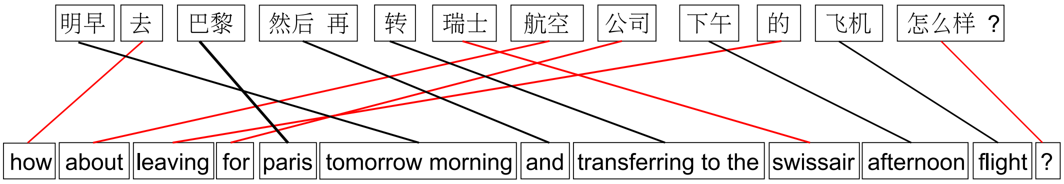

**Figure 3.4.** Top scoring segmentation in FA with standard leaving-one-out and with standard reordering model for the sentence from Figure 3.2. Incorrect phrase alignments are marked with red.

the training data. We found this part to be roughly 10% of the whole set on the IWSLT data described in Section 5.2. Therefore, instead of removing the phrases in question from the phrase table, we assigned them a very low probability. We experimented with two different methods of choosing this probability.

- **Standard leaving-one-out.** We assign a fixed probability $\alpha$ close to zero to singleton phrase pairs.

$$\bar{p}_H(\tilde{f}|\tilde{e}) := \begin{cases} \alpha & \text{if } N_H(\tilde{f},\tilde{e}) = \bar{N}_H(\tilde{f},\tilde{e}) \\ \dfrac{N_H(\tilde{f},\tilde{e}) - \bar{N}_H(\tilde{f},\tilde{e})}{N(\tilde{e}) - \bar{N}(\tilde{e})} & \text{else} \end{cases} \tag{3.6}$$

- **Length-based leaving-one-out.** The probability we assign to singleton phrase pairs is computed from a fixed probability $\beta$ and the summed lengths of source phrase $\tilde{f}$ and target phrase $\tilde{e}$.

$$\bar{p}_H(\tilde{f}|\tilde{e}) := \begin{cases} \beta^{(|\tilde{f}|+|\tilde{e}|)} & \text{if } N_H(\tilde{f},\tilde{e}) = \bar{N}_H(\tilde{f},\tilde{e}) \\ \dfrac{N_H(\tilde{f},\tilde{e}) - \bar{N}_H(\tilde{f},\tilde{e})}{N(\tilde{e}) - \bar{N}(\tilde{e})} & \text{else} \end{cases} \tag{3.7}$$

For our example, we can see in Figure 3.4 that with standard leaving-one-out the FA produces much shorter phrases. However, standard leaving-one-out has the drawback that in all sentences containing singletons the phrase translation model is again strongly biased towards segmenting the whole sentence as a single phrase pair. This is due to the fact that in each possible segmentation there is always at least one phrase pair $(\tilde{f},\tilde{e})$ with the low model probability $\bar{p}_H(\tilde{f}|\tilde{e}) = \alpha$ which is also assigned to the phrase pair spanning the whole sentence: $\bar{p}_H(f_1^J|e_1^I) = \alpha$. This is remedied by the length-based leaving-one-out method, where longer phrases are assigned lower model probabilities. Therefore singleton phrases can be learned from knowledge about phrase pairs from other sentences.

SOURCE: 这 架 相机 怎么样 ?

| phrase translation model | translation |
|---|---|
| heuristic model | how about this camera ? |
| trained model | does this camera ? |

REFERENCE: how about this camera ?

**Figure 3.5.** Example from IWSLT development set. The weighted count model (cf. Section 4.3.2) serves as the trained model.

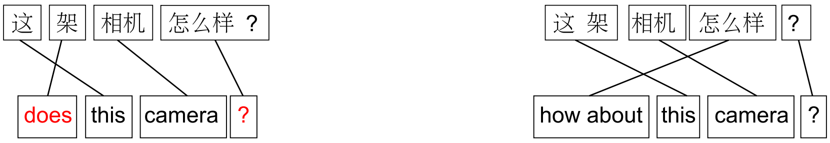

**Figure 3.6.** Different segmentations of the example shown in Figure 3.5. The left hand segmentation is produced by the weighted count model, the right hand segmentation by the heuristic. Incorrect phrase translations are displayed in red.

## 3.3 Reordering during training

Consider the example from the IWSLT development data set (cf. Section 5.2) shown in Figure 3.5 which contrasts the translations produced by the heuristic phrase model described in Section 2.3 with the one produced by the trained model which will be specified in Section 4.3.2. The corresponding phrase segmentations are displayed in Figure 3.6. For this example the heuristic model translates the source sentence correctly, whereas the trained model produces an incorrect translation. A closer look at the phrase tables reveals the reason for the incorrect translation by the trained phrase model: The inverse translation probability for the phrase ('*怎么样 ?*','*?*') is overestimated (cf. Table 3.3). The better segmentation on the right hand side in Figure 3.6 translates ' 怎么样' with '*how about*'.

**Table 3.3.** Phrase translation probabilities for the relevant phrase from the example in Figure 3.5.

| phrase model | $p$('怎么样 ?'\|'?') | $p$('?'\|'怎么样 ?') |
|---|---|---|
| heuristic model | 0.0013 | 0.11 |
| trained model | 0.00072 | 0.70 |
| trained model with $\lambda_{RM} = 0$ | 0.00032 | 0.54 |

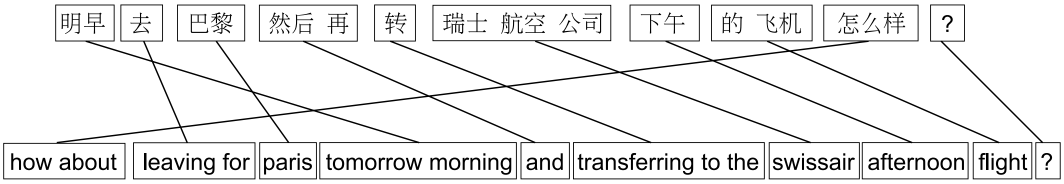

**Figure 3.7.** Top scoring segmentation for the sentence from Figure 3.2 produced by FA with standard leaving-one-out and $\lambda_{RM} = 0$.

The overestimation can be traced back to several sentence pairs in the training data. Figure 3.4 shows one of them together with the top scoring phrase segmentation produced by forced alignment. We observe, that the Chinese phrase ' 怎么样' is located at the end of the sentence, whereas the corresponding English phrase '*how about*' is the first phrase in the sentence. As a result, the correct segmentation is strongly penalized by the reordering model $h_{RM}$ which is therefore not among the top scoring in forced alignment. To remedy that, we propose to set the scaling factor for the reordering model to $\lambda_{RM} = 0$. It can be argued that most of the reordering information is contained in the words of the sentence pair and can be found by the translation model alone. Therefore the reordering model may not be needed to produce a segmentation $s_1^K$. Figure 3.7 shows the improved segmentation for the training example resulting from setting the reordering penalty $\lambda_{RM} = 0$ and in Table 3.3 we can see a slightly improved model probability.

## 3.4 Skips and Deletions

Another difficulty we are facing in training is that for some training sentences it can be impossible for the decoder to find a good segmentation. This may be due to non-literal translations or words from one language that do not have an equivalent in the other language. Consider for example the sentence pair in Figure 3.8.

The problem in this example is the repetition of the word '*he*' in the English sentence, whereas the corresponding '他' only appears once in the Chinese sentence. This makes it difficult for the decoder to produce a good segmentation. As a possible solution we suggest the introduction of a word omission model which would allow the decoder to ignore target words which do not correspond to any word in the source sentence and vice versa. In the following target word omissions will be referred to as skips and source word omissions as deletions.

We propose a very simple model for skips and deletions. In training, the decoder is allowed to omit single words. We introduce additional models $h_{SKIP}(\cdot,\cdot,\cdot)$ and

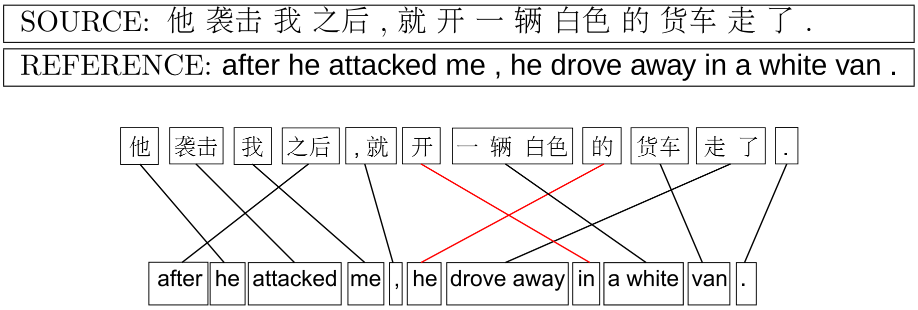
SOURCE: 他 袭击 我 之后 , 就 开 一 辆 白色 的 货车 走 了 .
REFERENCE: after he attacked me , he drove away in a white van .
他
袭击
我
之后
, 就
开
一 辆 白色
的
货车
走 了
.
after
he
attacked
me
,
he
drove away
in
a white
van
.

**Figure 3.8.** Example from IWSLT training data set with top scoring segmentation from FA. Incorrect phrase alignments are displayed in red.

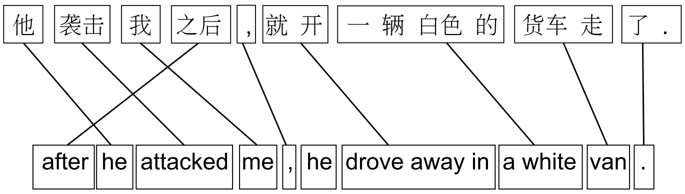

**Figure 3.9.** Top scoring segmentation for the sentence from figure 3.8 produced by FA with skip model.

$h_{DEL}(\cdot,\cdot,\cdot)$ which assign a fixed value as a penalty for each omitted word:

$$h_{SKIP}(e_1^I, s_1^K, f_1^J) = \gamma_{SKIP} \cdot (\#\text{ of skipped words}) \tag{3.8}$$

$$h_{DEL}(e_1^I, s_1^K, f_1^J) = \gamma_{DEL} \cdot (\#\text{ of deleted words}) \tag{3.9}$$

Furthermore we introduce an upper limit $C_{SKIP}$ and $C_{DEL}$ respectively for words that can be skipped or deleted in sequence. When experimenting with this model we set $C_{SKIP} = C_{DEL} = 1$.

Figure 3.9 shows the top scoring segmentation of the example from Figure 3.8 with the proposed skip model. Here, the additional '*he*' is skipped so that the decoder can produce a more reasonable phrase segmentation.

## 3.5 Phrase extensions

As an alternative to the skip and deletion model we experimented with phrase extensions. Instead of omitting the words in question we allow the decoder to extend phrases from the phrase table by a limited number of words with a fixed penalty. We define $cat(f_1^j, f'^{j'}_1) := f_1 \cdots f_j f'_1 \cdots f'_{j'}$ to be the concatenation of the phrases $f_1^j$ and $f'^{j'}_1$. When we extend a phrase $\tilde{f}$ by $f'^{j'}_1$, the phrase translation probabilities remain constant:

$$p(cat(\tilde{f}, f'^{j'}_1)|\tilde{e}) := p(\tilde{f}|\tilde{e}) \tag{3.10}$$

Extensions are possible on both source and target side, also at the same time, and both at the front and the back of the phrases. The corresponding probabilities are analogous to Equation (3.10). However, we introduce a model $h_{EXT}(\cdot,\cdot,\cdot)$ which assigns a constant penalty for each extension word:

$$h_{EXT}(e_1^I, s_1^K, f_1^J) = \gamma_{EXT} \cdot (\text{total \# of extension words}) \tag{3.11}$$

The maximum length of the extension phrase $f'^{j'}_1$ is restricted by $j' \leq C_{EXT}$. For our experiments we set $C_{EXT} = 1$.

Note that a phrase can only be extended if the resulting phrase pair does not already exist in the phrase table, whereas there is no such restriction on the skip and deletion words. Furthermore, the word-based model $h_{Lex}(\cdot,\cdot,\cdot)$ ignores omitted words, while extended phrases are regarded in the same way as any other phrase.

## 3.6 Phrase table initialization

### 3.6.1 Heuristic

The heuristic phrase extraction described in Section 2.3 represents the state of the art to estimate phrase translation probabilities in SMT. It is therefore our first choice for the initialization of the phrase table in training. However, there are some drawbacks to this.

Firstly, the quality of the phrase table is dependent on the provided word alignment. Flawed word alignments can disallow extracting the correct phrase pairs from a sentence pair and thus restrict the forced alignment procedure to choose from a pool of poor phrase pairs. Secondly, especially for rare phrases the distribution is already quite peaked and therefore contains a certain degree of determinism, which can make it hard for incorrectly estimated phrase pairs to be unlearned in training. An example is given in Table 3.4. It shows all phrase table entries for the Chinese phrase ' 公司 工作' extracted from the IWSLT training data set. We can see that the choice is limited to very few translations.

**Table 3.4.** All phrase table entries for the Chinese phrase $\tilde{f} =$'公司 工作' extracted by the heuristic from the IWSLT training data set.

| $\tilde{f}$ | $\tilde{e}$ | $p_H(\tilde{f}\|\tilde{e})$ | $p_H(\tilde{e}\|\tilde{f})$ |
|---|---|---|---|
| 公司 工作 | company | 0.023 | 0.10 |
| 公司 工作 | working for | 0.33 | 0.033 |
| 公司 工作 | the company | 0.17 | 0.033 |
| 公司 工作 | working | 0.018 | 0.033 |

### 3.6.2 PESA

As an alternative model to initialize the phrase table we considered the PESA model proposed by [Vogel 05]. They argue that phrase alignment should be considered separately from word alignment. Therefore, instead of a Viterbi word alignment only a word translation table is needed for this model.

To estimate the translation probability for a phrase pair $(\tilde{f}, \tilde{e}) = (f_{j_1}^{j_2}, e_{i_1}^{i_2})$ in a given sentence pair $(f_1^J, e_1^I)$ the standard IBM1 alignment model [Brown & Pietra$^+$ 93] is modified. Both source and target sentence are split into two components which are considered separately. One component models the likelihood of the phrase pair by itself, whereas the other component models context information by considering the rest of the sentence. This results in two numbers to be computed:

- **Inner sum.** For each word inside the source phrase we sum over the word translation probabilities of target words within the candidate target phrase.
- **Outer sum.** For each word outside the source phrase we sum over the word translation probabilities of target words outside of the candidate target phrase.

Additionally, the position alignment probability, which is $1/I$ for the IBM1 model, is modified to $1/|\tilde{f}| = 1/(i_2-i_1+1)$ for the inner sum and to $1/(I-|\tilde{f}|) = 1/(I-i_2+i_1-1)$ for the outer sum. We receive the following sentence level model:

$$\begin{aligned}\bar{p}_{PESA}(f_{j_1}^{j_2}|e_{i_1}^{i_2}) = \quad & \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} \frac{1}{i_2-i_1+1} p(f_j|e_i) \\ & \cdot \prod_{j \notin (j_1 \cdots j_2)} \sum_{i \notin (i_1 \cdots i_2)} \frac{1}{I-i_2+i_1-1}\, p(f_j|e_i) \qquad (3.12)\end{aligned}$$

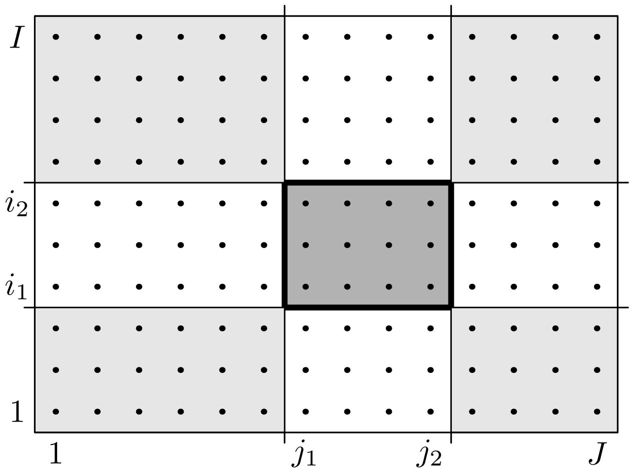

**Figure 3.10.** Illustration of the PESA phrase translation model. The inner sum is marked in a dark shade, the outer sum in light gray.

The inner and outer sum are illustrated in Figure 3.10.

Initialization with the PESA model rather than with the heuristic model can have the following advantages. There is no restriction on the phrases that can be extracted by a possibly flawed word alignment, yielding a larger pool of phrase pairs that can be chosen from in training. If we restrict the phrase length on the source side to $f_{max} = 6$ and on the target side to $e_{max} = 12$ we receive a PESA phrase table with 254M entries for the IWSLT data set, in opposition to 2.2M for the heuristic phrase table. In addition to that the overall distribution is softer than the one given by the heuristic. Therefore it is easier for forced alignment to recover from badly estimated phrase pairs. Table 3.5 shows the candidate phrases corresponding to the example in Table 3.4 produced by the PESA model.

For efficiency reasons we applied some pruning to the extracted phrases. For each sentence pair $(f_1^J, e_1^I)$ we extract only phrase pairs $(\tilde{f}, \tilde{e})$ with $\bar{p}_{PESA}(\tilde{f}, \tilde{e}) > 0.001$ and restrict the number of target phrases $\tilde{e}$ for each source phrase $\tilde{f}$ to 10. Thus we reduce the size of the phrase table to 5.5M entries for IWSLT.

Symmetrization of word translation probabilities for both translation directions is a widely used technique for translation related tasks. [Och & Ney 00] report that symmetric word translation models are more reliable and lead to improved word alignments. Therefore, for our experiments we chose to use a symmetrization of the IBM1 word translation model to compute the PESA probabilities:

**Table 3.5.** All phrase table entries for the Chinese phrase $\tilde{f}$ =’公司 工作’ estimated by the PESA model from the IWSLT training data set.

| $\tilde{f}$ | $\tilde{e}$ | $p_{PESA}(\tilde{f}\|\tilde{e})$ | $p_{PESA}(\tilde{e}\|\tilde{f})$ |
|---|---|---|---|
| 公司 工作 | company | 0.0094 | 0.52 |
| 公司 工作 | working for | 0.15 | 0.057 |
| 公司 工作 | work for | 0.011 | 0.052 |
| 公司 工作 | corporation | 0.00046 | 0.016 |
| 公司 工作 | the company | 0.0015 | 0.013 |
| 公司 工作 | company , | 0.0014 | 0.013 |
| 公司 工作 | job with | 0.0018 | 0.0094 |
| 公司 工作 | working for nanyo | 0.023 | 0.0035 |
| 公司 工作 | a job | 0.00016 | 0.0026 |
| 公司 工作 | been working for | 0.0040 | 0.0026 |
| 公司 工作 | to work | 0.00068 | 0.0025 |
| 公司 工作 | work | 0.0025 | 0.0017 |
| 公司 工作 | working | 0.0070 | 0.0011 |
| 公司 工作 | worked | 0.024 | 0.00070 |
| 公司 工作 | work at | 0.00039 | 0.00053 |
| 公司 工作 | worked at | 0.032 | 0.00029 |

$$p(f|e) = \frac{p_{IBM1}(f|e) + p_{IBM1}(e|f)}{2} \tag{3.13}$$

The IBM1 probabilities were trained with GIZA++. To obtain a phrase translation model for the whole training data, we interpret the sentence level probabilities $\bar{p}_{PESA}(\tilde{f}, \tilde{e})$ as phrase counts and compute the relative frequencies:

$$p_{PESA}(\tilde{f}|\tilde{e}) = \frac{N_{PESA}(\tilde{f}, \tilde{e})}{\sum\limits_{\tilde{f}'} N_{PESA}(\tilde{f}', \tilde{e})} \tag{3.14}$$

with

$$N_{PESA}(\tilde{f}, \tilde{e}) = \sum_{(f_1^J, e_1^I)} \bar{p}_{PESA}(\tilde{f}, \tilde{e}) \tag{3.15}$$

Like for the heuristic model we computed the PESA translation probabilities in both translation directions.

# 4 Phrase translation modeling

## 4.1 Word alignment re-estimation

In Section 2.3 we described how we can utilize the word alignment to build a phrase table by extracting the phrases that are consistent with the word alignment. We store the phrase level word alignment along with the translation probabilities for each extracted phrase pair in the phrase table. Thus we can use the single-best segmentation produced by forced alignment to generate a new word alignment for the training data. If we encounter different phrase level word alignments for the same phrase pair, the one with the highest count is chosen. By applying the phrase extraction heuristic on the re-estimated word alignment, we receive a new phrase table which can be used for translation.

To measure the distance between two alignments $A \subseteq I \times J$ and $A' \subseteq I \times J$ we introduce the alignment distance (AD) to be defined as follows:

$$AD(A, A') = 1 - \frac{2 \cdot |A \cap A'|}{|A| + |A'|} \tag{4.1}$$

Note that this definition is identical to the alignment error rate (AER) described in [Och & Ney 00] if we ignore the distinction between sure and possible alignment points. However, AER is always computed against a reference to measure the quality of an alignment $A$, whereas AD compares two alignments which are both not necessarily assumed to be correct.

Table 4.1 gives some statistics for the first four iterations on the IWSLT data (cf. Section 5.2). We compare the number of alignment points, the alignment distance (AD) to the alignment produced by GIZA++, the AD to the alignment from the previous iteration, and the number of phrases that are extracted by the heuristic.

**Table 4.1.**

| FA iteration |
|---|
| 0 (GIZA++) |
| 1 |
| 2 |
| 3 |
| 4 |

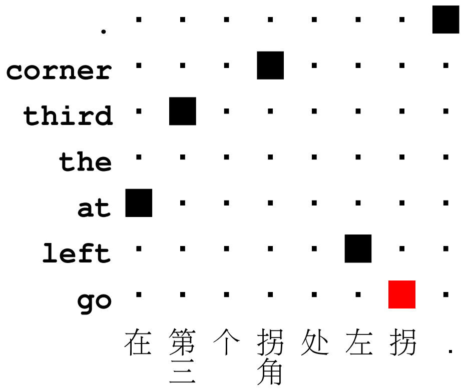

**Figure 4.1.** Word alignment from the IWSLT training data set. The black squares denote alignment points in both the word alignment produced by GIZA++ and after the first iteration of FA. The word alignment marked with the red square only appears after the first iteration of FA.

In Table 4.1 we observe that the difference to the original alignment increases with the number of iterations, while the difference to the previous iteration decreases, which indicates convergence. The most noticeable characteristic, however, is that after the second iteration the number of alignment points has increased by five percent. An inspection of the alignments reveals that in many cases the original alignment points are retained and word alignments are added for previously unaligned words. Figure 4.1 shows an example for this effect. As a result of the greater number of aligned words, the size of the resulting phrase table decreases, as phrase extraction becomes more restricted.

## 4.2 Indicator features

In training, the initial phrase table constitutes the phrase pairs that are available to compute the forced alignment. However, in most cases only a fraction of them are in fact used to produce the segmentations. For a given sentence pair $(f_1^J, e_1^I)$, the heuristic model for example can extract overlapping phrases and allows for more than one possible translation for the same phrase if words are unaligned. A valid phrase segmentation of $(f_1^J, e_1^I)$, however, requires each word to belong to exactly one phrase. Assuming that phrase pairs that were used in forced alignment are more likely to produce good translations at decoding time than phrase pairs that were not, we want to distinguish between those two subsets of the original phrase table. Therefore, in addition to the models specified in Section 2.2.3 we propose a binary phrase level feature $\bar{h}_{Ind}(\tilde{f}, \tilde{e})$ which fires for each phrase pair $(\tilde{f}, \tilde{e})$ that was encountered in training. For a given sentence pair $(f_1^J, e_1^I)$ with phrase segmentation $s_1^K$, $s_k = (i_k; b_k, j_k)$, we obtain the following sentence level model:

$$h_{Ind}(e_1^I, s_1^K, f_1^J) := \sum_{k=1}^{K} \bar{h}_{Ind}(\tilde{f}_k, \tilde{e}_k) \tag{4.2}$$

with

$$\bar{h}_{Ind}(\tilde{f}, \tilde{e}) := \begin{cases} 1 & \text{if } (\tilde{f}, \tilde{e}) \text{ was seen in training} \\ 0 & \text{else} \end{cases} \tag{4.3}$$

Here $\tilde{f}_k$ and $\tilde{e}_k$ are defined as in Equations (2.2) and (2.3). A phrase pair $(\tilde{f}, \tilde{e})$ is defined to be seen in training if it occurs in the single-best segmentation $\hat{s}_1^{\hat{K}}$ for at least one sentence pair in training. The scaling factor for this feature is trained by MERT (cf. Section 2.2.5) along with the weights for the other models.

For the IWSLT data, about 3% of the heuristically extracted phrase pairs were seen in training. We can argue that if only this small number of phrases is assigned a firing indicator feature, we should not expect a significant difference in translation performance. Therefore, we also experimented with extending the definition of the indicator features $\bar{h}_{Ind}(\tilde{f}, \tilde{e})$ to $N$-best lists.

**Table 4.2.** Number of phrases seen in training with different sizes $N$ of the $N$-best list on the IWSLT data set. The initial phrase table has 2 187 004 entries and was produced by the heuristic described in Section 2.3.

| $N$ | # seen phrase pairs |
|---|---:|
| 1 | 70 626 |
| 10 | 204 893 |
| 20 | 272 447 |
| 50 | 375 994 |

An $N$-best list is the set of the $N$ highest scoring segmentations for a sentence pair. Then we define a phrase pair to be seen in training if it occurs in a segmentation from an $N$-best list for at least one sentence pair in training. In Table 4.2 you can see the growing number of phrases seen in training with increasing $N$.

## 4.3 Generative phrase models

### 4.3.1 Count model

The simpler of our generative phrase models estimates phrase translation probabilities by their relative frequencies in the training data, similar to the heuristic model (cf. Section 2.3) but with counts from training rather than on the basis of a word alignment:

$$p_C(\tilde{f}|\tilde{e}) = \frac{N_C(\tilde{f},\tilde{e})}{\sum\limits_{\tilde{f}'} N_C(\tilde{f}',\tilde{e})} \tag{4.4}$$

with

$$N_C(\tilde{f},\tilde{e}) = \sum_{(f_1^J,e_1^I,\hat{s}_1^{\hat{K}})} \sum_{\substack{k=1\ldots\hat{K},\\ \hat{s}_k=(i_k,b_k,j_k)}} \delta(\tilde{f},\tilde{f}_k)\cdot\delta(\tilde{e},\tilde{e}_k) \tag{4.5}$$

where $(f_1^J, e_1^I)$ are sentence pairs from the training data, $\hat{s}_1^{\hat{K}}$ is the corresponding single-best segmentation produced by FA, $\tilde{f}_k$ and $\tilde{e}_k$ are defined as in Equations (2.2) and (2.3) and $\delta(\cdot,\cdot)$ is the Kronecker function:

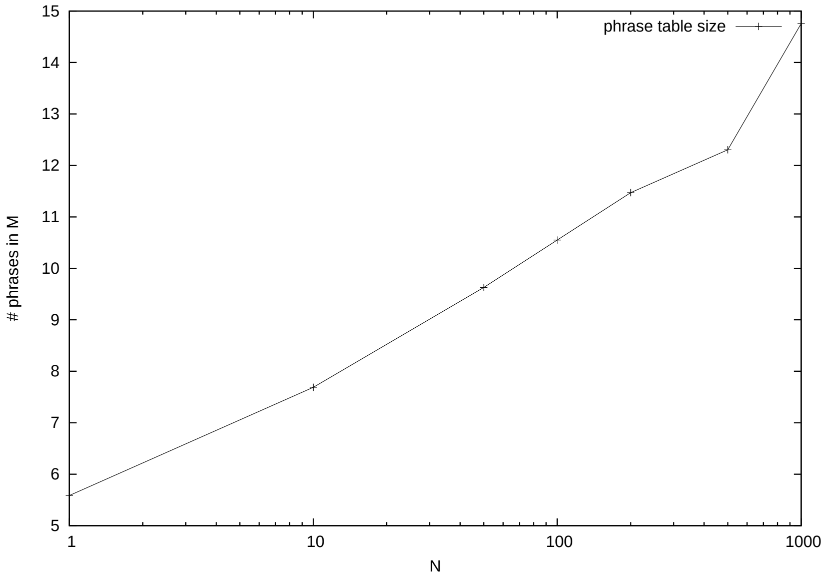

**Figure 4.2.** Phrase table size plotted against the number $N$ of $N$-best segmentations on a logarithmic scale for the Europarl data set (cf. Section 5.2).

$$\delta(\tilde{f}, \tilde{f}') := \begin{cases} 1 & \text{if } \tilde{f} = \tilde{f}' \\ 0 & \text{else} \end{cases} \tag{4.6}$$

With this model we can only assign translation probabilities to phrase pairs seen in training. As described in Section 4.2 those are only a fraction of the phrase pairs in the initial phrase table. Therefore the size of the phrase table is reduced in comparison with initialization, which may be desirable with respect to memory load at translation time. In the Europarl data, the trained count model has about 5.9M entries in opposition to 86M for the heuristic model. Note that the phrase pairs in the phrase table of the count model are exactly the phrases which would be assigned firing indicator features in this setup.

Analogous to the indicator features, the count model can be extended for $N$-best lists by taking the counts from the $N$-best lists rather than only from a single-best segmentation. The size of the phrase table increases with $N$. Figure 4.2 shows the number of phrases in the phrase table plotted against the size of the $N$-best list. For $N = 1000$ we reach a size of 15M entries.

### 4.3.2 Weighted count model

The probability distribution $Pr(s_1^K|e_1^I, f_1^J)$ over the space of possible segmentations $s_1^K$ is modeled by $p_{FA}(s_1^K|e_1^I, f_1^J)$ as specified in Equation (3.1). Thus we can interpret $p_{FA}(s_1^K|e_1^I, f_1^J)$ as the confidence of the system that $s_1^K$ is a correct segmentation for the sentence pair $(f_1^J, e_1^I)$. Assuming that this confidence is a measure for the quality of the different segmentations, we can argue that phrase pairs occurring in a segmentation with a high confidence should during translation be preferred to phrase pairs from a segmentation with low confidence. To take this into account we developed the weighted count model.

To make use of the model distribution we update the count model in the following way. Phrase translation probabilities are still estimated by their relative frequencies, but the counts are taken from the whole space of possible segmentations $s_1^K$, weighting them with their corresponding model probability:

$$p_W(\tilde{f}|\tilde{e}) = \frac{N_W(\tilde{f}, \tilde{e})}{\sum\limits_{\tilde{f}'} N_W(\tilde{f}', \tilde{e})} \tag{4.7}$$

with

$$N_W(\tilde{f}, \tilde{e}) = \sum_{(f_1^J, e_1^I)} \sum_{s_1^K} \sum_{\substack{k=1\dots K, \\ s_k=(i_k, b_k, j_k)}} \delta(\tilde{f}, \tilde{f}_k) \cdot \delta(\tilde{e}, \tilde{e}_k) \cdot p_{FA}(s_1^K|e_1^I, f_1^J) \tag{4.8}$$

where $(f_1^J, e_1^I)$ are sentence pairs from the training data, $\tilde{f}_k$ and $\tilde{e}_k$ are defined as in Equations (2.2) and (2.3) and $\delta(\cdot, \cdot)$ is defined as in Equation (4.6). The definition of the weighted counts $N_W(\tilde{f}, \tilde{e})$ is a generalization of $N_C(\tilde{f}, \tilde{e})$. If the whole probability mass is concentrated on the single-best segmentation, $N_C(\tilde{f}, \tilde{e})$ and $N_W(\tilde{f}, \tilde{e})$ are identical:

$$p_{FA}(\hat{s}_1^{\hat{K}}|e_1^I, f_1^J) := 1 \qquad \Rightarrow \qquad N_C(\tilde{f}, \tilde{e}) = N_W(\tilde{f}, \tilde{e}) \tag{4.9}$$

This model is very similar to the one proposed in [DeNero & Gillick$^+$ 06], but shows two significant differences. Firstly, [DeNero & Gillick$^+$ 06] only allow the segmentation probability to be $p_{FA}(s_1^K|e_1^I, f_1^J) > 0$ if the phrase pair $(\tilde{f}, \tilde{e})$ is consistent with the Viterbi word alignment. We do not have this restriction, allowing us to recover from flaws in the word alignment. Secondly we can hope to circumvent the overfitting issues reported in [DeNero & Gillick$^+$ 06] by application of the leaving-one-out method for estimating the segmentation probability $p_{FA}(s_1^K|e_1^I, f_1^J)$.

As enumerating the whole search space of possible segmentations $s_1^K$ for summation is infeasible, in our implementation the weighted counts $N_W(\tilde{f}, \tilde{e})$ are approximated. Instead of doing an exhaustive search, for each sentence pair $(f_1^J, e_1^I)$ the counts are computed from an $N$-best list. For all our experiments we set $N = 1000$. The probability mass is generally concentrated on the top scoring few segmentations. On the IWSLT training data, on average the ten top scoring segmentations encompass 88% and the 100 top scoring segmentations 98% of the whole probability mass occupied by the 1000 best segmentations. Therefore we can assume that this approximation does not have a significant impact on the resulting phrase table.

### 4.3.3 Phrase prior probabilities

One of the advantages of the heuristic model in Section 2.3 is that in addition to modeling the phrase translation probabilities it implicitly contains a segmentation model. Taking a close look at Equation (2.22) we note that the normalizing factor $N(\tilde{e})$ is not the marginalization of the phrase count over the target phrases $\tilde{e}$. This is different from the generative phrase models in Equations (4.4) and (4.7). We have mentioned in Section 2.3.3 that using the monolingual count $N(\tilde{e})$ as normalizing factor rather than the proper marginalization results in a deficient phrase model, as summation to unity does not hold. However, it has the advantage of contributing information about the phrase segmentation. When the word alignment for a given sentence pair $(f_1^J, e_1^I)$ prohibits the extraction of all phrase pairs containing a specific target phrase $\tilde{e}$, it contributes to the monolingual count $N(\tilde{e})$ but not to the phrase count $N_H(\tilde{f}, \tilde{e})$ for any source phrase $\tilde{f}$. The reason is that no phrase segmentation consistent with the word alignment contains the phrase $\tilde{e}$. This is reflected by the heuristic model assigning low translation probabilities $p_H(\tilde{f}|\tilde{e})$ to the corresponding phrase pairs. Therefore, monolingual phrases, which are consistent with the word alignment more often, are preferred in translation. This is useful for two reasons. Firstly, it penalizes rare phrase pairs, whose probability can not be estimated reliably due to lack of examples to be learned from, and as a result are often overestimated. Secondly, it downweights monolingual phrases that are part of a longer idiom and should therefore not be considered separately. Consider the example in Figure 4.3. The right hand translation produced by the weighted count model fails to translate the Chinese character ' 买', which means '*buy*'. The reason is the overestimation of the probability of the phrase pair (' 买 一 个','*a*'). The Chinese phrase '买 一 个' appears ten times in the training data. However, in FA it occurs in only two sentences, both of which do not contain the word '*buy*' on

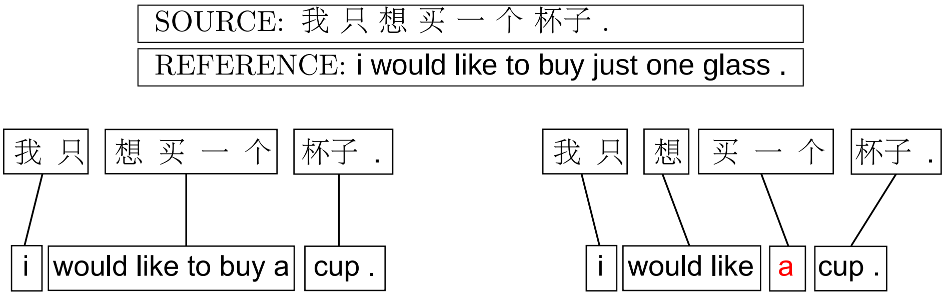

**Figure 4.3.** Example from IWSLT development set. The left hand translation was produced with the heuristic model, the right hand translation by the weighted count model. Incorrect phrase translations are displayed in red.

the English side. Therefore it is impossible to learn the correct translation. However, we can hope to improve translation quality for this example by exploiting the segmentation information.

The generative phrase models we have introduced so far fail to take the monolingual segmentations into account. We will discuss two ways of tackling this shortcoming here.

### Combined segmentation and translation model

A straightforward approach to this problem is to mimic the method employed for the heuristic model. We can hope to obtain the same level of additional segmentation information by simply replacing the marginal counts $\sum_{\tilde{f}'} N_C(\tilde{f}', \tilde{e})$ and $\sum_{\tilde{f}'} N_W(\tilde{f}', \tilde{e})$ with the monolingual phrase count $N(\tilde{e})$ as a normalization factor. We obtain the following models:

$$p_{C_{SEG}}(\tilde{f}|\tilde{e}) = \frac{N_C(\tilde{f}, \tilde{e})}{N(\tilde{e})} \tag{4.10}$$

$$p_{W_{SEG}}(\tilde{f}|\tilde{e}) = \frac{N_W(\tilde{f}, \tilde{e})}{N(\tilde{e})} \tag{4.11}$$

Figure 4.4 shows the improved translation of the combined segmentation and weighted count model.

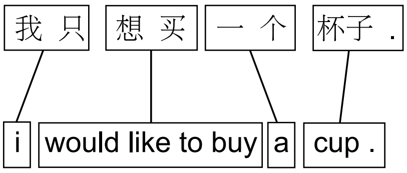

**Figure 4.4.** Translation produced by the combined segmentation and weighted count model for the example from Figure 4.3.

### Separate segmentation model

[Shen & Delaney+ 08] suggest a separate phrase segmentation model which is trained using forced alignment. They assume that the segmentation of each phrase is independent and augment the phrase translation model with a prior probability on source and target phrase. Thus the probability of a segmentation $s_1^K$ of a sentence pair $(f_1^J, e_1^I)$ into $K$ phrases is modeled as:

$$p_{SEG}(s_1^K|e_1^I, f_1^J) := \prod_{k=1}^{K} \Big(p(\tilde{f}_k) \cdot p(\tilde{e}_k)\Big) \tag{4.12}$$

$\tilde{f}_k$ and $\tilde{e}_k$ are defined as in Equations (2.2) and (2.3). Adopting this idea we propose to incorporate two new models $h_{eSEG}$ and $h_{fSEG}$ into our log-linear framework. To fit our phrase models we adapt the modeling used in [Shen & Delaney+ 08] and obtain:

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

with

$$p(\tilde{e}) = \frac{\sum_{\tilde{f}'} N_*(\tilde{f}', \tilde{e})}{N(\tilde{e})} \tag{4.14}$$

Again, $\tilde{f}_k$ and $\tilde{e}_k$ are defined as in Equations (2.2) and (2.3). The definition of $h_{fSEG}$ is analogous $h_{eSEG}$. To ensure consistency with the phrase translation model, for

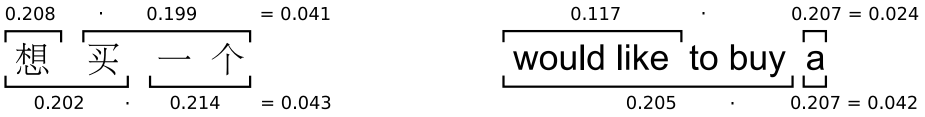

**Figure 4.5.** Illustration of the prior probabilities for the relevant part of the example in Figure 4.3. On the left hand side $p(\tilde{f})$ is shown for different source segmentations. On the right hand side $p(\tilde{e})$ is shown for different target segmentations. The phrase priors are computed from the weighted counts $N_W(\cdot,\cdot)$.

the bilingual phrase counts $N_*$ we insert either $N_C$ or $N_W$, depending on which phrase model is being used.

Note that the combined model described above is a special case of this separate model. If we link the the scaling factors of the phrase priors to the corresponding phrase translation model, setting $\lambda_{eSEG} = \lambda_{Phr}$ and $\lambda_{fSEG} = \lambda_{iPhr}$, the two approaches are identical.

In Figure 4.5 we can see the resulting phrase priors for our example sentence. It shows that the phrase priors favor the better segmentation given in Figure 4.4 over the flawed one produced without the priors.

## 4.4 Interpolation of phrase tables

It was mentioned, that [DeNero & Gillick$^+$ 06] found their generative phrase model to clearly underperform the heuristic model. Their analysis identifies one of the major problems to be the high level of determinism in the generative model, which can be quantified by a low entropy. They discuss several smoothing methods in order to retain the original entropy. Out of these they found phrase table interpolation with the heuristic model to work best. Improvements over the pure heuristic model by interpolating the phrase tables produced by the two approaches are reported.

We experimented with two types of phrase table interpolation, linear and log-linear.

**Linear interpolation:**

$$p_{LIN}(\tilde{f}|\tilde{e}) = \omega \cdot p_H(\tilde{f}|\tilde{e}) + (1-\omega) \cdot p_*(\tilde{f}|\tilde{e}) \tag{4.15}$$

where $\omega$ is the interpolation weight and for $p_*$ we insert one of our generative models, either $p_C$ or $p_W$.

At decoding time, a phrase pair $(\tilde{f},\tilde{e})$ is only usable, if it is assigned a sufficiently high probability $p(\tilde{f},\tilde{e})$. Therefore, we can interpret linear interpolation as taking the union of the two phrase tables: to be usable after the interpolation, in most cases it will be sufficient for a phrase pair if its probability is high enough in one of the phrase tables.

**Log-linear interpolation:**

$$p_{LL}(\tilde{f}|\tilde{e}) = \Big(p_H(\tilde{f}|\tilde{e})\Big)^{\omega} \cdot \Big(p_*(\tilde{f}|\tilde{e})\Big)^{(1-\omega)} \tag{4.16}$$

Again, $\omega$ is the interpolation weight. Log-linear interpolation can be interpreted as taking the intersection of the two phrase tables: to be usable after the interpolation, a phrase pair needs to have a high probability in both phrase tables. For the Europarl data, Figures 4.6 and 4.7 show the performance in Bleu and TER (cf. Section 5.1) respectively of the two interpolation methods plotted against the interpolation weight $\omega$. Here, the count model (cf. Section 4.3.1) and the heuristic model (cf. Section 2.3.3) are interpolated. The scaling factors $\lambda_1^M$ are trained for the heuristic model and are kept fixed. We can see that for both methods the curves are approximately convex. The maximum for the linear interpolation is reached for a weight $\omega$ close to zero, meaning the trained model has a higher weight. For the log-linear interpolation the best performance is reached for a weight $\omega$ close to one, which means the heuristic model is given a higher weight.

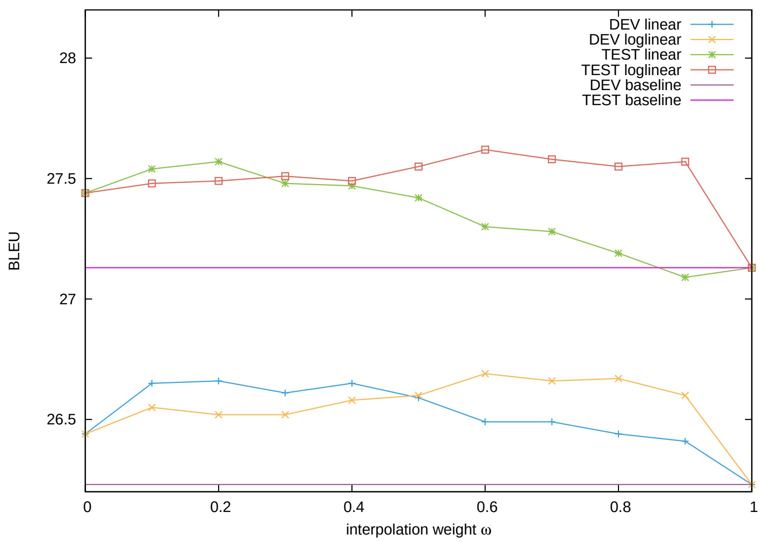

**Figure 4.6.** Performance of linear and log-linear interpolation of phrase tables on DEV and TEST the Europarl data set (cf. Section 5.2) measured in Bleu score. The scaling factors $\lambda_1^M$ are kept fixed. $\omega = 0$ is equivalent to the count model and $\omega = 1$ is equivalent to the heuristic model.

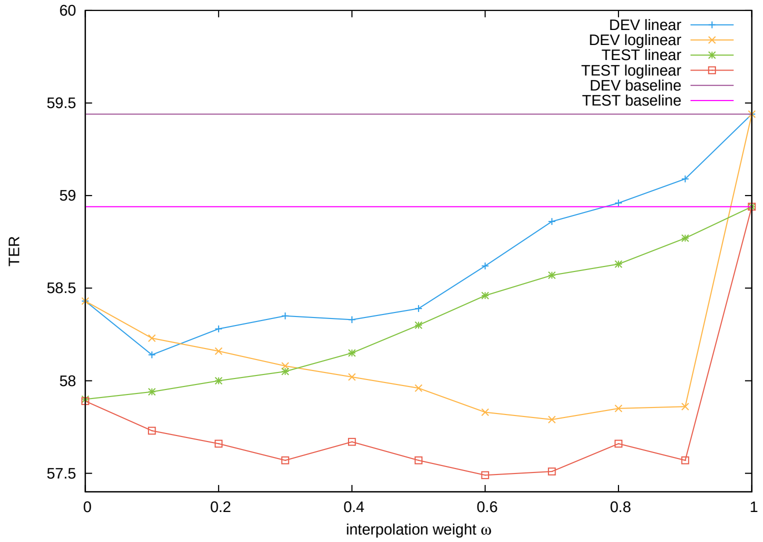

**Figure 4.7.** Performance of linear and log-linear interpolation of phrase tables on DEV and TEST the Europarl data set measured in TER. The scaling factors $\lambda_1^M$ are kept fixed. $\omega = 0$ is equivalent to the count model and $\omega = 1$ is equivalent to the heuristic model.

# 5 Experiments

In this chapter we will first review the evaluation measures we used and give a description of the data sets on which experiments were conducted. After that we specify the experimental setup and give the results, which are discussed in detail. One of our proposed methods, namely the interpolation of the phrase tables produced by the weighted count model (cf. Section 4.3.2) and the heuristic (cf. Section 2.3), consistently outperforms the baseline system on both data sets.

## 5.1 Evaluation measures

Automatic evaluation of different MT systems is a difficult task and there is no established standard metric. Given different hypothesis translations, it can be hard even for human experts to decide which one is the best. Some statistics for meta-evaluation of a translation task are given in [Callison-Burch & Fordyce$^{+}$ 08], showing that human annotators disagreed on the ranking of two translation hypotheses in 42% of the cases. Further, human evaluation is expensive and not feasible for any meaningful amount of experimental data. However, there are a number of evaluation metrics available, all of which have different shortcomings and advantages. Those metrics assign a score or error rate to a hypothesis translation by comparison with one or more reference translations. The two error metrics we chose for evaluation of our results are the Bleu score and the translation edit rate (TER), which are commonly used for current research. Additionally, in some of our experiments we need to evaluate word alignment quality. The standard evaluation metric for this purpose is the alignment error rate (AER).

### 5.1.1 Bleu score

Currently the most popular evaluation metric for machine translation systems is the Bleu score, which was introduced by [Papineni & Roukos$^{+}$ 02]. The results reported in [Callison-Burch & Osborne$^{+}$ 06] show that it has shortcomings when comparing conceptually different systems but can be considered reliable if used to compare variants of the same system. The Bleu score is an accuracy measure. It is a combination of the geometric mean of $n$-gram precisions and a brevity penalty, which penalizes too short hypotheses. Given a reference translation $\hat{e}_1^{\hat{I}}$, the Bleu score for a hypothesis $e_1^I$ is computed as follows:

$$\textsc{Bleu}(e_1^I, \hat{e}_1^{\hat{I}}) := BP(I, \hat{I}) \cdot \prod_{n=1}^{4} Prec_n(e_1^I, \hat{e}_1^{\hat{I}})^{1/4} \tag{5.1}$$

with

$$BP(I, \hat{I}) := \begin{cases} 1 & \text{if } I \geq \hat{I} \\ \exp(1 - I/\hat{I}) & \text{if } I < \hat{I} \end{cases} \tag{5.2}$$

$$Prec_n(e_1^I, \hat{e}_1^{\hat{I}}) := \frac{\sum\limits_{w_1^n} \min \left\{ C(w_1^n | e_1^I), C(w_1^n | \hat{e}_1^{\hat{I}}) \right\}}{\sum\limits_{w_1^n} C(w_1^n | e_1^I)} \tag{5.3}$$

Here, $C(w_1^n|e_1^I)$ denotes the count, i.e. the number of occurrences, of an $n$-gram $w_1^n$ in a sentence $e_1^I$. The denominator of the $n$-gram precision evaluates to the number of $n$-grams in the hypothesis, i.e. $I - n + 1$. The Bleu score can be extended to take more than one reference translation into account. We apply document level Bleu, meaning that the $n$-gram counts are collected over the whole data set rather than on the basis of single sentences.

### 5.1.2 TER

The translation edit rate [Snover & Dorr$^+$ 06] is an error metric. It counts the number of edits required to change a hypothesis into one of the reference translations. In contrast to the well-known word error rate (WER) it allows shifts, i.e. movements of contiguous word sequences within the hypothesis. The other edit operations available are insertions, deletions and substitutions of single words. All edits, including shifts of any number of words by any distance, have equal cost. The number of edit operations is divided by the average number of reference words per sentence.

$$TER = \frac{\text{\# of edits}}{\text{avg. \# of reference words}} \tag{5.4}$$

When we have more than one reference translation, the TER is defined by the minimum number of edits needed to produce one of the references. TER is also computed on document level, meaning the edit counts are collected over the whole data set.

### 5.1.3 AER

In Section 4.1 we already defined alignment distance (AD) as a measure of how much two word alignments differ. AER [Och & Ney 00] works in a similar way by measuring the distance to a reference word alignment produced by human experts. However, in contrast to AD, in the reference alignment we distinguish between two kinds of alignment points: an $S$ (sure) alignment which is used for unambiguous alignments and a $P$ (possible) alignment which is used for alignments that might or might not exist ($S \subseteq P$).

The alignment error rate of an alignment $A \subseteq I \times J$ with a reference word alignment composed of the sure alignment points $S \subseteq I \times J$ and the possible alignment points $P \subseteq I \times J$ is based on precision and recall:

$$recall = \frac{|A \cap S|}{|S|}, \;\; precision = \frac{|A \cap P|}{|A|} \tag{5.5}$$

and the following error rate:

$$AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{5.6}$$

Note that $AER(S, S; A) = AD(S, A)$. A detailed description of the alignment error rate can be found in [Och & Ney 00].

## 5.2 Data sets

The *International Workshop on Spoken Language Translation* (IWSLT) [Fordyce 07] is an MT evaluation campaign organized by the *Consortium for Speech Translation Advanced Research* (C-Star). Its focus is the translation of spoken language in the travel domain. The principal source for training, development and evaluation data is the *Basic Travel Expression Corpus* (BTEC) [Takezawa & Sumita$^+$ 02]. Our training data set ($TRAIN$) consists of the Chinese-English part of the training

**Table 5.1.** Statistics for the IWSLT Chinese-English data

| | | Chinese | English |
|---|---|---|---|
| TRAIN | Sentences | 42 942 | |
| | Running Words | 380 259 | 420 431 |
| | Vocabulary | 11 760 | 9 933 |
| | Singletons | 4 637 | 3 937 |
| DEV | Sentences | 500 | |
| | Running Words | 3 578 | 62 520 |
| | Vocabulary | 950 | 3 878 |
| | OOVs (running words) | 75 | 28 177 |
| TEST | Sentences | 506 | |
| | Running Words | 3 837 | 63 525 |
| | Vocabulary | 938 | 4 099 |
| | OOVs (running words) | 73 | 28 913 |

data released for the 2007 IWSLT evaluation and the evaluation data released for the 2003, 2006 and 2007 IWSLT evaluations. The evaluation data sets for the 2004 and 2005 IWSLT evaluation campaigns serve as our development ($DEV$) and test ($TEST$) set. Both $DEV$ and $TEST$ contain 16 English reference translations for each Chinese sentence. Table 5.1 shows some statistics on the data sets. It includes the number of words occurring only once (singletons) for $TRAIN$ and the number of words not occurring in the training data (out-of-vocabulary, OOV) for $DEV$ and $TEST$. The IWSLT data set is a small corpus with roughly 43K training sentences, which allows rapid experimenting.

The Europarl corpus [Koehn 05] is collected from the proceedings of the European Parliament. From the 11 available languages we chose the language pair German-English. For $DEV$ and $TEST$ we take the development and test set published for the *ACL 2008 Workshop on Statistical Machine Translation* (WMT08). Here, we have only one reference translation for each German sentence in $DEV$ and $TEST$. With roughly 1.3M training sentences the Europarl data set is considerably larger than the IWSLT corpus. Therefore experiments require much more computation time and we use it only to run a few choice experiments which showed promise on the IWSLT data. Statistics for the Europarl data are given in Table 5.2.

**Table 5.2.** Statistics for the Europarl German-English data

| | | German | English |
|---|---|---|---|
| TRAIN | Sentences | 1 311 815 | |
| | Running Words | 34 398 651 | 36 090 085 |
| | Vocabulary | 336 347 | 118 112 |
| | Singletons | 168 686 | 47 507 |
| DEV | Sentences | 2 000 | |
| | Running Words | 55 118 | 58 761 |
| | Vocabulary | 9 211 | 6 549 |
| | OOVs (running words) | 284 | 77 |
| TEST | Sentences | 2 000 | |
| | Running Words | 56 635 | 60 188 |
| | Vocabulary | 9 254 | 6 497 |
| | OOVs (running words) | 266 | 89 |

## 5.3 Results

### 5.3.1 Experimental setup

The experiments we ran to evaluate the different training methods and phrase models we propose have the following setup. We are given the three data sets $TRAIN$, $DEV$ and $TEST$. For our baseline, we first use GIZA++ to compute the Viterbi word alignment on $TRAIN$. Next we apply the heuristic described in Section 2.3 to obtain a phrase table by extraction of phrases from the word alignment. The scaling factors $\lambda_1^M$ are computed with MERT (cf. Chapter 2.2.5) on the $DEV$ data set.

The phrase table used for the baseline is also used to initialize FA. Alternatively, we can compute a phrase table with the PESA model (cf. Section 3.6.2) from $TRAIN$ for initialization. Then, FA is run on the training data $TRAIN$ from which we obtain the phrase segmentations $s_1^K$. Those are used in the following ways:

- **Word alignment.** If we have stored the within-phrase word alignment in the initial phrase table, we can use the segmentations $s_1^K$ to produce a new word alignment from which we can extract a phrase table using the heuristic, as was described in Section 4.1.
- **Indicator features.** We can augment the initial phrase table by adding features indicating whether a phrase was used in FA (cf. Section 4.2).

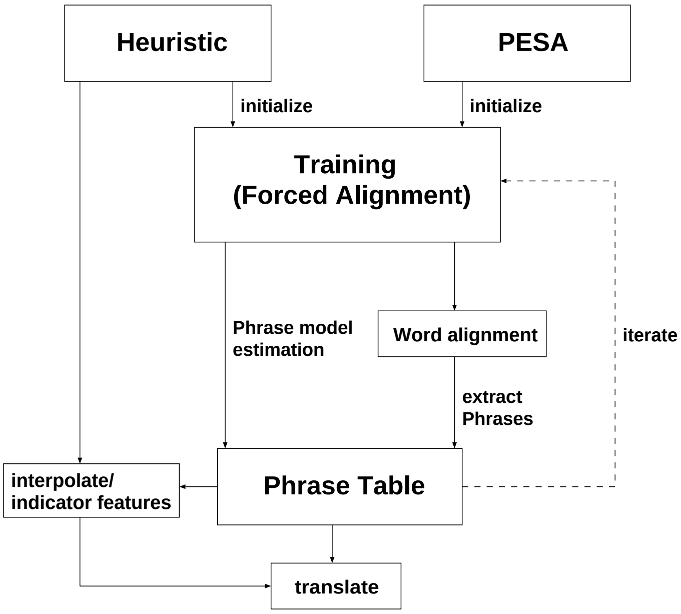

**Figure 5.1.** Illustration of the experimental setup.

- **Generative phrase models.** We can use the counts from the segmentations $s_1^K$ to build a phrase table according to the proposed phrase models which were introduced in Chapters 4.3.1 and 4.3.2.
- **Phrase segmentation model.** In Section 4.3.3 we proposed two different phrase segmentation models which assign prior probabilities to the monolingual phrases. These are estimated from phrase counts in FA.

Afterwards, the scaling factors are trained on $DEV$ for the new phrase table. By feeding back the new phrase table into FA we can reiterate the training procedure. When training is finished the performance is evaluated on $DEV$ and $TEST$. Alternatively, we can apply interpolation of the new phrase table with the original one estimated heuristically (cf. Chapter 2.3), retrain the scaling factors and evaluate afterwards. An illustration of the experimental setup is given in Figure 5.1.

### 5.3.2 Training setup experiments

In Chapter 3 we discussed several different setups for the forced alignment procedure, whose performance we evaluated on the IWSLT data. Two different implementations of leaving-one-out and a simple phrase length restriction were considered to counteract the tendency of FA to overfitting and overestimating long phrases. Furthermore, we proposed two different ways of dealing with words in the source or target sentence that have no correspondence to a word in the other, a word omission model and a phrase extension model. In addition to that, we considered to overcome problems arising from different word order by dropping the reordering penalty. Here, we will compare the effectiveness of these setups. We chose to evaluate them by their performance with the count model (cf. Section 4.3.1) estimated on an $N$-best list with $N = 1000$. For initialization we used the heuristic (cf. Section 2.3) and MERT was done for Bleu score. We compare the following setups:

- **Standard FA.** Here the setup is identical to the one used in free translation, except for the restriction on the target sentence.
- **Restricted phrase length.** The maximum phrase length is restricted to $f_{max} = 4$ and $e_{max} = 8$.
- **Standard l1o.** The standard leaving-one-out method is applied in training. As a penalty we set $-log(\alpha) = 200$.
- **Length-based l1o.** Here we apply the length-based leaving-one-out method. As a penalty we set $-log(\beta) = 2$
- $\lambda_{RM} = 0$**.** In these experiments there is no penalty on reordering.
- **Skip/del.** Skips and deletions are allowed as described in Section 3.4. The penalties are set to $\gamma_{SKIP} = \gamma_{DEL} = 10$.
- **Phrase ext.** We allow phrase extensions (cf. Section 3.5). To facilitate direct comparison we set the penalty to be identical to the skip and deletion model: $\gamma_{DEL} = 10$.

The values for the different penalties are hand adjusted based on experience and performance on a few choice training sentences.

Table 5.3 shows the results. We can clearly see that systems trained with leaving-one-out are superior to the one trained with standard FA. On average the phrase length restriction also works better than standard FA, but is outperformed by leaving-one-out. It is not clear, which of the leaving-one-out methods is superior. Solely using either skip and deletion models or phrase extensions does not show much promise. Dropping the reordering penalty by setting $\lambda_{RM} = 0$ does not prove to have any effect on the translation performance. However, when combining

**Table 5.3.** Performance of the different training setups discussed in Section 3 on the IWSLT data.

| | setup | Bleu | TER |
|---|---|---|---|
| **DEV** | baseline | 57.4 | 34.9 |
| | standard FA | 56.0 | 36.7 |
| | restricted phrase length | 55.7 | 36.3 |
| | standard l1o | 56.7 | 35.8 |
| | length-based l1o | 57.1 | 35.6 |
| | standard l1o + skip/del | 56.7 | 35.2 |
| | standard l1o + phrase ext. | 56.2 | 36.3 |
| | standard l1o + $\lambda_{RM} = 0$ | 56.9 | 34.9 |
| | standard l1o + restr. phr. len. | 57.0 | 35.3 |
| | length-based l1o + skip/del + $\lambda_{RM} = 0$ | 56.7 | 35.4 |
| **TEST** | baseline | 61.8 | 29.9 |
| | standard FA | 60.7 | 31.3 |
| | restricted phrase length | 61.4 | 30.8 |
| | standard l1o | 61.7 | 31.4 |
| | length-based l1o | 61.2 | 30.8 |
| | standard l1o + skip/del | 61.0 | 31.4 |
| | standard l1o + phrase ext. | 61.3 | 31.4 |
| | standard l1o + $\lambda_{RM} = 0$ | 61.1 | 30.9 |
| | standard l1o + restr. phr. len. | 62.1 | 30.9 |
| | length-based l1o + skip/del + $\lambda_{RM} = 0$ | 61.7 | 30.5 |

length-based leaving-one-out with skip and deletion models and setting $\lambda_{RM} = 0$, we observe the most consistent improvements over standard FA. Therefore, this was the setup we chose for most of the further experiments. Later we found the combination of standard leaving-one out with a restriction of the phrase lengths to produce comparable or slightly better results.

On the whole we can say that the application of leaving-one-out proves to be crucial for our models to produce competitive results. The other specific training setups we proposed do not seem to have a significant positive or negative impact on translation performance.

**Table 5.4.** Comparison of alignment quality measured in AER for GIZA++ and FA word alignments.

| | # alignment points | AER(%) |
|---|---:|---:|
| GIZA++ | 8844 | 21.0 |
| FA | 8865 | 24.8 |

### 5.3.3 Word alignment experiments

We described how we can re-estimate the word alignment by using the forced alignment procedure and how to use the re-estimated alignment to create a new phrase table in Section 4.1. For 505 sentences in the Europarl data set, we had human annotated word alignments available on which we could measure the quality of our re-estimated alignment by means of the AER metric (cf. Section 5.1.3). Table 5.4 summarizes the result. We can see that in this setup the word alignment produced by FA is significantly less accurate than the GIZA++ alignment, by 3.8% AER. Here, the alignment distance between the two is 18.2% AD and therefore considerably higher than for the IWSLT data (cf. Table 4.1). A human inspection of the generated alignments for the Europarl data does not reveal any noticeable patterns on how the re-estimated word alignments differ from the GIZA++ alignments, but confirms the higher quality of the latter.

To complement the experiments above we ran tests on how re-estimation of the word alignments affect translation quality when used for the heuristic from Section 2.3. These were done on the IWSLT data and the performance along the first three iterations is plotted in Figure 5.2. On both $DEV$ and $TEST$ set translation performance is declining with growing number of iterations, although the Bleu scores seem to be subject to some fluctuations.

### 5.3.4 Indicator feature experiments

Indicator features are binary features which indicate whether a phrase pair was seen in training (cf. Chapter 4.2). The corresponding experiments were conducted on the IWSLT data. Here, the training setup utilized standard leaving-one-out, standard reordering penalty and no skip and deletion or phrase extension models. We incorporated three indicator features into the system, one for the single best segmentation and one each for the 10-best and the 50-best list. The phrase translation probability estimates were kept identical to the baseline system. After the introduction of the indicator features, the scaling factors $\lambda_1^M$, including one factor

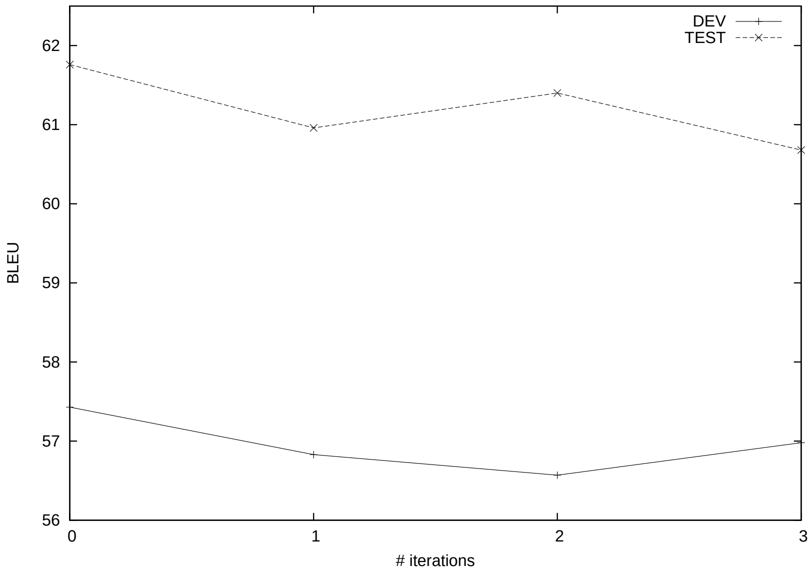

**Figure 5.2.** Translation performance with re-estimated word alignment plotted against number of iterations.

**Table 5.5.** Performance of the indicator features discussed in Section 4.2. Three indicator features were incorporated into the baseline system, one for the single best segmentation and one each for the 10-best and 50-best segmentations.

| | | Bleu | TER |
|---|---|---|---|
| **DEV** | baseline | 57.4 | 34.9 |
| | + indicator features | 57.3 | 35.2 |
| **TEST** | baseline | 61.8 | 29.9 |
| | + indicator features | 61.4 | 30.1 |

for each indicator feature, were retrained for Bleu score on *DEV* using MERT (cf. Section 2.2.5).

The results are shown in Table 5.5. We observe a slight degradation of performance compared to the baseline. In theory, this should not be possible. The system was left unchanged, except for adding new information with the indicator features. If this additional information can not contribute to producing good translations, this should be mirrored by the corresponding scaling factors produced by MERT. If the scaling factors for all three indicator features were set to $\lambda_{Ind} = 0$, the translations would be identical to the ones produced by the baseline system, resulting in the same Bleu score. However, as we have already mentioned in Chapter 2.2.5, MERT is not guaranteed to find a global optimum. If the dimensionality of the search space is increased, the number of local maxima is likely to increase as well, thus making MERT more prone to output suboptimal solutions. Furthermore, as we mentioned in Section 2.2.5, MERT is carried out with monotone phrasal alignment, thus training the parameters for a setup different from the one producing the final translations. The three additional features increase the potential for overfitting, so that the parameters given by MERT are fit for the development data with the particular restriction to monotone phrasal alignment, but will fail to produce good results in a different setup. These circumstances may be responsible for the observed degradation in Bleu score.

### 5.3.5 Generative phrase model experiments

#### IWSLT experiments

For evaluation of the proposed generative phrase models we ran two sets of experiments, one on IWSLT and one on the Europarl data. The main results for the Chinese-English IWSLT data set after the first training iteration are given in Table 5.6. Here, we experimented with both initialization methods we described in Section 3.6. For the heuristic initialization the training was setup with length-based leaving-one-out, deletion and skip model and no reordering penalty ($\lambda_{RM} = 0$). For the initialization with the PESA model, standard FA was used for training. Interpolation of phrase tables was done linearly. The results confirm the conclusions drawn by [DeNero & Gillick$^+$ 06]. On both $DEV$ and $TEST$ the generative phrase models are not competitive with the baseline. Interpolation of phrase tables yields improvements of up to 0.6 Bleu. While it is not clear which of the generative models performs better for interpolation, we find the interpolation of the weighted count model with the baseline to outperform the baseline system on both $DEV$ and $TEST$. When initializing with the PESA model we see some improvements over the translation performance of the initial phrase table, however the results are also below the heuristic baseline.

In Section 4.3 we observed that the phrase tables produced by the generative phrase models only contain a subset of the phrases in the initial phrase table. The results in the row labeled *baseline filtered* give us an idea at what effect the reduction in phrase table size has on the translation performance. Here, the phrase translation scores are identical to the baseline phrase table, but we only keep those phrase pairs that were seen in training. These are exactly the phrases which are available for translation by our generative phrase models. We can see that this leads to a

**Table 5.6.** Results for the generative phrase models on the IWSLT data. MERT was applied for BLEU score. The count model uses a 1000-best list. For the results given in row *baseline filtered* we filtered the baseline phrase table so that it only contained phrases seen in training.

| | initialization: | heuristic | | PESA | |
|---|---|---|---|---|---|
| | | BLEU | TER | BLEU | TER |
| **DEV** | baseline | 57.4 | 34.9 | 54.7 | 36.4 |
| | baseline filtered | 56.9 | 35.5 | | |
| | count model | 56.7 | 35.4 | 55.7 | 36.5 |
| | weighted counts | 55.6 | 36.6 | 55.7 | 35.7 |
| interpolation | baseline + count model | 58.0 | 34.9 | 57.3 | 35.1 |
| | baseline + weighted counts | 57.5 | 34.9 | 57.3 | 35.3 |
| **TEST** | baseline | 61.8 | 29.9 | 59.7 | 32.1 |
| | baseline filtered | 61.1 | 30.5 | | |
| | count model | 61.7 | 30.5 | 61.8 | 30.5 |
| | weighted counts | 59.4 | 32.5 | 60.6 | 30.6 |
| interpolation | baseline + count model | 61.7 | 30.8 | 61.2 | 30.5 |
| | baseline + weighted counts | 62.4 | 29.7 | 61.4 | 30.1 |
| **TRAIN** | baseline | 71.8 | 19.6 | | |
| | baseline with l1o | 44.3 | 39.2 | | |
| | count model | 61.7 | 26.4 | | |
| | weighted counts | 61.0 | 27.0 | | |

reduction in BLEU score compared to the full phrase table. From this we conclude that in this setup the smaller set of phrases available to the generative phrase models has disadvantages at decoding time. Averaged over development and test set the count model shows roughly the same performance as the filtered baseline phrase table. This indicates that the slightly inferior translation quality of the count model mainly results from the reduced number of phrases.

To get further insight into what phrases are being used for the different translations, we ran forced alignment on *DEV* as well. We can assume that the phrases appearing there are useful to produce good translations. The phrase pairs used for FA on *DEV* were compared with the ones used for the baseline translation and for the count model translation. We can see in Table 5.7 that 57% of the distinct phrase pairs used in the count model translation also appeared in the baseline translation. Furthermore, 74% of these phrase pairs are seen in forced alignment on *DEV*, compared to 72% of the distinct phrases used in the baseline translation. We

**Table 5.7.** Statistics on the number of distinct phrase pairs used for translation for the baseline, the count model and FA on $DEV$ (for all of the 16 reference translations) of the IWSLT data. The fields contain the number of distinct phrase pairs that are being used for both row and column label. The diagonal contains the respective total number of distinct phrase pairs.

| | baseline | count model | FA on $DEV$ |
|---|---|---|---|
| baseline | **1330** | 801 | 959 |
| count model | | **1400** | 1039 |
| FA on $DEV$ | | | **28132** |

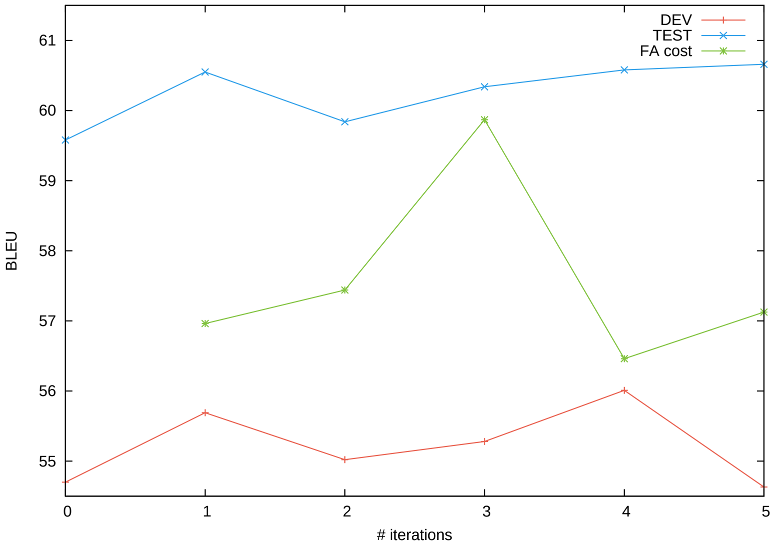

**Figure 5.3.** Performance of the weighted count model through several iterations on the IWSLT data. The performance on $DEV$ and $TEST$ is given in Bleu. The green line plots the corresponding model costs of FA on the training data. The model costs for iteration 0 are based on a different model and are therefore not comparable. Initialization was done with the PESA model (cf. Chapter 3.6.2).

conclude that the choice of phrases used for translation is not responsible for the underperformance of our generative phrase models.

### Iteration of FA

To get an idea of the performance of our generative phrase models over the course of several iterations we take a look at Figure 5.3. The performance of the weighted

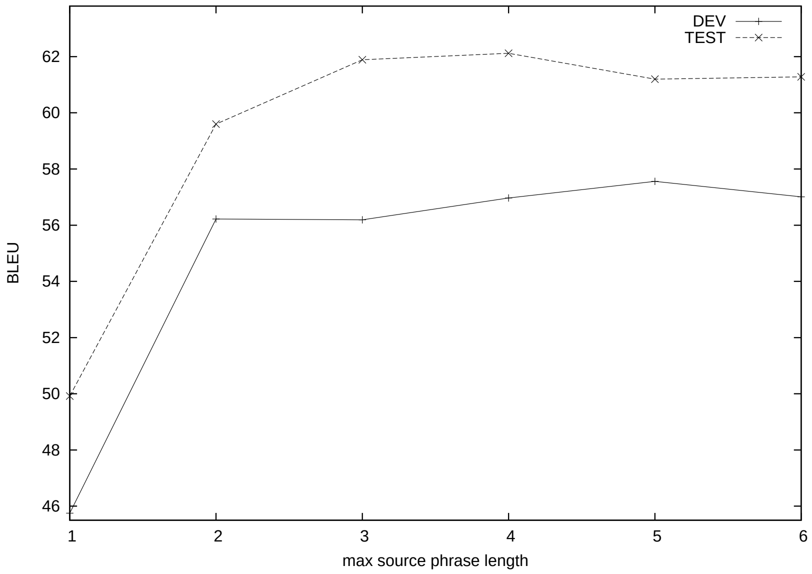

**Figure 5.4.** Translation performance of the count model plotted against maximum phrase length on the IWSLT data.

count model on $DEV$ and $TEST$ is plotted for the first five iterations. The PESA model served for initialization of the first iteration. The subsequent iterations were then initialized with the phrase table estimated by the weighted count model in the preceding iteration. Additionally, Figure 5.3 contains one curve which shows the total cost for the segmentations on $TRAIN$ in FA. As initialization was done with a different model, the cost for the first training iteration does not bear a correspondence to the following iterations and is therefore left out. The scaling factors $\lambda_1^M$ were trained with MERT on $DEV$ for the PESA model and then kept fixed for the training iterations. For the translations, however, $\lambda_1^M$ was re-estimated for the new phrase table after each iteration.

We can see in Figure 5.3 that the weighted count model leads to an improvement over the initialization after the first iteration. For the subsequent iterations we see a decline followed by a second increase of Bleu score reaching the maximum in iteration four with 56.0 Bleu on $DEV$ and 60.6 Bleu on $TEST$. This maximum coincides with the minimum model costs in FA. Other than that there does not seem to be any connection between the model costs in training and the corresponding translation scores. On $DEV$ we observe another drop in translation quality after the fifth iteration. On the whole there is no indication that continuing the iterative training procedure will lead to more significant improvements.

**Table 5.8.** Results for the generative phrase models on the Europarl data. MERT was applied for Bleu score, initialization was done with the heuristic from Section 2.3. The count model uses a 1000-best list. For the results given in row *baseline filtered* we filtered the baseline phrase table so that it only contained phrases seen in training.

| | iteration: | 1 | | 2 | |
|---|---|---|---|---|---|
| | | BLEU | TER | BLEU | TER |
| **DEV** | baseline | 26.2 | 59.4 | | |
| | baseline filtered | 26.9 | 60.2 | | |
| | count model | 27.0 | 59.7 | 26.9 | 59.8 |
| | weighted counts | 26.9 | 60.0 | | |
| interpolation | baseline + count model | 27.2 | 59.6 | | |
| | baseline + weighted counts | 27.0 | 59.7 | | |
| **TEST** | baseline | 27.1 | 58.9 | | |
| | baseline filtered | 27.7 | 59.8 | | |
| | count model | 27.7 | 59.6 | 27.6 | 59.7 |
| | weighted counts | 27.6 | 59.2 | | |
| interpolation | baseline + count model | 28.0 | 59.4 | | |
| | baseline + weighted counts | 27.8 | 59.4 | | |

### Phrase length restriction

In further experiments we examined the impact of phrase length on the results. For this we ran FA with different restrictions on phrase length. We tested the setups $f_{max} = 1, \ldots, 6$ with $e_{max} = 2f_{max}$. The performance of the count model is plotted in Figure 5.4. We can see a significant increase in translation quality from $f_{max} = 1$ to $f_{max} = 2$. Further increasing the maximum phrase length yields moderate improvements until the maximum is reached at $f_{max} = 5$ for $DEV$ and $f_{max} = 4$ for $TEST$. Also, we tried to interpolate the phrase tables produced with different phrase length restrictions, but did not find this to lead to any improvements.

### Europarl experiments

The results for the German-English Europarl data are shown in Table 5.8. For this set of experiments the training was setup with standard leaving-one-out and no reordering penalty ($\lambda_{RM} = 0$). Here, phrase table interpolation of the weighted count model with the baseline phrase table was done linearly. For the combination of the count model with the baseline we tested both linear and log-linear interpolation.

**Table 5.9.** Comparison of the performance of the linear and log-linear phrase table interpolation on Europarl. The phrase tables used for interpolation were produced by the heuristic (cf. Section 2.3) and the count model (cf. 4.3.1).

| | | BLEU | TER |
|---|---|---|---|
| **DEV** | count model | 27.0 | 59.7 |
| interpolation | + baseline (linear) | 27.0 | 59.7 |
| | + baseline (log-linear) | 27.2 | 59.6 |
| **TEST** | count model | 27.7 | 59.6 |
| | + baseline (linear) | 27.8 | 59.4 |
| | + baseline (log-linear) | 28.0 | 59.4 |

In contrast to the results on IWSLT, we can see that the generative phrase models consistently outperform the heuristic baseline by 0.5 to 0.8 BLEU. The count model has minor advantages over the weighted count model. When interpolating the count model in a log-linear way with the heuristic phrase table, we reach a performance gain of 1.0 BLEU on $DEV$ and 0.9 BLEU on $TEST$. The interpolation of the weighted count model with the heuristic phrase table also shows consistent improvements over the baseline, confirming the results on IWSLT. A second iteration with the count model shows no significant change in translation performance. A possible reason is that leaving-one-out was not implemented for application on the generative phrase models due to time constraints, which may be done in future work.

Table 5.9 shows a comparison of linear and log-linear interpolation of the count model with the baseline phrase table. Here, the linear interpolation leads to a slight improvement over the pure count model. The log-linear interpolation works slightly better, yielding a moderate gain of 0.2 BLEU on $DEV$ and 0.3 BLEU on $TEST$ over the pure count model. The progression of translation performance over different interpolation weights was already shown in Figures 4.6 and 4.7.

Unlike on the IWSLT data, the reduction of the phrase table size from the heuristic model to our generative models seems to be preferable for translation on the Europarl data. The results produced by the heuristic phrase table filtered to contain only the phrases seen in training are nearly as good as the ones given by the count model. The critical point for producing good translations, therefore, seems to be the choice of phrases made available at decoding time. However, while on the IWSLT data the smaller phrase table provided by the generative phrase models has disadvantages, for the Europarl data it yields better results than the one produced by the heuristic.

### Comparison of count model and weighted count model

When comparing the two different generative phrase models, we find that in nearly all setups the simple count model produces better translations than the more sophisticated weighted count model. We hypothesize three possible reasons for this observation:

- In Section 4.3.2 we stated that most of the probability mass is concentrated on the few best scoring segmentations. This supports the assumption that the estimated distribution $p_{FA}(s_1^K|e_1^I, f_1^J)$ over the segmentations $s_1^K$ may assign too high probabilities to the top scoring segmentations. In this case we might hope to improve translation quality by applying smoothing techniques.
- The estimated probability distribution $p_{FA}(s_1^K|e_1^I, f_1^J)$ is of poor quality. If this is the case, a refinement of the FA procedure with the goal of producing a better estimate of the real distribution might lead to better results.
- The weighted count model is more deterministic than the count model due to the fact that the top scoring few segmentations are assigned most of the weight. The greater ambiguity provided by the count model may be preferable to this peaked distribution at decoding time. In this case, the count model is better suited to meet the ambiguities of natural languages. We might hope to achieve further improvements by introducing some means to encourage FA to produce a greater variety of segmentations.

In future work, some additional research needs to be done to determine whether one of the above hypotheses is true so that we can take the appropriate measures for further improvement.

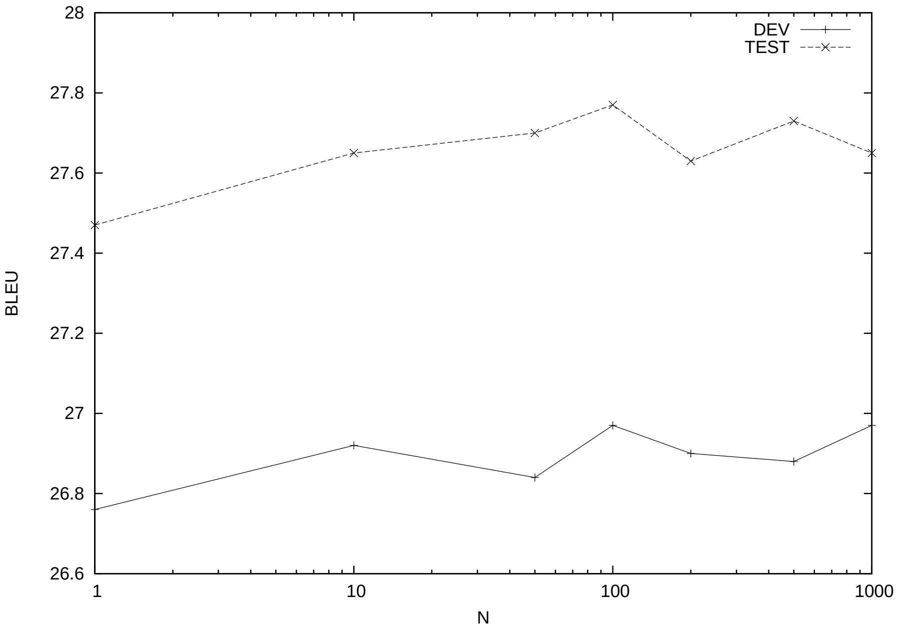

**Figure 5.5.** Performance in Bleu of the count model plotted against the number $N$ of $N$-best segmentations on a logarithmic scale for the Europarl data set.

### Size of $N$-best list

We ran further experiments which examine the impact of the size $N$ of the $N$-best list for the count model. We have observed in Figure 4.2 that the size of the phrase table is subject to significant increase with growing $N$. For $N = 10$ it has 7.7M entries and reaches a size of 14.8M for $N = 1000$. However, Figure 5.5 shows that this increase in ambiguity has very little influence on translation performance on the Europarl data. We can see that using an $N$-best list of at least size 10 is slightly superior to only using the single-best segmentation. It leads to an improvement of roughly 0.2 Bleu. However, we can not deduce a clear connection between a further increase and translation performance from the graph, although setting $N = 100$ seems to yield the best results here.

### Size of training corpus

It is clear that the generative phrase models perform considerably better on the Europarl data than on the IWSLT data. A prominent difference between the two data sets is the size of the training corpus. To determine whether this is the reason

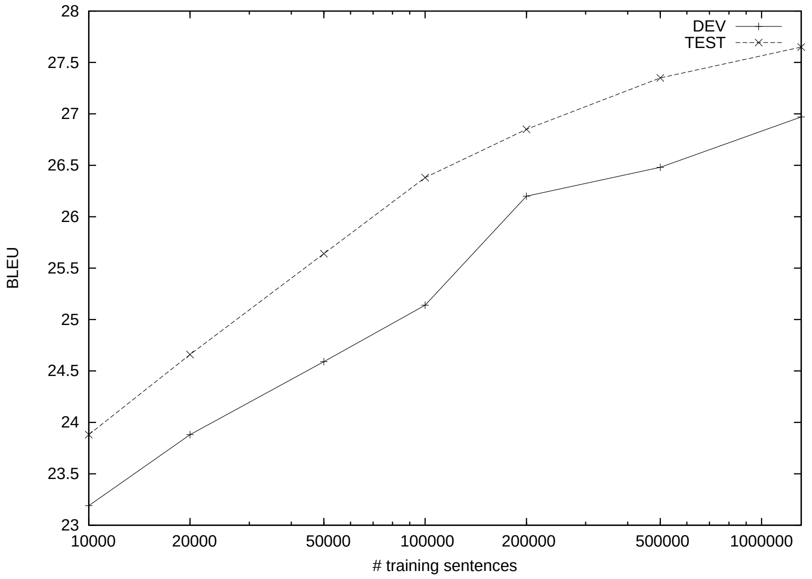

**Figure 5.6.** Performance in Bleu of the count model for $N = 1000$ plotted against the number of training sentences on a logarithmic scale for the Europarl data set.

for the difference in performance we investigate the connection between the size of the training corpus and translation performance measured in Bleu in Figure 5.6. To get a realistic view of these correlations all steps would have to be taken on the constrained training data, including the word alignments and the heuristic estimation of the phrase translation probabilities. Unfortunately, due to time constraints we had to resort to the following simpler method. We initialized with the heuristic and kept the initial phrase table fixed, meaning that the initialization was estimated on the full training data in all cases. This of course can only give us a very rough estimate of the interdependencies between training corpus size and performance. We can see clearly in Figure 5.6 that a greater amount of data in training leads to better results. The absolute difference in Bleu score between using 10 000 sentences and the full data set of 1.3M sentences to train the count model is roughly 3.8 points. This is surprisingly little with regard to the fact that the number of phrases available differs by a factor of 32 (cf. Figure 5.7). These results indicate that the training corpus size is not the reason for the difference in performance of our models on the two data sets, although further experiments will have to be conducted for confirmation.

**Figure 5.7.** Phrase table size of the count model for an N-best list size of $N = 1000$ plotted against the number of training sentences on a logarithmic scale for the Europarl data set.

### Example sentences

To conclude this chapter we will have a look at two example sentences from the Europarl data, for which the count model improved the translation over the baseline system. Figure 5.8 shows a sentence from the development data set $DEV$. The baseline translation produced with the heuristic phrase translation model described in Chapter 2.3 fails to carry over the two content words '*macht*' and '*angepasst*' into the English translation. This is remedied by the count model translation. The improvements can be traced back to the phrase translation model. The heuristic estimates the translation probability of the incorrect phrase pairs ('*gute Fortschritte macht*','*good progress*') and ('*angepasst hat , um*','*in order*') to

$$p_H('in\ order'|'gute\ Fortschritte\ macht') = 1 \text{ and}$$
$$p_H('good\ progress'|'angepasst\ hat\ ,\ um') = 0.5.$$

The count model phrase table contains neither of those phrase pairs and can therefore not use them for translation. This also illustrates why filtering the original phrase table to contain only phrase pairs seen in training has nearly the same effect as using the count model.

SOURCE:
Wenn die Tschechische Republik auch weiterhin gute Fortschritte macht und die Europäische Union ihre institutionellen Strukturen und Politiken angepasst hat , um die Erweiterung bewältigen zu können [...]

REFERENCE:
If the Czech Republic continues to make progress and the European Union adjusts its institutional structure and policies in order to cope with enlargement [...]

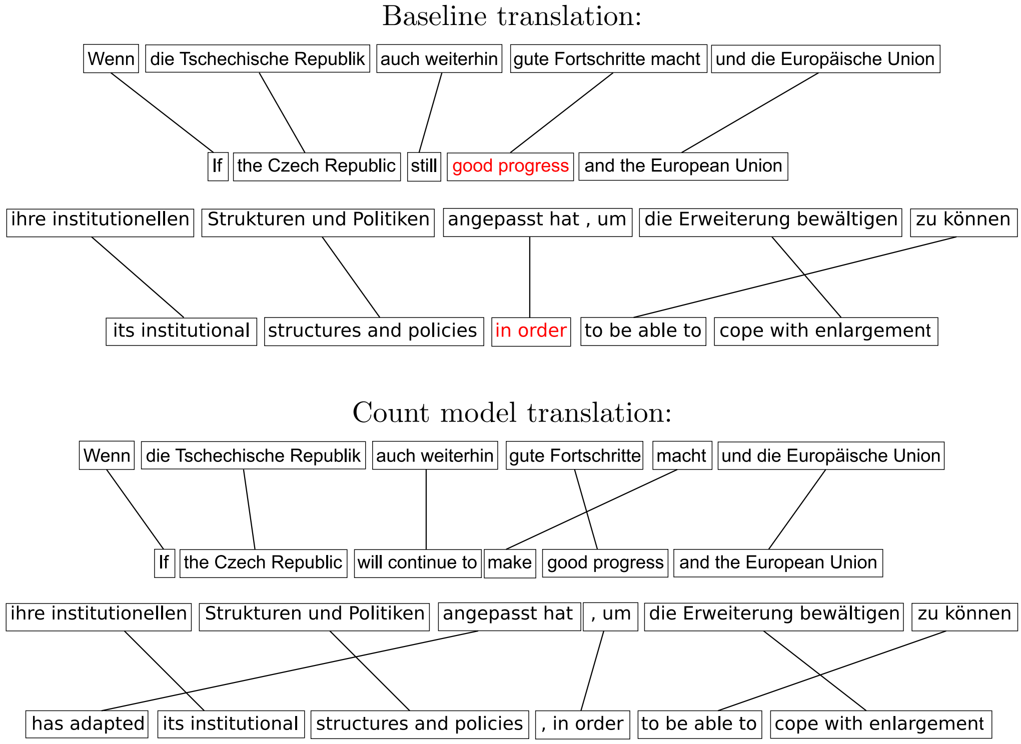

**Figure 5.8.** Example from the Europarl development data set and the corresponding translations produced with the heuristic from Section 2.3 and with the count model. Incorrect phrase translations are marked red.

SOURCE: Dies ist ein zentraler und wichtiger Punkt bei der vorbeugenden Arbeit .

REFERENCE: This is a crucial and important point in preventive work .

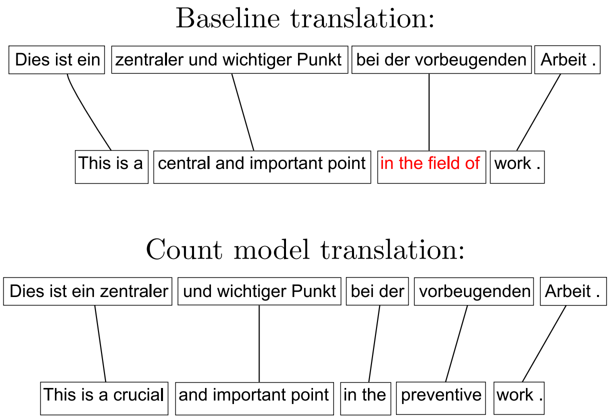

**Figure 5.9.** Example from the Europarl test data set and the corresponding translations produced with the heuristic from Section 2.3 and with the count model. Incorrect phrase translations are marked red.

The example in Figure 5.9 is taken from $TEST$. Here, the baseline system fails to translate '*vorbeugenden*' correctly. Similar to the example given in Figure 5.8 the heuristic strongly overestimates the corresponding phrase translation probability, while the phrase table produced by the count model does not contain the phrase pair in question.

**Table 5.10.** Performance of the combined segmentation and weighted counts model discussed in Section 4.3.3.

| | | IWSLT | | Europarl | |
|---|---|---|---|---|---|
| | | Bleu | TER | Bleu | TER |
| **DEV** | weighted counts | 56.4 | 36.0 | 26.9 | 60.0 |
| | + seg. model | 55.5 | 36.0 | 26.7 | 59.8 |
| **TEST** | weighted counts | 60.2 | 31.7 | 27.7 | 59.6 |
| | + seg. model | 60.0 | 31.2 | 27.6 | 59.2 |

### 5.3.6 Phrase prior experiments

In Section 4.3.3 we proposed two different ways of incorporating a segmentation model into our system. For the following experiments, the training setup will be identical to Section 5.3.5 for the Europarl data. On the IWSLT data we chose to use standard leaving-one-out and no reordering penalty. Skip and deletion models and phrase extensions were not applied.

We will first look at the combined segmentation and translation model $p_{W_{SEG}}(\tilde{f}|\tilde{e})$ based on the weighted count model. The results are shown in Table 5.10 for both the IWSLT and Europarl data. On both data sets and both $DEV$ and $TEST$ we can see a decrease in Bleu score compared to the plain weighted count model.

We have mentioned in Section 4.3.3 that the combined segmentation and translation model is a special case of incorporating the segmentation model separately. We can examine the performance of the separate segmentation model in Table 5.11. Again, we observe a degradation of translation performance for both the count and the weighted count model on the IWSLT data. For the reasons we have described in Section 5.3.4 for the indicator features, this should in theory be impossible, as the only change to the system is the addition of new information. However, the instability of MERT and the overfitting effects discussed in Section 5.3.4 can explain this phenomenon. On the Europarl data, we observe that the addition of the segmentation model yields a small increase in Bleu score on both data sets and for both generative phrase models.

In additional experiments we tested adding the two different separate segmentation models to the baseline system on the IWSLT data. The results in Table 5.12 reveal similar effects to the ones we have described above. Like with the generative phrase models, we observe that the addition of the segmentation model leads to a slight degradation of translation quality. Only on $DEV$ the segmentation model based on the count model described in Section 4.3.1 shows an increase of 0.3 Bleu. However,

**Table 5.11.** Performance of the separate segmentation model discussed in Section 4.3.3. The count model uses a 1000-best list.

| | | **IWSLT** | | **Europarl** | |
|---|---|---|---|---|---|
| | | Bleu | TER | Bleu | TER |
| **DEV** | count model | 56.9 | 34.9 | 27.0 | 59.7 |
| | + seg. model | 56.8 | 35.1 | 27.2 | 59.9 |
| | weighted counts | 56.4 | 36.0 | 26.9 | 60.0 |
| | + seg. model | 55.4 | 37.0 | 27.0 | 59.9 |
| **TEST** | count model | 61.1 | 30.9 | 27.7 | 59.6 |
| | + seg. model | 61.0 | 30.7 | 27.8 | 59.5 |
| | weighted counts | 60.2 | 31.7 | 27.7 | 59.6 |
| | + seg. model | 59.7 | 32.7 | 27.8 | 59.5 |

**Table 5.12.** Performance of the separate segmentation model discussed in Section 4.3.3 as addition to the baseline system on the IWSLT data set. The count model uses a 1000-best list.

| | | Bleu | TER |
|---|---|---|---|
| **DEV** | baseline | 57.4 | 34.9 |
| | + count seg. model | 57.7 | 35.2 |
| | + weighted count seg. model | 57.3 | 35.4 |
| **TEST** | baseline | 61.8 | 29.9 |
| | + count seg. model | 60.2 | 31.3 |
| | + weighted count seg. model | 61.4 | 30.6 |

this is put into perspective by the same system clearly underperforming the baseline on $TEST$ by 1.6 Bleu.

# 6 Conclusion

In this chapter, we will give a summary of the work presented in this thesis and discuss possible directions for further research on the topic.

## 6.1 Summary

In this work we have introduced novel ways of modeling phrase translation probabilities for SMT and an appropriate framework to train them. We have developed a machine translation method which consistently performs better or equal to the baseline system by combining the current state of the art with a novel model.

The model training implements forced alignment (FA), for which we constrain the translation decoder to a given target sentence. We have shown that in order to prevent overfitting and get good training results, application of the leaving-one-out method is a crucial point.

The information gained in training can be used to train generative phrase models, re-estimate word-alignment or be incorporated into the state-of-the-art system. To evaluate the different approaches, experiments were conducted on two different data corpora, the German-English part of the medium-sized Europarl corpus and the Chinese-English part of the IWSLT data set, which is a small limited domain task. We found that employment of FA to re-estimate word alignments as well as the indicator features we proposed do not lead to improvements in translation quality.

Our two proposed generative phrase models outperform the baseline system on the Europarl corpus by up to 0.8 Bleu, however do not yield competitive results on the smaller IWSLT data set. We have determined that the key point for a competitive performance is the choice of phrase pairs which are available at decoding time, rather than the translation probability estimates assigned to them.

Also, we described two different ways of incorporating a segmentation model into the system, either by combining it directly with the phrase translation scores, or by adding it as a separate model. Our results for both methods show no significant effect on translation quality.

All results mentioned above were produced after a single iteration of training. Reiterating the training process several times so far has not lead to any improvements. However, leaving-one-out was not applied after the first iteration. As we have found leaving-one-out to be important for good training results, this is planned to be tested in future experiments.

The best results on both data sets resulted from interpolating the phrase tables of our generative phrase models with the baseline phrase table. These results show improvements of up to 1.0 Bleu on the Europarl data set and 0.6 Bleu on IWSLT. With the interpolation of the weighted count model with the baseline phrase table we have developed a method which consistently outperforms the state-of-the-art baseline system.

The software we implemented in the course of this work includes the leaving-one-out method for forced alignment, the word omission model and the estimation of phrase translation probabilities based on the novel models.

## 6.2 Future Work

There are a number of open questions that our work has not been able to answer conclusively. Further research could be conducted on the following topics.

We have found that our generative models work well on the Europarl data, but are not competitive on the IWSLT data. The experiments that were run in the course of this thesis do not provide sufficient evidence to determine the reason. The two data sets differ in several characteristics - among others corpus size, language pair, number of reference translations and domain - whose influence on the methods should be investigated separately.

Further, it has been established that the choice of phrases available at translation time has a greater influence on translation quality than the method of estimating translation probabilities. The training procedure presented in this work provides a qualitative selection process for these phrases. However, it may be possible to do this selection by simpler methods.

Incorporating the estimated probability distribution $p_{FA}(s_1^K|e_1^I, f_1^J)$ over the phrase segmentations into the model did not perform as well as using a simple count model. This may indicate, that the definition of $p_{FA}(s_1^K|e_1^I, f_1^J)$ is suboptimal and should be reassessed.

A possible refinement for the training procedure would be the introduction of lexicalized skips and deletions. Rather than assigning a constant penalty for omitting a word, we would allow the penalty to be dependent on the specific word or phrase. Thus a more fine-grained control of word omissions could be possible.

Finally, reiterating the training procedure is a promising direction for future research. Implementation of leaving-one-out for further iterations should be tested, as we have shown its effectiveness on the first iteration. Furthermore, if we always choose the best performing system, here the interpolated phrase table, to initialize the next iteration, we may be able to further improve translation quality.

# List of Figures

# List of Tables

# Bibliography

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

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

[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 & Pietra$^+$ 93] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, R. L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–312, June 1993.

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

[Callison-Burch & Osborne$^+$ 06] C. Callison-Burch, M. Osborne, P. Koehn. Re-evaluating the Role of BLEU in Machine Translation Research. In *11th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 249–256, Trento, Italy, April 2006.

[Dempster & Laird$^+$ 77] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Vol. 39, No. 1, pp. 1–38, 1977.

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

[Fordyce 07] C. S. Fordyce. Overview of the IWSLT 2007 Evaluation Campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Trento, Italy, October 2007.

[Ganitkevitch 08] J. Ganitkevitch. Lexical Triggers for Statistical Machine Translation. Diploma Thesis, October 2008.

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

[Kneser & Ney 95] R. Kneser, H. Ney. Improved Backing-Off for M-gram Language Modelling. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 181–184, Detroit, MI, May 1995.

[Knight 99] K. Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, Vol. 25, No. 4, pp. 607–615, 1999.

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

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

[Mauser & Zens$^+$ 06] A. Mauser, R. Zens, E. Matusov, S. Hasan, H. Ney. The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation. In *International Workshop on Spoken Language Translation (IWSLT)*, pp. 103–110, Kyoto, Japan, November 2006.

[Mauser 05] A. Mauser. Improved Word Alignment and Extraction for Statistical Machine Translation. Diploma Thesis, September 2005.

[Nelder & Mead 65] J. Nelder, R. Mead. A Simplex Method for Function Minimization. *The Computer Journal)*, Vol. 7, pp. 308–313, 1965.

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

[Och & Ney 00] 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, October 2000.

[Och & Ney 02] F. J. Och, H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, PA, July 2002.

[Och & Ney 03] F. J. Och, H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, Vol. 29, No. 1, March 2003.

[Och & Ney 04] F. J. Och, H. Ney. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417–450, December 2004.

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

[Papineni & Roukos$^+$ 02] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, Philadelphia, PA, July 2002.

[Press & Teukolsky$^+$ 02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK, 2002.

[Shen & Delaney$^+$ 08] W. Shen, B. Delaney, T. Anderson, R. Slyh. The MIT-LL/AFRL IWSLT-2008 MT System. In *Proceedings of IWSLT 2008*, pp. 69–76, Hawaii, U.S.A., October 2008.

[Snover & Dorr$^+$ 06] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, August 2006.

[Steinbiss & Tran$^+$ 94] V. Steinbiss, B. Tran, H. Ney. Improvements in Beam Search. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP'94)*, pp. 2143–2146, September 1994.

[Stolcke 02] A. Stolcke. SRILM – An Extensible Language Modeling Toolkit. In *Proc. Int. Conf. on Spoken Language Processing*, Vol. 2, pp. 901–904, Denver, CO, 2002.

[Takezawa & Sumita$^+$ 02] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, S. Yamamoto. Toward a Broad-Coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World. In *Proc. of the Third Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 147–152, Las Palmas, Spain, May 2002.

[Tillmann & Ney 00] C. Tillmann, H. Ney. Word Re-ordering and DP-based Search in Statistical Machine Translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Vogel 05] S. Vogel. PESA: Phrase Pair Extraction as Sentence Splitting. In *Proceedings of MTSummit X*, Phuket, Thailand, September 2005.

[Zens & Och$^+$ 02] R. Zens, F. J. Och, H. Ney. Phrase-Based Statistical Machine Translation. In M. Jarke, J. Koehler, G. Lakemeyer, editors, *25th German Conf.*

*on Artificial Intelligence (KI2002)*, Vol. 2479 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 18–32, Aachen, Germany, September 2002. Springer Verlag.

[Zens 08] R. Zens. *Phrase-based Statistical Machine Translation: Models, Search, Training.* Ph.D. thesis, Computer Science Department, RWTH Aachen – University of Technology, Germany, February 2008.