DIPLOMARBEIT

Kontextabhängige statistische Übersetzungsmodelle

Autor: Richard Zens
Matr.-Nr.: 213216
Betreuer: Dipl.-Inform. Franz Josef Och

Gutachter: Prof. Dr-Ing. Hermann Ney
Prof. Dr. Juraj Hromkovič

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik VI
Computer Science Department
RWTH Aachen - University of Technology
Germany

Hiermit versichere ich, daß ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 28. Mai 2002

(Richard Zens)

Abstract

In this diploma thesis context-based translation models for statistical machine translation are presented. The first one is a phrase-based translation model. It uses bilingual phrases as smallest translation units instead of single words. The basic idea of this approach is to segment the source language sentence into phrases, then to translate each phrase and finally combine these phrase translations to the sentence translation.

In the second approach, the translation model uses a context-dependant lexicon model. Typically, in a single word based approach, the translation of a source language word depends only on the word itself. In the model presented here, the translation also depends on the already translated source words and their translations. So this model is based on some kind of a two-dimensional n-gram, namely an n-gram on the alignment path.

For the phrase-based translation model different variants of the search are described. A monotone one that is very efficient. There are two non-monotone search variants described: one takes the possible reorderings from 'outside', from a so-called reordering graph. The other one is based on pattern pairs. These pattern pairs model explicitly possible reorderings of whole phrases.

The results are presented on the VERBMOBIL corpus (German-English), the FUB corpus (Italian-English) and the El Periodico corpus (Spanish-Catalan). It will be shown that the context-based translation models outperform the single-word based models.

Kurzfassung

In dieser Diplomarbeit werden kontextabhängige Übersetzungsmodelle für die statistische maschinelle Übersetzung präsentiert. Das erste ist ein phrasenbasiertes Übersetzungsmodell. Es verwendet bilinguale Phrasen als kleinste Übersetzungseinheiten anstelle von einzelnen Wörtern. Die zugrundeliegende Idee bei diesem Ansatz ist, den Quellsatz in Phrasen zu segmentieren, dann jede Phrase zu übersetzen und schließlich aus diesen Phrasenübersetzungen die Übersetzung des Satzes zusammenzusetzen.

In dem zweiten Ansatz wird ein kontextabhängiges Lexikonmodell in das Übersetzungsmodell eingebracht. Typischerweise hängt bei einzelwortbasierten Ansätzen die Übersetzung eines Wortes nur von dem Wort selbst ab. In dem Modell hier hängt sie auch von den zuvor übersetzten Wörtern und deren Übersetzungen ab. Dieses Modell basiert auf einer Art zweidimensionalem n-Gramm, nämlich einem n-Gramm auf dem Alignmentpfad.

Für die phrasenbasierte Übersetzung werden verschiedene Suchvarianten beschrieben. Eine monotone Suche welche sehr effizient ist. Außerdem werden noch zwei nicht-monotone Varianten beschrieben: die eine Variante nimmt die möglichen Umordnung von 'außen', und zwar von einem sogenannten Umordnungsgraphen. Die andere Variante basiert auf Pattern-Pairs. Diese Pattern-Pairs modellieren explizit Umstellungen ganzer Phrasen.

Die Resultate werden für das Verbmobil Korpus (Deutsch-Englisch), das FUB Korpus (Italienisch-Englisch) und das El Periodico Korpus (Spanisch-Katalanisch) präsentiert.

Es wird gezeigt werden, dass die kontextabhängigen Modelle den einzelwortbasierten Modellen überlegen sind.

Inhaltsverzeichnis

1 Einführung **1**

2 Grundlagen **3**

2.1 Bayes'sche Entscheidungsregel 3
2.2 Direkter Ansatz 4
2.3 Sprachmodell 6
2.4 Übersetzungsmodell 7
2.5 Einzelwortbasiertes Lexikonmodell 9
2.6 Einzelwortbasierte Alignmentmodelle 10
2.6.1 IBM1-, IBM2- und HMM-Modell 10
2.6.2 IBM3-Modell 11
2.7 Alignment Templates 12

3 Modelle **15**

3.1 Phrasenbasierte Übersetzung 15
3.1.1 Motivation 15
3.1.2 Bilinguale Phrasen 17
3.1.3 Phrasenbasiertes Übersetzungsmodell 19
3.1.4 Varianten 21
3.2 Pattern-Pairs 25
3.2.1 Motivation 25
3.2.2 Definition und Extraktion 26
3.2.3 Pattern-Pair Übersetzungsmodell 28
3.3 Zweidimensionale n-Gramme 31
3.3.1 Motivation 31
3.3.2 Definition 31
3.3.3 Übersetzungsmodell mit zweidimensionalen n-Grammen 33
3.3.4 Kombination mit der phrasenbasierten Übersetzung . . 34

4 Suche **35**
4.1 Monotone Suche 36
4.2 Nicht-Monotone Suche 38
4.2.1 Umordnungsgraph 39
4.2.2 Pattern-Pairs 42
4.3 Pruning 44

5 Ergebnisse **47**
5.1 Training 47
5.2 Testkriterien 48
5.3 Verbmobil 50
5.3.1 Korpus 50
5.3.2 Monotone phrasenbasierte Übersetzung 52
5.3.3 Nicht-Monotone phrasenbasierte Übersetzung 56
5.3.4 Vergleich mit anderen statistischen Ansätzen 60
5.4 FUB 61
5.5 El Periodico 63
5.6 Effizienz 65

6 Zusammenfassung und Ausblick **67**

A Modellskalierungsfaktoren **69**

Literaturverzeichnis **73**

Abbildungsverzeichnis

2.1 Architektur des Übersetzungsansatzes basierend auf der Bayes'schen Entscheidungsregel. 4
2.2 Wortaligniertes Satzpaar. 8

3.1 Beispiel für einzelwortbasierte Übersetzung (SWB) und phrasenbasierte Übersetzung (PBT). 15
3.2 Phrasenbasierte Übersetzung des Satzes '*abends würde ich gerne entspannen und vielleicht in die Sauna gehen .*' 16
3.3 Wortaligniertes Satzpaar. 17
3.4 Extrahierte bilinguale Phrasen. 17
3.5 Algorithmus `extract-BP` zur Extraktion bilingualer Phrasen. . 18
3.6 Beispiel für eine Übersetzung mit Pattern-Pairs. 25
3.7 Beispiele für Pattern-Pairs. 26
3.8 Beispiel für eine Übersetzung mit Pattern-Pairs. 27
3.9 Beispiele für die Erzeugung von Pattern-Pairs. 28

4.1 Algorithmus `MonSearch` zur monotonen Suche. 37
4.2 Umordnungsgraph zu "*welches Cafe ?*". 40
4.3 Beispiel für die Anwendung von Pattern-Pairs 44

5.1 VERBMOBIL: Vergleich der mWER für Maximum Approximation und Summenkriterium 54
5.2 VERBMOBIL: Auswirkung des Lexikonskalierungsfaktors 57

Tabellenverzeichnis

5.1 VERBMOBIL Statistik der Korpora 51
5.2 VERBMOBIL: Statistik der bilingualen Phrasen 51
5.3 VERBMOBIL: Statistik der zweidimensionalen n-Gramme . . . 52
5.4 VERBMOBIL: Statistik der Pattern-Pairs 53
5.5 VERBMOBIL: Baseline Varianten für monotone phrasenbasierte Übersetzung. 53
5.6 VERBMOBIL: Auswirkungen von Wortstrafe und Phrasenstrafe 55
5.7 VERBMOBIL: Varianten monotoner phrasenbasierter Übersetzung. 55
5.8 VERBMOBIL: Lexikonvarianten und zweidimensionale n-Gramme in Kombination mit monotoner phrasenbasierter Übersetzung. 57
5.9 VERBMOBIL: Größe der Umordnungsgraphen 58
5.10 VERBMOBIL: Vergleich monotoner und nicht-monotoner phrasenbasierter Übersetzung. 58
5.11 VERBMOBIL: Varianten nicht-monotoner phrasenbasierter Übersetzung. 59
5.12 VERBMOBIL: Pattern-Pair Varianten 59
5.13 VERBMOBIL: Pattern-Pair Varianten (Level 3). 60
5.14 VERBMOBIL: Vergleich mit anderen Systemen. 60
5.15 VERBMOBIL: Übersetzungsbeispiele. 61
5.16 FUB: Statistik der Korpora 62
5.17 FUB: Statistik der bilingualen Phrasen 62
5.18 FUB: Ergebnisse 63
5.19 El Periodico: Statistik der Korpora 64
5.20 El Periodico Ergebnisse 64
5.21 VERBMOBIL: Auswirkungen des Prunings 65
5.22 VERBMOBIL: Übersetzungszeiten pro Satz der verschiedenen Suchverfahren 66

A.1 VERBMOBIL: Skalierungsfaktoren bei Übersetzung auf dem Umordnungsgraphen 69

A.2 VERBMOBIL: Skalierungsfaktoren bei monotoner Übersetzung 70
A.3 VERBMOBIL: Skalierungsfaktoren bei Übersetzung mit Pattern-Pairs 70
A.4 FUB: Skalierungsfaktoren bei monotoner Übersetzung 71
A.5 El Periodico: Skalierungsfaktoren bei monotoner Übersetzung . 71

Kapitel 1

Einführung

Das Ziel der maschinellen Übersetzung ist es, die Bedeutung eines Satz der Quellsprache in einen Satz der Zielsprache zu übertragen. Die Verfahren der maschinellen Übersetzung besitzen einen Automatisierungsgrad, der von maschinell unterstützter Übersetzung bis hin zur vollautomatischen Übersetzung reicht.

Statistische Ansätze haben sich im Bereich der vollautomatischen Übersetzung als sehr erfolgreich herausgestellt. Hier sei als Beispiel das VERBMOBIL Projekt erwähnt. Bei diesem Projekt wurde ein System entwickelt, welches gesprochene Spontansprache erkennt und übersetzt. Die Abschlussevaluation ergab eine Fehlerrate von 29% für einen statistischen Ansatz, der von der RWTH entwickelt wurde. Die anderen Verfahren, darunter ein klassischer regelbasierter Ansatz, erreichten Fehlerraten zwischen 52% und 62% [Tessiore & v. Hahn 00].

Statistische Ansätze haben den weiteren Vorteil, dass die Übersetzungssysteme automatisch trainiert werden können. Das heißt, ausgehend von einem bilingualen Textkorpus werden alle Informationen, die zur Übersetzung benötigt werden, ohne menschliches Zutun erzeugt. Somit können dieses Ansätze prinzipiell für beliebige Sprachpaare, für die ein bilinguales Textkorpus existiert, verwendet werden. Beispielsweise wurden die Verfahren in dieser Diplomarbeit für die Sprachpaare Deutsch-Englisch, Italienisch-Englisch und Spanisch-Katalanisch eingesetzt.

Einzelwortbasierte Ansätze sind unter den statistischen Übersetzungsverfahren weit verbreitet. Bei diesen wird der Kontext eines Wortes für die Übersetzung nur sehr wenig benutzt, nämlich ausschließlich über das Sprachmodell. Daher werden in dieser Diplomarbeit drei Verfahren untersucht die Kontextinformationen explizit in die Übersetzung einfließen lassen.

Das erste Verfahren basiert auf bilingualen Phrasen. Hier werden anstelle

einzelner Wörter ganze Phrasen als kleinste Übersetzungseinheiten verwendet. Dieses phrasenbasierte Verfahren ist dem Alignment Template Ansatz [Och & Tillmann$^+$ 99] sehr ähnlich, besitzt jedoch ein deutlich einfacheres Modell. Hiervon werden verschiedene Varianten untersucht.

Als Erweiterung dieses Verfahrens wird ein Pattern-Pair Modell analog zu [Block 00] untersucht. Dieses modelliert explizit Umstellungen ganzer Phrasen. Es erlaubt auch die Übersetzung nicht-zusammenhängender Phrasen.

Das dritte Verfahren ist dem einzelwortbasierten Ansatz ähnlich. Es übersetzt einzelne Wörter, macht deren Übersetzung jedoch von dem lokalen Kontext abhängig. Es basiert auf einer Art von zweidimensionalem n-Gramm, einem n-Gramm auf dem Alignmentpfad.

Die Diplomarbeit ist wie folgt gegliedert:

Kapitel 2 – Grundlagen. Es werden die Grundlagen der statistischen maschinellen Übersetzung kurz beschrieben.

Kapitel 3 – Modelle. Hier werden folgende kontextabhängige Übersetzungsmodelle vorgestellt:

- Phrasenbasierte Übersetzung.
  Es werden bilinguale Phrasen anstelle von einzelnen Wörtern als kleinste Übersetzungseinheiten dienen.

- Pattern-Pairs.
  Die Pattern-Pairs modellieren explizit Umstellungen ganzer Phrasen.

- Zweidimensionale n-Gramme.
  Hier wird ein kontextabhängiges Lexikonmodell vorgestellt. Die Übersetzung eines Wortes wird von den zuvor übersetzten Wörtern und deren Übersetzungen abhängig gemacht.

Kapitel 4 – Suche. In diesem Kapitel werden ein monotones und zwei nichtmonotone Suchverfahren für die phrasenbasierte Übersetzung beschrieben.

Kapitel 5 – Ergebnisse. Hier werden die verschiedenen Experimente und deren Ergebnisse dargestellt.

Kapitel 6 – Zusammenfassung und Ausblick. Es wird eine kurze Zusammenfassung der Diplomarbeit gegeben. Außerdem wird auf weitere Verbesserungs- und Erweiterungsmöglichkeiten hingewiesen.

Anhang. Im Anhang sind die Skalierungsfaktoren der Modelle für die verschiedenen Experimente aufgelistet.

Kapitel 2

Grundlagen

2.1 Bayes'sche Entscheidungsregel

Das Ziel maschineller Übersetzung ist die automatische Übertragung der Bedeutung eines Satzes der Quellsprache $f_1^J = f_1 ... f_j ... f_J$ in einen Satz der Zielsprache $e_1^I = e_1 ... e_i ... e_I$. In der statistischen maschinellen Übersetzung wird die Beziehung zwischen den beiden Sätzen durch die bedingte Wahrscheinlichkeit $Pr(e_1^I | f_1^J)$[1] beschrieben. Mit Hilfe dieses Modells kann bereits die Übersetzung durchgeführt werden, indem man folgendes Maximierungsproblem löst:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \tag{2.1}$$

$$= \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \tag{2.2}$$

In der zweiten Gleichung wurde die Bayes'sche Formel angewendet. Die Dekomposition in zwei Wissensquellen reduziert die Komplexität der beiden Modelle und macht somit deren Modellierung einfacher. Man erhält nun folgende Komponenten:

1. Das **Sprachmodell** $Pr(e_1^I)$. Es beschreibt die Wohlgeformtheit der Übersetzungshypothese (siehe Abschnitt 2.3).

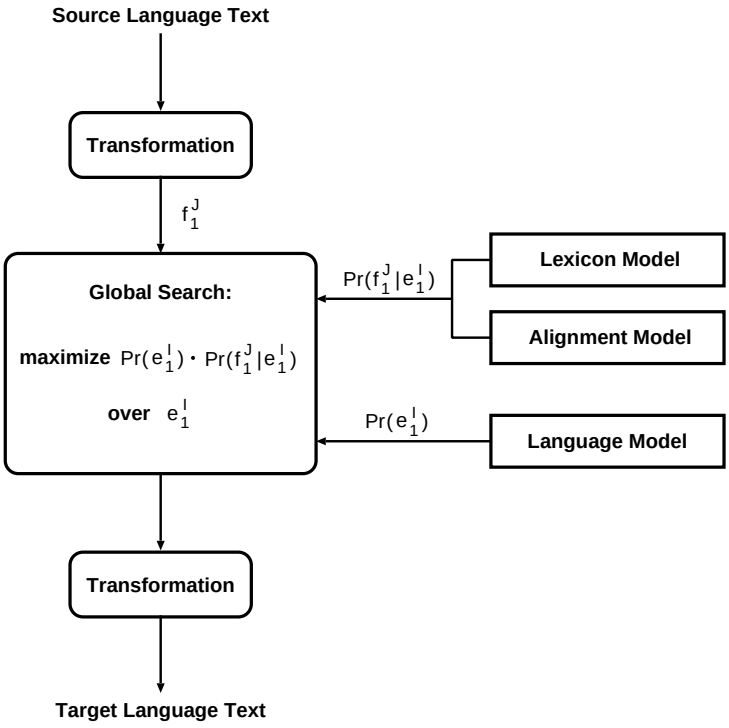
2. Das **Übersetzungsmodell** $Pr(f_1^J | e_1^I)$. Es beschreibt, wie gut die beiden Sätze als Übersetzungen zueinander passen (siehe Abschnitt 2.4).

[1] In dieser Diplomarbeit wird folgende Notation verwendet: das Symbol $Pr(\cdot)$ wird für allgemeine Wahrscheinlichkeitsverteilungen ohne spezifische Annahmen verwendet. Andererseits wird für Verteilungen mit speziellen Modellannahmen das Symbol $p(\cdot)$ benutzt.

3. Die **Suche**. Die Suche entspricht der argmax-Operation. Hier gilt es, die global beste Hypothese zu finden.

Die hieraus resultierende Architektur eines Übersetzungssystems ist in Abbildung 2.1 dargestellt. Hierbei wurde das Übersetzungsmodell $Pr(f_1^J|e_1^I)$ in Lexikon- und Alignmentmodell aufgeteilt (siehe Abschnitt 2.4).

Abbildung 2.1: Architektur des Übersetzungsansatzes basierend auf der Bayes'schen Entscheidungsregel.



2.2 Direkter Ansatz

Der Bayes-Ansatz im vorherigen Abschnitt besitzt gewisse Schwachstellen. Eine dieser Schwachstellen ist, dass zusätzliche Wissensquellen nur sehr schwierig eingebunden werden können. Daher wird in diesem Abschnitt eine Methode vorgestellt, die das einfache Hinzufügen weiterer Modelle erlaubt. Hierzu wird die Posteriorwahrscheinlichkeit $Pr(e_1^I|f_1^J)$ direkt modelliert. Der Maximum Entropy Ansatz [Berger & Della Pietra$^+$ 96] ist dazu besonders gut ge-

eignet. In [Och & Ney 02a] wird im Rahmen eines solchen Maximum Entropy Ansatzes folgendes Modell präsentiert:

$$Pr(e_1^I|f_1^J) = \frac{\exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J))}{\sum_{e'^I_1} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J))} \tag{2.3}$$

Die Posteriorwahrscheinlichkeit wird hier durch eine Menge von Featurefunktionen $h_m(e_1^I, f_1^J)$ modelliert. Jede dieser Funktionen wird mit einem Parameter λ_m gewichtet. Als Entscheidungsregel erhält man nun:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I|f_1^J) \right\} \tag{2.4}$$

$$= \underset{e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{2.5}$$

Hierbei ist wichtig, dass die rechenintensive Normierung wegfällt. Diese Entscheidungsregel kann als Verallgemeinerung der Bayes'schen Entscheidungsregel aufgefasst werden. Hierzu wählt man zwei Featurefunktionen: jeweils den Logarithmus von Sprachmodell- und Übersetzungsmodellwahrscheinlichkeit.

$$h_1(e_1^I, f_1^J) = \log Pr(e_1^I) \tag{2.6}$$

$$h_2(e_1^I, f_1^J) = \log Pr(f_1^J|e_1^I) \tag{2.7}$$

Die Skalierungsfaktoren λ_1 und λ_2 werden auf 1 gesetzt. Dies ergibt genau die Bayes'sche Entscheidungsregel. Experimente haben gezeigt, dass die Verwendung der 'inversen' Übersetzungswahrscheinlichkeit $Pr(e_1^I|f_1^J)$ anstelle von $Pr(f_1^J|e_1^I)$ vergleichbare, oft sogar bessere Übersetzungsergebnisse liefert wie die Standardrichtung $Pr(f_1^J|e_1^I)$. Für diesen Fall wählt man:

$$h_2(e_1^I, f_1^J) = \log Pr(e_1^I|f_1^J) \tag{2.8}$$

In dieser Diplomarbeit werden die Ergebnisse für jeweils beide Richtungen angegeben. Auch wenn diese Art der Modellierung der Posteriorwahrscheinlichkeit $Pr(e_1^I|f_1^J)$ von Maximum Entropy herstammt, werden die Skalierungsfaktoren λ_1^M in dieser Diplomarbeit nicht mit Maximum Entropy trainiert. Sie werden mit Hilfe eines Developmentkorpus so eingestellt, dass die Fehlerrate minimiert wird. Hierzu wird eine Heuristik verwendet. Eine detailliertere Beschreibung dieses Vorgehens wird in Abschnitt 5.1 geliefert werden.

2.3 Sprachmodell

In diesem Abschnitt wird das statistische n-Gramm Sprachmodell kurz beschrieben. Des weiteren wird das absolute Discounting mit Interpolation vorgestellt, da dieses später in Abschnitt 3.3 verwendet wird.

Das Sprachmodell beschreibt die Wohlgeformtheit der erzeugten Übersetzung. Es soll helfen, syntaktisch fehlerhafte Sätze zu vermeiden. Jeder möglichen Wortfolge e_1^I wird eine Wahrscheinlichkeit $Pr(e_1^I)$ zugeordnet. Je höher diese Wahrscheinlichkeit ist, desto typischer sollte diese Wortfolge für die Zielsprache sein, zumindest für den Bereich, für den das Sprachmodell trainiert wurde. Die Ausführungen hier basieren auf [Ney 01]. Die Wahrscheinlichkeit $Pr(e_1^I)$ wird meist faktorisiert, um sie auf Wortebene modellieren zu können.

$$Pr(e_1^I) = \prod_{i=1}^{I} Pr(e_i|e_1^{i-1}) \tag{2.9}$$

Wie man sieht, macht man die Wahrscheinlichkeit eines Wortes e_i von seinen Vorgängern e_1^{i-1} abhängig. Man beschränkt diese Abhängigkeit typischerweise auf eine bestimmte Anzahl der unmittelbaren Vorgänger, da diese den stärkeren Einfluss auf das aktuelle Wort e_i haben. Diese relevanten Vorgänger nennt man die Historie h des Sprachmodells. Unter einem n-Gramm versteht man eine Folge von n Wörtern. Daher spricht man von einem n-Gramm Sprachmodell, falls man die $n-1$ direkten Vorgänger betrachtet. Das heißt man schränkt die Historie bezüglich ihrer Länge auf $n-1$ ein. Die Historie zusammen mit dem vorherzusagenden Wort ergeben dann ein n-Gramm. Man erhält also:

$$Pr(e_1^I) = \prod_{i=1}^{I} p(e_i|e_{i-n+1}^{i-1}) \tag{2.10}$$

$$= \prod_{i=1}^{I} p(e_i|h_i) \tag{2.11}$$

Für den Fall, dass $n = 1$ gilt spricht man von einem Unigramm-Modell, bei $n = 2$ von einem Bigramm-Modell und bei $n = 3$ von einem Trigramm-Modell. Zu diesen Basismodellen gibt es eine Reihe von Verfeinerungen. Eine solche ist das absolute Discounting mit Interpolation. Dieses wird im Folgenden beschrieben.

Die Wahrscheinlichkeiten der n-Gramme werden über ihre relativen Häufigkeiten in einem (möglichst großen) Trainingskorpus geschätzt.

$$p(e|h) = \frac{N(h,e)}{N(h)} \tag{2.12}$$

Hierbei steht $N(h, e)$ für die absolute Häufigkeit (Count) des n-Grammes (h, e) in dem Trainingkorpus und $N(h)$ für die Häufigkeit der Historie h.

Es entsteht das Problem, dass im Verhältnis zu den vorhandenen Daten sehr viele Modellparameter zu schätzen sind (*data sparseness problem*). Insbesondere gibt es viele n-Gramme die nicht in dem Trainingskorpus vorkommen. Bei einer Vokabulargröße von 10 000 gibt es zum Beispiel 10^{12} mögliche Trigramme. Die üblichen Trainingskorpora sind aber deutlich kleiner. Verwendet man die relativen Häufigkeiten, so wird diesen ungesehenen Ereignissen eine Wahrscheinlichkeit von Null zugeordnet. Dies entspricht der Entscheidung, dass diese Wortfolge auch im Test nicht auftreten kann. Da man möglichst keine solche harten Entscheidungen treffen möchte, wendet man Glättungsmethoden an.

Eine solche Glättungsmethode, die sich bezüglich ihrer Leistungsfähigkeit und ihrer Einfachheit bewährt hat, ist das absolute Discounting mit Interpolation. Die Idee ist folgende: Sei e das vorherzusagende Wort und h die Historie. Man zieht von dem Count $N(h, e)$ den Discountingwert d ab, das heißt die Counts werden absolut verringert. Die so frei werdende Wahrscheinlichkeitsmasse wird nun der sogenannten Backing-Off Verteilung $p(e|\bar{h})$ zugewiesen. Hierbei steht $\bar{h}$ für die verallgemeinerte Historie. Bei n-Grammen verkürzt man typischerweise die Historie um 1 und erhält so die verallgemeinert Historie $\bar{h}$. Der Vorfaktor dieser Backing-Off Verteilung ergibt sich aus der Normierungsbedingung. Hierbei steht V für die Vokabulargröße und $N_0(h)$ für die Anzahl von Wörtern die nicht nach der Historie h gesehen wurden. Man erhält folgendes Modell:

$$p(e|h) \quad = \quad \max\left\{\frac{N(h,e)-d}{N(h)}, 0\right\} + d \cdot \frac{V - N_0(h)}{N(h)} \cdot p(e|\bar{h}) \qquad (2.13)$$

In dem Experimenten wird ein Trigramm-Sprachmodell mit absolutem Discounting und Interpolation verwendet. Außerdem wird diese Glättungsmethode bei den zweidimensionalen n-Grammen in Abschnitt 3.3 verwendet werden.

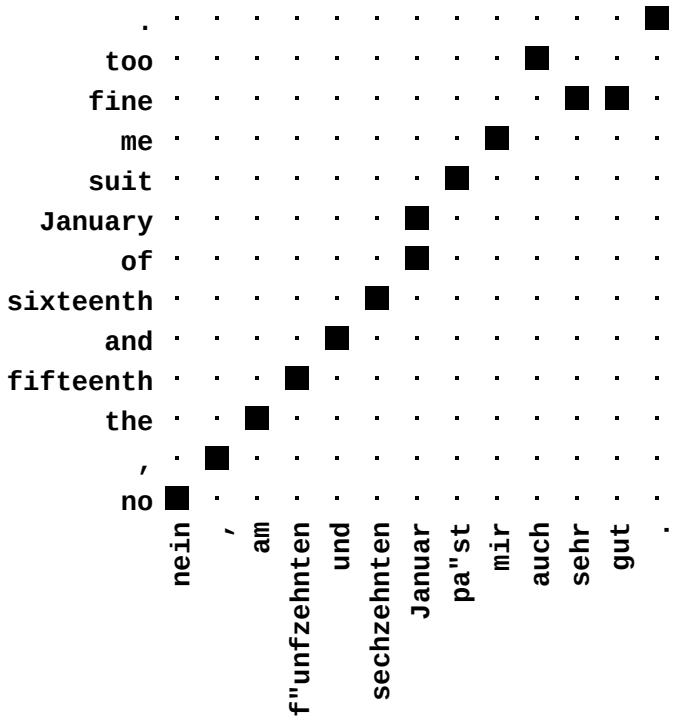
2.4 Übersetzungsmodell

In diesem Abschnitt wird die übliche Dekomposition der Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ beschrieben. Ein wichtiger Punkt in der Übersetzung ist die Wort-zu-Wort Abhängigkeit zwischen dem Satz der Quellsprache f_1^J und dem Satz der Zielsprache e_1^I. Diese Abhängigkeit wird durch das sogenannte *Alignment* modelliert [Brown & Della Pietra$^+$ 93, Vogel & Ney$^+$ 96].

Das Alignment ordnet jedem Wort des Quellsatzes die zugehörigen Wörter des Zielsatzes zu. Hierbei sind prinzipiell beliebige Zuordnungen möglich. Das Alignment A ist eine Relation mit $A \subseteq J \times I$.

Ein Beispiel für ein Alignment ist in Abbildung 2.2 zu sehen. Bei den Alignments eines bilingualen Textes stellt man fest, dass nur selten weite Sprünge auftreten. Das heißt, die Alignments eines Wortes liegen typischerweise in der Nähe der Alignments der umgebenden Wörter. Zusätzlich ist das Alignment häufig über weite Teile monoton. In dem Beispiel ist der Beginn des Alignments monoton und erst am Ende treten Wortumordnungen auf.

Abbildung 2.2: Wortaligniertes Satzpaar.



Das Alignment wird nun als sogenannte *hidden variable* in das Übersetzungmodell eingeführt.

$$Pr(f_1^J|e_1^I) = \sum_A Pr(f_1^J, A|e_1^I) \tag{2.14}$$

$$= Pr(J|e_1^I) \cdot \sum_A Pr(A|e_1^I, J) \cdot Pr(f_1^J|A, e_1^I, J) \tag{2.15}$$

Wie man in Gleichung (2.15) sieht, erhält man drei Verteilungen:

- Die Satzlängenwahrscheinlichkeit $Pr(J|e_1^I)$. Sie wird nur der Vollständigkeit halber erwähnt, da sie typischerweise weggelassen wird.
- Die Lexikonwahrscheinlichkeit $Pr(f_1^J|A, e_1^I, J)$.
- Die Alignmentwahrscheinlichkeit $Pr(A|e_1^I, J)$.

In den folgenden beiden Abschnitten werden für den einzelwortbasierten Ansatz das Lexikonmodell in Abschnitt 2.5 und verschiedene Alignmentmodelle in Abschnitt 2.6 beschrieben. Anschließend wird in Abschnitt 2.7 der Alignment Template Ansatz beschrieben. Dieser ist dem phrasenbasierten Ansatz in Abschnitt 3.1 sehr ähnlich.

2.5 Einzelwortbasiertes Lexikonmodell

Bei der einzelwortbasierten Übersetzung werden typischerweise nicht beliebige Alignments erlaubt, sondern nur Alignments mit gewissen Einschränkungen. Dazu nimmt man an, das Alignment sei eine Funktion $a : \{1, ..., J\} \rightarrow \{0, ..., I\}$ die jedem Wort des Quellsatzes ein Wort des Zielsatzes zuordnet. Hierbei gibt es die Besonderheit, dass e_0 als leeres Wort auch möglich ist. Dies bedeutet, ein Wort des Quellsatzes ist zu keinem Wort des Zielsatzes e_1^I aligniert. Das Wort wird also nicht übersetzt. Alignments dieser Art schreibt man meist als $a_1^J := a_1...a_j...a_J$. Dies bedeutet, dem Wort f_j wird das Wort $e_i = e_{a_j}$ zugeordnet.

Für die übliche Dekomposition der Lexikon- und Alignmentwahrscheinlichkeit ist diese Einschränkung des Alignments nötig. Die Lexikonwahrscheinlichkeit wird wie folgt faktorisiert:

$$Pr(f_1^J|a_1^J, e_1^I, J) = \prod_{j=1}^{J} Pr(f_j|f_1^{j-1}, a_1^J, e_1^I, J) \tag{2.16}$$

$$= \prod_{j=1}^{J} p(f_j|e_{a_j}) \tag{2.17}$$

In Gleichung (2.17) wurde die Abhängigkeit einzig auf das zugeordnete Wort eingeschränkt. Daher heißen Ansätze, welche diese Modellannahme verwenden auch einzelwortbasiert (*single-word based*). Es handelt sich um ein kontextunabhängiges Lexikonmodell. Das heißt, die Lexikonwahrscheinlichkeit ist unabhängig von den Wörtern in der Umgebung. In Abschnitt 3.3 wird

eine Möglichkeit vorgestellt, die Lexikonwahrscheinlichkeit kontextabhängig zu modellieren. Bei diesem Modell wird das Wissen über die zuvor alignierten Wörter mit in die Lexikonwahrscheinlichkeit einfließen. Dieses Wissen wird bei den einzelwortbasierten Modellen nicht genutzt.

Bei dem einzelwortbasierten Modell besteht durch die Einschränkung des Alignments noch ein zusätzliches Problem: Ein Wort der Quellsprache kann nicht durch mehrere Wörter der Zielsprache übersetzt werden. Bei der Übersetzung von Deutsch nach Englisch wäre zum Beispiel die Übersetzung von *'übermorgen'* mit *'the day after tomorrow'* nicht möglich. Um dieses Problem zu umgehen, wird das Vokabular der Zielsprache erweitert. In diesem Beispiel würde ein neues Wort *'the_day_after_tomorrow'* in das englische Vokabular aufgenommen. Das so erweiterte Vokabular wird dann für die Übersetzung verwendet. Bei der phrasenbasierten Übersetzung in Abschnitt 3.1 tritt dieses Problem nicht auf. Daher ist dort auch keine solche provisorische Lösung notwendig.

2.6 Einzelwortbasierte Alignmentmodelle

Für die Alignmentwahrscheinlichkeit $Pr(a_1^J|e_1^I, J)$ gibt es eine Vielzahl von Modellen. Einige grundlegende werden in diesem Abschnitt kurz dargestellt. In [Brown & Della Pietra$^+$ 93, Vogel & Ney$^+$ 96] sind diese detaillierter beschrieben.

2.6.1 IBM1-, IBM2- und HMM-Modell

Bei dem IBM1-, IBM2- und HMM-Modell wird die Alignmentwahrscheinlichkeit wie folgt faktorisiert:

$$Pr(a_1^J|e_1^I, J) = \prod_{j=1}^{J} Pr(a_j|a_1^{j-1}, e_1^I, J) \tag{2.18}$$

Bei den verschiedenen Modellen werden nun unterschiedliche Annahmen über die Wahrscheinlichkeit $Pr(a_j|a_1^{j-1}, e_1^I, J)$ gemacht.

- **IBM1-Modell**. Bei dem IBM1-Modell wird hier eine Gleichverteilung über die Positionen des Zielsatzes (inklusive des leeren Wortes e_0) angenommen.

$$Pr(a_j|a_1^{j-1}, e_1^I, J) = p(a_j|I) \tag{2.19}$$

$$= \frac{1}{I+1} \tag{2.20}$$

- **IBM2-Modell**. Hier wird die Wahrscheinlichkeit abhängig gemacht von den Satzlängen von Quellsatz J und Zielsatz I sowie von der aktuellen Position j im Quellsatz.

$$Pr(a_j|a_1^{j-1}, e_1^I, J) \quad = \quad p(a_j|j, I, J) \tag{2.21}$$

- **HMM-Modell**. Bei dem HMM-Modell wird die Wahrscheinlichkeit genauso wie beim IBM2-Modell von den beiden Satzlängen J und I abhängig gemacht. Außerdem besteht hier die Abhängigkeit von der zuvor alignierten Position im Zielsatz a_{j-1}.

$$Pr(a_j|a_1^{j-1}, e_1^I, J) \quad = \quad p(a_j|a_{j-1}, I, J) \tag{2.22}$$

2.6.2 IBM3-Modell

Das Modell IBM3 basiert auf der sogenannten *Fertilität*. Es ist in seinem Aufbau deutlich komplizierter als die einfachen Modelle in Abschnitt 2.6.1. Unter der Fertilität ϕ_i eines Wortes des Zielsatzes e_i versteht man die Anzahl der alignierten Wörter des Quellsatzes.

$$\phi_i \quad := \quad \sum_{j=1}^{J} \delta(a_j, j) \tag{2.23}$$

In das IBM3-Modell fließt nun eine Wahrscheinlichkeitsverteilung $p(\phi|e)$ ein. Diese modelliert, dass dem Wort des Zielsatzes e genau ϕ Wörter des Quellsatzes zugeordnet sind. Die Dekomposition der Alignmentwahrscheinlichkeit erfolgt bei dem IBM3-Modell etwas anders als bei den vorherigen Modellen. Sie wird entlang der Positionen i des Zielsatzes faktorisiert und nicht wie bisher entlang der Positionen j des Quellsatzes. Dazu sei B_i definiert als die Menge der Positionen der Wörter des Quellsatzes welche e_i zugeordnet sind.

$$B_i := \{j|a_j = i\} \tag{2.24}$$

Anschaulich entspricht dies einer Zeile in der Alignmentmatrix. Es wird häufig als *inverses* Alignment bezeichnet. Es ist zu beachten, dass jede dieser Menge B_i eine Mächtigkeit größer als eins haben kann, aber dass weiterhin jedes Wort des Quellsatzes genau einmal abgedeckt wird. Somit bilden die B_i eine Partition der Menge $\{1, ..., J\}$. Die Mächtigkeit einer Menge B_i entspricht nun genau der Fertilität ϕ_i des entsprechenden Wortes:

$$\phi_i \quad = \quad |B_i| \tag{2.25}$$

Die Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ wird nun wie folgt zerlegt. Dabei wird das Alignment B_0^I als *hidden variable* eingesetzt.

$$Pr(f_1^J|e_1^I) = \sum_{B_0^I} Pr(f_1^J, B_0^I|e_1^I) \tag{2.26}$$

$$= \sum_{B_0^I} Pr(B_0^I|e_1^I) \cdot Pr(f_1^J|e_1^I, B_0^I) \tag{2.27}$$

Im Unterschied zu der Zerlegung in Gleichung (2.15), tritt hier keine Satzlängenwahrscheinlichkeit auf. Die Satzlänge J ist durch die Fertilitäten bestimmt:

$$J = \sum_{i=0}^{I} \phi_i \tag{2.28}$$

Die Lexikonwahrscheinlichkeit $Pr(f_1^J|e_1^I, B_0^I)$ wird auch hier kontextunabhängig modelliert. Die Alignmentwahrscheinlichkeit $Pr(B_0^I|e_1^I)$ wird nun folgendermaßen zerlegt:

$$Pr(B_0^I|e_1^I) = Pr(B_0|B_1^I, e_1^I) \cdot \prod_{i=1}^{I} Pr(B_i|B_1^{i-1}, e_1^I) \tag{2.29}$$

$$= p(B_0|B_1^I) \cdot \prod_{i=1}^{I} p(B_i|e_i) \tag{2.30}$$

$$= p(\phi_0|\phi_1^I) \cdot \prod_{i=1}^{I} p(\phi_i|e_i)\phi_i! \prod_{j \in B_i} p(j|i, J) \tag{2.31}$$

In der ersten Gleichung wurde die Wahrscheinlichkeit $Pr(B_0|B_1^I, e_1^I)$ abgespalten, da diese gesondert modelliert wird. Im zweiten Schritt wurden die Modellannahmen gemacht. Im letzten Schritt wurde die Wahrscheinlichkeit $p(B_i|e_i)$ mit Hilfe der Fertilitätswahrscheinlichkeit $p(\phi_i|e_i)$ und der Alignmentwahrscheinlichkeit $p(j|i, J)$ weiter zerlegt. Auf eine Beschreibung der Modellierung der Wahrscheinlichkeit $p(\phi_0|\phi_1^I)$ wird hier verzichtet.

In [Och & Ney 02b] findet sich eine detaillierte Beschreibung aller hier vorgestellten Alignmentmodelle. Darüber hinaus werden dort unter anderem auch die Modelle IBM4 und IBM5 präsentiert.

2.7 Alignment Templates

In diesem Abschnitt wird eine kurze Beschreibung des Alignment Template Ansatzes [Och & Tillmann$^+$ 99] gegeben. Dieses Verfahren nutzt Alignment-

schablonen (*alignment templates*), bestehend aus einer Phrase der Quell- und Zielsprache sowie dem Alignment innerhalb der Phrasen, zur Übersetzung. Diese Phrasen sind allerdings nicht Folgen von Wörtern der Sprachen, sondern Folgen von Wortklassen. Dieser Ansatz ist dem in Abschnitt 3.1 beschriebenen phrasenbasiertem Ansatz sehr ähnlich. Dort wird allerdings das Alignment innerhalb der Phrasen vernachlässigt werden.

Bei dem Alignment Template Ansatz wird die Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ mit Hilfe von zwei *hidden variables* zerlegt. Zum Einen ist dies die Folge der Alignment Templates z_1^K und zum Anderen die Folge der Alignments innerhalb der Templates $\tilde{a}_1^K$.

$$Pr(f_1^J|e_1^I) = \sum_{z_1^K, \tilde{a}_1^K} Pr(\tilde{a}_1^K|e_1^I) \cdot Pr(z_1^K|\tilde{a}_1^K, e_1^I) \cdot Pr(f_1^J|z_1^K, \tilde{a}_1^K, e_1^I) \quad (2.32)$$

$$= \sum_{z_1^K, \tilde{a}_1^K} \prod_{k=1}^{K} p(\tilde{a}_k|\tilde{a}_{k-1}) \cdot p(z_k|\tilde{e}_k) \cdot p(\tilde{f}_k|z_k, \tilde{e}_k) \quad (2.33)$$

Man erhält nun drei Wahrscheinlichkeitsverteilungen:

- Die Wahrscheinlichkeit des Phrasenalignments $p(\tilde{a}_k|\tilde{a}_{k-1})$.
- Die Wahrscheinlichkeit der Anwendung eines Alignment Templates $p(z_k|\tilde{e}_k)$
- Die Phrasenübersetzungswahrscheinlichkeit $p(\tilde{f}_k|z_k, \tilde{e}_k)$

Bei diesem Ansatz kommt eine nicht-monotone Suche zum Einsatz. Dies ist nur eine sehr kurze Darstellung dieses Ansatzes. Weitere Details finden sich in [Och & Tillmann$^+$ 99].

Kapitel 3

Modelle

In diesem Kapitel werden die verschiedenen kontextabhängigen Übersetzungsmodelle präsentiert. Es wird zunächst in Abschnitt 3.1 die phrasenbasierte Übersetzung beschrieben. Anschließend wird in Abschnitt 3.2 das Pattern-Pair Modell vorgestellt. In Abschnitt 3.3 wird das Übersetzungsmodell mit zweidimensionalen n-Grammen dargestellt.

3.1 Phrasenbasierte Übersetzung

3.1.1 Motivation

Ein Nachteil des einzelwortbasierten Ansatzes ist, dass der Kontext nicht in das Übersetzungsmodell einfließt. Für viele Wörter hängt die Übersetzung jedoch von dem lokalen Kontext ab. Bei dem einzelwortbasierten Ansatz wird diese Unterscheidung nur durch das Sprachmodell durchgeführt. Das Sprachmodell ist jedoch in vielen Fällen nicht in der Lage diese Auswahl richtig zu treffen. Ein Beispiel ist in Abbildung 3.1 zu sehen.

Abbildung 3.1: Beispiel für einzelwortbasierte Übersetzung (SWB) und phrasenbasierte Übersetzung (PBT).

Quellsatz	was halten Sie vom Hotel Gewandhaus ?
Referenz	what do you think about hotel Gewandhaus ?
SWB	what do you from the hotel Gewandhaus ?
PBT	what do you think of hotel Gewandhaus ?

Eine Möglichkeit, den Kontext in das Übersetzungsmodell einfließen zu lassen, ist die Verwendung von bilingualen Phrasen. Das heißt, es werden nicht Übersetzungen für einzelne Wörter, sondern für ganze Phrasen gelernt. Somit ist die Übersetzung eines Wortes abhängig von dem lokalen Kontext innerhalb der Phrase. Eine Phrase ist hier einfach eine Folge von Wörtern. Eine bilinguale Phrase ist ein Phrasenpaar $(\tilde{f}, \tilde{e})$, bestehend je aus einer Phrase der Quell- und Zielsprache. Im Unterschied zu den Alignment Templates wird hier das Alignment innerhalb der Phrase vernachlässigt. Die zugrundeliegende Idee für die phrasenbasierte Übersetzung ist nun die folgende:

1. Segmentiere den Quellsatz in Phrasen.

2. Übersetze jede Phrase.

3. Setze den Zielsatz aus den Übersetzungen der einzelnen Phrasen zusammen.

Ein Beispiel für die Segmentierung und Übersetzung eines Satzes ist in Abbildung 3.2 dargestellt. Wie im Beispiel zu sehen ist, werden Satzzeichen als normale Wörter behandelt.

Abbildung 3.2: Phrasenbasierte Übersetzung des Satzes '*abends würde ich gerne entspannen und vielleicht in die Sauna gehen .*'

Segmentierung des Quellsatzes in Phrasen	Phrasenübersetzung
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
.	.

Im Folgenden wird zunächst in Abschnitt 3.1.2 die Definition von bilingualen Phrasen geliefert, sowie deren Extraktion aus einem Trainingskorpus beschrieben. Dann wird die Dekomposition der Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ dargestellt (Abschnitt 3.1.3). Daraufhin werden in Abschnitt 3.1.4 verschiedene Varianten des phrasenbasierten Übersetzungsmodells präsentiert.

3.1.2 Bilinguale Phrasen

Eine *bilinguale Phrase* ist ein Paar von m Wörtern der Quellsprache und n Wörtern der Zielsprache. Zusätzlich sollen die Wortfolgen jeweils zusammenhängend sein. Es gibt also keine Lücken. Ferner müssen die Phrasen konsistent mit der Alignmentmatrix sein. Dies bedeutet, dass die m Wörter der Quellphrase nur zu den n Wörtern der Zielphrase aligniert sind, und umgekehrt. Das folgende Kriterium beschreibt die Menge $\mathcal{BP}$ der bilingualen Phrasen eines Satzpaares (f_1^J, e_1^I) mit Alignmentmatrix A:

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, A) \;=\; & \big\{ \big(f_j^{j+m}, e_i^{i+n} \big) : \\ & \forall (i', j') \in A : j \leq j' \leq j+m \leftrightarrow i \leq i' \leq i+n \big\} \end{aligned} \tag{3.1}$$

Dieses Kriterium ist identisch mit dem Alignment Template Kriterium in [Och & Tillmann$^+$ 99]. In Abbildung 3.3 ist ein wortaligniertes Satzpaar und in Abbildung 3.4 sind die aus diesem Paar extrahierten bilingualen Phrasen angegeben. Eine bilinguale Phrase bildet immer ein Rechteck in der Alignmentmatrix. Für dieses gilt: alle Alignments von Wörtern der bilingualen Phrase, sei es in der Quellsprache oder in der Zielsprache, liegen innerhalb des Rechteckes.

Abbildung 3.3: Wortaligniertes Satzpaar.

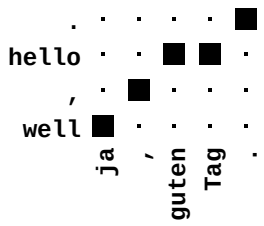


Abbildung 3.4: Extrahierte bilinguale Phrasen.

Quellphrase	Zielphrase
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.
.	.

Ausgehend von einem parallelen Text werden die bilingualen Phrasen nun wie folgt erzeugt. Es wird zunächst für jedes Satzpaar das Alignment bestimmt. Hierfür werden einzelwortbasierte Alignmentmodelle für beide Übersetzungsrichtungen trainiert. Dieses Training wird mit dem Tool GIZA++

[Och & Ney 00] durchgeführt. Es wird die Vereinigung der beiden Viterbi-Alignments gebildet. Weitere Details des Trainings sind in Abschnitt 5.1 beschrieben.

Nun werden die bilingualen Phrasen extrahiert. Dies kann mit dem Algorithmus `extract-BP` in Abbildung 3.5 gemacht werden. Für diesen wurde die Annahme gemacht, das Alignment sei eine Funktion $a : \{1, ..., J\} \rightarrow \{1, ..., I\}$. Er kann recht einfach erweitert werden, um beliebige Alignments zu handhaben. Der Algorithmus zählt alle möglichen Phrasen der Zielsprache auf. Für die zu dieser Phrase alignierten Wörter der Quellsprache wird mit der Funktion `consec()` überprüft, ob diese zusammenhängend (*consecutive*) sind. Falls dies zutrifft, so wird diese bilinguale Phrase zu der Menge aller bilingualen Phrasen hinzugefügt.

Abbildung 3.5: Algorithmus `extract-BP` zur Extraktion bilingualer Phrasen.

```
INPUT: f_1^J, e_1^I, a_1^J
FOR i_2 = 1 TO I DO
    FOR i_1 = 1 TO i_2 DO
        SP = {j | ∃i : i_1 ≤ i ≤ i_2 ∧ a_j = i}
        IF consec(SP) THEN
            j_1 = min{SP}
            j_2 = max{SP}
            BP = BP ∪ {(f_{j_1}^{j_2}, e_{i_1}^{i_2})}
OUTPUT: BP
```

Die Phrasenübersetzungswahrscheinlichkeiten $p(\tilde{f}|\tilde{e})$ werden durch relative Häufigkeiten geschätzt. Diese werden mittels eines Trainingskorpus bestimmt.

$$p(\tilde{f}|\tilde{e}) \quad = \quad \frac{N(\tilde{f}, \tilde{e})}{N(\tilde{e})} \tag{3.2}$$

Das Symbol $N(\tilde{e})$ bezeichnet die absolute Häufigkeit (Count) der Phrase $\tilde{e}$ in dem Trainingskorpus. $N(\tilde{f}, \tilde{e})$ ist die Häufigkeit, mit der die Phrase $\tilde{e}$ als Übersetzung der Phrase $\tilde{f}$ auftritt.

Hierbei ist noch ein Detail zu beachten. Es kann vorkommen, dass ein Wort der Zielsprache e nicht aligniert wurde. Dadurch kann es zu einer Quellphrase $\tilde{f}$ zwei mögliche Zielphrasen geben: einmal mit dem Wort e und einmal ohne. Beide erfüllen das Kriterium in (3.1). Jede dieser beiden Phrasen trägt dann mit 1/2 zu dem jeweiligen Count bei. Allgemein: gibt es N mögliche Zielphrasen $\tilde{e}_1, ..., \tilde{e}_N$ zu einer Quellphrase $\tilde{f}$, so trägt jede mit $1/N$ zu dem jeweiligen Count $N(\tilde{f}, \tilde{e}_n)$ bei.

3.1.3 Phrasenbasiertes Übersetzungsmodell

Eine Segmentierung $S = (\tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K)$ eines Satzpaares (f_1^J, e_1^I) ist eine Zerlegung in K bilinguale Phrasen inklusive des Alignments auf Phrasenebene $\tilde{a}_1^K$. Es wird von einem eins-zu-eins Alignment auf Phrasenebene ausgegangen. Das heißt, jeder Phrase der Quellsprache $\tilde{f}_k$ ist genau eine Phrase der Zielsprache $\tilde{e}_{\tilde{a}_k}$ zugeordnet und umgekehrt. Da die Phrasen in Ziel- und Quellsprache unterschiedlich lang sein können, ist dies keine echte Einschränkung. Eine Segmentierung S eines Satzpaares (f_1^J, e_1^I) enthält also die folgenden Informationen:

1. Eine Zerlegung des Quellsatzes f_1^J in eine Phrasenfolge $\tilde{f}_1^K$.
2. Eine Zerlegung des Zielsatzes e_1^I in eine Phrasenfolge $\tilde{e}_1^K$.
3. Einem Alignment auf Phrasenebene $\tilde{a}_1^K$.

Dies ist soweit analog zu der Segmentierung bei dem Alignment Template Ansatz. Im Unterschied zu den Alignment Templates wird hier jedoch das Alignment innerhalb der Phrasen vernachlässigt. Außerdem werden hier die Wörter selbst verwendet, während bei den Alignment Templates die Wörter durch ihre jeweilige Wortklasse ersetzt werden. Die Segmentierung S wird nun in das Übersetzungsmodell als *hidden variable* eingeführt:

$$Pr(f_1^J|e_1^I) = \sum_S Pr(f_1^J, S|e_1^I) \tag{3.3}$$

$$= \sum_S Pr(S|e_1^I) \cdot Pr(f_1^J|S, e_1^I) \tag{3.4}$$

$$= \sum_S Pr(S|e_1^I) \cdot Pr(\tilde{f}_1^K|\tilde{e}_1^K, \tilde{a}_1^K) \tag{3.5}$$

Die Phrasenübersetzungswahrscheinlichkeit $Pr(\tilde{f}_1^K|\tilde{e}_1^K, \tilde{a}_1^K)$ wird nun analog zu der Wortübersetzungswahrscheinlichkeit bei dem einzelwortbasierten An-

satz modelliert:

$$Pr(\tilde{f}_1^K|\tilde{e}_1^K, \tilde{a}_1^K) = \prod_{k=1}^{K} Pr(\tilde{f}_k|\tilde{f}_1^{k-1}, \tilde{e}_1^K, \tilde{a}_1^K) \tag{3.6}$$

$$= \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}) \tag{3.7}$$

In dem zweiten Schritt wurde bezüglich der Quellsprache eine zero-order Abhängigkeit angenommen, das heißt die Wahrscheinlichkeit ist unabhängig von den Vorgängerphrasen. Bezüglich der Zielsprache wurde die Abhängigkeit auf die alignierte Phrase eingeschränkt. Zusammen ergibt sich, dass die Übersetzungswahrscheinlichkeit einer Phrase der Quellsprache nur von der zugeordneten Phrase der Zielsprache abhängt. Dieses Übersetzungsmodell ist nun auf Phrasenebene kontextunabhängig und vergleichbar mit dem einzelwortbasierten Lexikonmodell auf Wortebene in Gleichung (2.17). Innerhalb der Phrasen wird der lokale Kontext jedoch berücksichtigt.

Als nächstes wird nur ein monotones Phrasenalignment betrachtet, es gilt $\tilde{a}_k = k$. Auf Phrasenebene sind die Übersetzungen nun monoton. Dies wird in einer sehr effizienten Suche resultieren (siehe 4.1). Daraus ergibt sich, dass Phrase $\tilde{f}_1$ von $\tilde{e}_1$ erzeugt wird, $\tilde{f}_2$ von $\tilde{e}_2$ und so weiter. Für die Wahrscheinlichkeit $Pr(\tilde{f}_1^K|\tilde{e}_1^K)$ ergibt sich somit:

$$Pr(\tilde{f}_1^K|\tilde{e}_1^K, \tilde{a}_1^K) = \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \tag{3.8}$$

Wie in Kapitel 5 gezeigt wird, erreicht man mit diesem einfachen Modell bereits deutliche Verbesserungen im Vergleich zu den kontextunabhängigen Übersetzungsmodellen wie sie bei dem einzelwortbasierten Ansatz eingesetzt werden.

Unter Verwendung eines n-Gramm Sprachmodell und der Bayes'schen Entscheidungsregel (2.2) erhält man folgendes Suchkriterium:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} \tag{3.9}$$

$$= \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h_i) \cdot \sum_S Pr(S|e_1^I) \cdot \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \right\} \tag{3.10}$$

$$\approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h_i) \cdot \sum_S \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k)^\lambda \right\} \tag{3.11}$$

Im letzten Schritt wurde die Segmentierungswahrscheinlichkeit $Pr(S|e_1^I)$ weggelassen. In Abschnitt 3.1.4 wird eine Variante vorgestellt, bei der diese Wahrscheinlichkeit modelliert wird. Des weiteren wurde der Skalierungsfaktor λ für das Übersetzungsmodell eingeführt [Tillmann 01].

In Gleichung (3.11) wird die Summe über alle Segmentierungen ausgeführt. Typischerweise wird in Übersetzungssystemen diese Summe durch den größten Summanden approximiert. Dies nennt man Maximum Approximation. Sie ist auch in der Spracherkennung weit verbreitet. In Kapitel 5 wird jedoch gezeigt, dass mit dem Summenkriterium bessere Ergebnisse erzielt werden können. Verwendet man die Maximum Approximation für die Summe über alle Segmentierungen, so erhält man folgendes Suchkriterium:

$$\hat{e}_1^I \approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h) \cdot \max_S \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k)^\lambda \right\} \tag{3.12}$$

3.1.4 Varianten

In diesem Abschnitt werden verschiedene Varianten der obigen Suchkriteria vorgestellt. Es wird jeweils das Suchkriterium für das Summenkriterium (3.11) präsentiert. Für die Maximum Approximation (3.12) ist es jeweils analog. Experimente haben gezeigt, dass diese Varianten die Übersetzungsergebnisse verbessern können (siehe Kapitel 5). Es werden folgende Varianten behandelt:

1. Modellierung der Segmentierungswahrscheinlichkeit $Pr(S|e_1^I)$
2. Wortstrafe (*word penalty*)
3. Phrasenstrafe (*phrase penalty*)
4. Glättung mit einem Lexikon

Modellierung der Segmentierungswahrscheinlichkeit

In dem oben beschriebenen Suchkriterium (3.11) wurde die Segmentierungswahrscheinlichkeit $Pr(S|e_1^I)$ weggelassen. Hier wird nun ein einfaches Modell für diese Wahrscheinlichkeit vorgestellt. Dabei wird nur die Anzahl der bilingualen Phrasen K relevant sein. Eine Segmentierung S segmentiert immer ein Satzpaar (f_1^J, e_1^I) aus Quellsatz und Zielsatz. Daher wird in Gleichung (3.13) die Annahme gemacht, die Wahrscheinlichkeit sei unabhängig

davon, ob sie über den Satz der Zielsprache e_1^I bedingt wird oder über den Satz der Quellsprache f_1^J.

$$\begin{aligned} Pr(S|e_1^I) &= Pr(S|f_1^J) && (3.13)\\ &= Pr(K|f_1^J) \cdot Pr(S|K, f_1^J) && (3.14)\\ &= p(K|f_1^J) \cdot Pr(S|K, f_1^J) && (3.15) \end{aligned}$$

In der zweiten Gleichung wurde die Wahrscheinlichkeit für die Phrasenanzahl $Pr(K|f_1^J)$ herausgezogen. Die Wahrscheinlichkeit $p(K|f_1^J)$ wird nun wie folgt geschätzt: Man definiert $s(k, j)$ als die Anzahl der Segmentierungen von f_1^j in k Segmente, die mit Hilfe der gelernten Phrasen gebildet werden können. Es werden hierbei also nicht alle möglichen Segmentierungen gezählt, sondern nur diejenigen die mit Phrasen aus dem Trainingskorpus erzeugt werden können. Um dies auszudrücken wird das Symbol $[N(f_1^j) > 0]$ wie folgt definiert:

$$[N(f_1^j) > 0] \quad := \quad \begin{cases} 1 & \text{falls } N(f_1^j) > 0 \\ 0 & \text{sonst} \end{cases} \qquad (3.16)$$

Hierbei ist $N(f_1^j)$ die absolute Häufigkeit der Phrase f_1^j in dem Trainingskorpus. Die Werte $s(k, j)$ lassen sich mit Dynamischer Programmierung effizient bestimmen. Die Rekursionsgleichungen lauten:

$$\begin{aligned} s(1, j) &= [N(f_1^j) > 0] \quad \forall 1 \leq j \leq J && (3.17)\\ s(k+1, j) &= \sum_{j'=k}^{j-1} \big(s(k, j') \cdot [N(f_{j'+1}^j) > 0]\big) && (3.18) \end{aligned}$$

Die erste Gleichung entspricht der Initialisierung. Sie bedeutet: gibt es eine Phrase f_1^j, so gibt es eine Segmentierung von f_1^j in ein Segment. Die zweite Gleichung beschreibt nun die Rekursion: Gibt es eine Phrase $f_{j'+1}^j$, so gibt es zu jeder Segmentierung von $f_1^{j'}$ in k Segmente genau eine Segmentierung von f_1^j in $k+1$ Segmente. Hierbei muss j' eine Position im Quellsatz zwischen k und $j-1$ sein. Es kann nicht kleiner sein, weil mindestens k Wörter notwendig sind um k Segmente zu bilden. Außerdem muss die neue Phrase $f_{j'+1}^j$ aus mindestens einem Wort bestehen. Somit ergibt sich die obige Gleichung.

Die Wahrscheinlichkeit $p(K|f_1^J)$ wird nun durch das Verhältnis der Anzahl der möglichen Segmentierungen in K Segmente zu der Gesamtzahl der möglichen Segmentierungen geschätzt :

$$p(K|f_1^J) \quad = \quad \frac{s(K, J)}{\sum_{K'} s(K', J)} \qquad (3.19)$$

Dies wird in Gleichung (3.10) eingesetzt. Die Wahrscheinlichkeit $Pr(S|K, f_1^J)$ wird nicht modelliert und daher weggelassen. Das Suchkriterium sieht nun folgendermaßen aus:

$$\hat{e}_1^I \approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h) \cdot \sum_S \left(p(K|f_1^J) \cdot \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \right) \right\} \quad (3.20)$$

Wortstrafe und Phrasenstrafe

Bei der Wort- und Phrasenstrafe handelt es sich nicht um Wahrscheinlichkeitsverteilungen, sondern um einfache Heuristiken. Wie man an den beiden Kriterien (3.21) und (3.22) sehen kann, lassen sie sich mit sehr geringem Aufwand in die Suche integrieren. Die freien Parameter c_W und c_P werden auf einem Developmentkorpus eingestellt.

Bei der Wortstrafe handelt es sich um eine konstante Strafe $c_W > 0$ pro Wort. Die Idee hierbei ist, die Satzlänge zu beeinflussen. Je stärker die Wortstrafe, desto mehr werden kürzere Sätze bevorzugt. Es ist auch erlaubt, dass die Wortstrafe c_W größer als 1 ist. Dann entspricht sie nicht mehr einer Strafe, sondern vielmehr einem Bonus pro Wort. Hierdurch können auch längere Sätze bevorzugt werden. Für einen Satz der Länge I ergibt sich ein Faktor $(c_W)^I$. Die Wortstrafe wird in der Spracherkennung häufig eingesetzt. Das Suchkriterium sieht nun folgendermaßen aus:

$$\hat{e}_1^I \approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} (p(e_i|h) \cdot c_W) \cdot \sum_B \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k)^\lambda \right\} \quad (3.21)$$

Analog zur Wortstrafe kann man auch eine Phrasenstrafe einführen. Dabei handelt es sich um eine konstante Strafe $c_P > 0$ pro Phrase. Die Idee hierbei ist, dass Segmentierungen in längere Phrasen bevorzugt werden sollen, da hier die Phrasenstrafe weniger oft angewendet wird. Auch hier wird $c_P > 1$ erlaubt, so dass auch ein Bonus pro Phrase möglich ist. Bei einer Segmentierung in K Phrasen ergibt sich ein Faktor $(c_P)^K$. Das Suchkriterium sieht nun folgendermaßen aus:

$$\hat{e}_1^I \approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h) \cdot \sum_B \prod_{k=1}^{K} \left(p(\tilde{f}_k|\tilde{e}_k)^\lambda \cdot c_P \right) \right\} \quad (3.22)$$

Glättung mittels Lexikon

Als nächstes wird die Glättung mit einem Lexikon behandelt. Dieses Lexikon ist dem einzelwortbasierten Lexikonmodell in Abschnitt 2.5 sehr ähnlich. Es

ist allerdings in der Lage beliebige Alignments zu handhaben. Bei dem einzelwortbasierten Modell ist es nicht erlaubt, dass einem Wort der Quellsprache mehrere Wörter der Zielsprache zugeordnet sind. Hier werden zwei Wege vorgestellt, diesen Fall zu handhaben: es wird das geometrische Mittel der einzelnen Wahrscheinlichkeiten genommen oder das arithmetische Mittel. Zur Bestimmung der Lexikonwahrscheinlichkeiten wird das Alignment innerhalb der Phrasen benötigt. Es wird dazu eine zweite *hidden variable* A für das Alignment eingeführt. Sei nun A_j definiert, als die Menge der Positionen der Wörter welche dem Wort f_j zugeordnet sind und e_{A_j} als die Menge ebendieser Wörter:

$$A_j := \{i|(j,i) \in A\} \tag{3.23}$$

$$e_{A_j} := \{e_i|i \in A_j\} \tag{3.24}$$

Anschaulich entspricht A_j einer Spalte der Alignmentmatrix. Für ein einzelnes Wort f_j ergibt sich nun dessen Lexikonwahrscheinlichkeit $p(f_j|e_{A_j})$, wie gesagt, als geometrisches Mittel (3.25) oder als arithmetisches Mittel (3.26) der einzelnen Wahrscheinlichkeiten:

$$p_g(f_j|e_{A_j}) = \prod_{i \in A_j} p(f_j|e_i)^{\frac{1}{|A_j|}} \tag{3.25}$$

$$p_a(f_j|e_{A_j}) = \frac{1}{|A_j|} \sum_{i \in A_j} p(f_j|e_i) \tag{3.26}$$

Bezüglich dieser beiden Methoden werden nur geringe Unterschiede erwartet, da es relativ selten vorkommen, einem Wort des Quellsatzes mehrere Wörter des Zielsatzes zugeordnet sind.

Diese Lexikonwahrscheinlichkeiten sind unabhängig von der Segmentierung in Phrasen. Daher kann man die Glättung der einzelnen Phrasen eines Satzes auch als log-lineare Interpolation mit dem Lexikonmodell auf Satzebene auffassen. Als Skalierungsfaktoren gibt es nun λ_1 für das Phrasenmodell und λ_2 für das Lexikonmodell. Das Suchkriterium sieht folgendermaßen aus:

$$\hat{e}_1^I \approx \underset{e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} p(e_i|h) \cdot \sum_B \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k)^{\lambda_1} \cdot \sum_A \prod_{j=1}^{J} p(f_j|e_{A_j})^{\lambda_2} \right\} \tag{3.27}$$

Hier ist nun für $p(f_j|e_{A_j})$ die gewünschte Variante mit geometrischem Mittel (3.25) oder arithmetischem Mittel (3.26) einzusetzen. Die Variante mit dem arithmetischen Mittel ist dem Alignment Template Modell (2.33) sehr ähnlich. Dort wird zusätzlich noch die Wahrscheinlichkeit des Phrasenalignments

$p(\tilde{a}_k|\tilde{a}_{k-1})$ genutzt. Außerdem besteht natürlich noch der Unterschied, dass die Alignment Templates Wortklassen verwenden. Ansonsten sind die beiden Modelle identisch.

Hier wurde die Glättung mit einem kontextunabhängigen Lexikonmodell vorgestellt. In Abschnitt 3.3.4 wird stattdessen ein kontextabhängiges Modell eingesetzt werden, die zweidimensionalen n-Gramme.

3.2 Pattern-Pairs

3.2.1 Motivation

Die bisherigen Ausführungen betrachteten nur monotone Übersetzungen. Wie in Kapitel 5 gezeigt wird, ist die Monotonie jedoch die Ursache vieler Übersetzungsfehler. Bei der Übersetzung zwischen Deutsch und Englisch ist zum Beispiel die Wortreihenfolge bei der Verbgruppe häufig unterschiedlich. Ein Beispiel ist in Abbildung 3.6 zu sehen.

Abbildung 3.6: Beispiel für eine Übersetzung mit Pattern-Pairs.

Quellsatz	ich würde gerne am Mittwoch den 26. Januar losfahren.
Referenz	I would like to leave on Wednesday the 26th of January.
monoton	I would like to on Wednesday the 26th of January to leave.
Pattern-Pairs	I would like to leave on Wednesday the 26th of January.

In diesem Abschnitt wird eine Erweiterung des phrasenbasierten Ansatzes aus Abschnitt 3.1 um sogenannte Pattern-Pairs vorgestellt. Diese Pattern-Pairs modellieren explizit Umordnungen von Phrasen. Die Ausführungen hier basieren auf [Block 00]. Dort wird ein Pattern-Pair Modell für Phrasen von Wortklassen vorgestellt. Es wird dort beschrieben, was Pattern-Pairs sind und wie sie in der Suche eingesetzt werden, die Modellgleichungen werden aber nicht angegeben.

Im nächsten Abschnitt 3.2.2 werden die Pattern-Pairs detaillierter beschrieben. Es wird eine formale Definition geliefert, sowie die Erzeugung der Pattern-Pairs beschrieben. In Abschnitt 3.2.3 wird das Pattern-Pair Übersetzungmodell dargestellt. Hierzu wird eine stochastische Grammatik verwendet werden.

Abbildung 3.7: Beispiele für Pattern-Pairs.

Quellanteil $\bar{f}$	Zielanteil $\bar{e}$
am V den fünften	on V the fifth
am V losfahren	leave on the V
fahren V los	are leaving V
fahren um V los	are leaving at V
V zu planen	to plan V
ich würde V vorschlagen	I would suggest V
nach V fahren	go to V
wenn Sie V wollen	if you want to V

3.2.2 Definition und Extraktion

Allgemein ist ein Pattern-Pair $(\bar{f}, \bar{e})$ eine Regel der folgenden Art:

$$\bar{f} = (\tilde{f}_1 V \tilde{f}_2) \quad \rightarrow \quad \bar{e} = (\tilde{e}_1 V \tilde{e}_2) \tag{3.28}$$

Hierbei ist V ein Platzhalter für eine beliebige bilinguale Phrase. Setzt man für V die bilinguale Phrase $(\tilde{f}, \tilde{e})$ ein, so bedeutet dies: die Folge $(\tilde{f}_1 \tilde{f} \tilde{f}_2)$ kann durch $(\tilde{e}_1 \tilde{e} \tilde{e}_2)$ übersetzt werden. Durch die Verwendung dieses Platzhalters ergibt sich eine Verallgemeinerungsfähigkeit der Pattern-Pairs. Einige Beispiele für Pattern-Pairs sind in Abbildung 3.7 zu sehen.

Es ist zu bemerken, dass $\tilde{e}_1$ *nicht* unbedingt die Übersetzung von $\tilde{f}_1$ sein muss, beziehungsweise $\tilde{e}_2$ von $\tilde{f}_2$, auch wenn dies durch die Indizes suggeriert wird. Der Teil der Quellphrase vor dem Platzhalter V wird mit $\tilde{f}_1$ bezeichnet und der Teil dahinter mit $\tilde{f}_2$. Analog gilt für die Zielphrase: $\tilde{e}_1$ ist der Teil vor dem Platzhalter V und $\tilde{e}_2$ ist der Teil dahinter. Es sollte gelten, dass die Phrasenfolge $(\tilde{e}_1 V \tilde{e}_2)$ als Ganzes gesehen eine mögliche Übersetzung der Folge $(\tilde{f}_1 V \tilde{f}_2)$ ist.

Es kann auch vorkommen, dass zum Beispiel $\tilde{e}_1$ sich weder $\tilde{f}_1$ noch $\tilde{f}_2$ richtig zuordnen lässt. Ein Beispiel für einen solchen Fall ist das Pattern-Pair '*am V losfahren* $\rightarrow$ *leave on the V*'. Hier kann '*leave on the*', was $\tilde{e}_1$ entspricht, weder '*am*' noch '*losfahren*' eindeutig zugeordnet werden. Für eine korrekte Zuordnung müsste man es aufteilen.

Wie man an diesem Beispiel sieht, ist es auch möglich, dass einer der beiden Teile $\tilde{e}_1$ oder $\tilde{e}_2$, beziehungsweise $\tilde{f}_1$ oder $\tilde{f}_2$ leer ist. Es wird somit genau eine Umordnung beschrieben, falls $\tilde{f}_1$ und $\tilde{e}_2$ leer sind, beziehungsweise können auch $\tilde{f}_2$ und $\tilde{e}_1$ leer sein. Die Pattern-Pairs haben dann folgende

Gestalt:

$$\bar{f} = (V\tilde{f}_2) \quad \rightarrow \quad \bar{e} = (\tilde{e}_1 V) \tag{3.29}$$
$$\bar{f} = (\tilde{f}_1 V) \quad \rightarrow \quad \bar{e} = (V\tilde{e}_2) \tag{3.30}$$

Es werden aber nicht nur Umordnungen ermöglicht, sondern es besteht auch die Möglichkeit nicht-zusammenhängende Phrasen korrekt zu übersetzen. Im Deutschen zum Beispiel ist die Verbphrase häufig nicht zusammenhängend. Dies verursacht typischerweise Probleme bei der Übersetzung. Mit Hilfe von Pattern-Pairs ist es möglich, die Verbphrase in vielen Fällen korrekt zu übersetzen. Ein Beispiel ist in Abbildung 3.8 zu sehen. Bei diesem kommt das Pattern-Pair '*wir fahren V zurück* → *we go back V*' zum Einsatz.

Abbildung 3.8: Beispiel für eine Übersetzung mit Pattern-Pairs.

Quellsatz	ich schlage vor wir fahren abends am 24. zurück .
monoton	I suggest we go in the evening on the 24th back .
Pattern-Pairs	I suggest we go back on the 24th in the evening .

Die hier beschriebenen Pattern-Pairs haben also folgende Eigenschaften:

- Umordnungen von Phrasen.
  Zum Beispiel: *V zu planen* → *to plan V*
- Übersetzung nicht-zusammenhängender Phrasen.
  Zum Beispiel: *fahren V los* → *are leaving V*
- Verallgemeinerung.
  Zum Beispiel: *am V den fünften* → *on V the fifth*

Die Pattern-Pairs werden aus den bilingualen Phrasen erzeugt. Für alle Paare von bilingualen Phrasen $((\tilde{f}, \tilde{e}), (\tilde{f}_s, \tilde{e}_s))$, bei denen $(\tilde{f}_s, \tilde{e}_s)$ als Subphrase von $(\tilde{f}, \tilde{e})$ auftritt, ersetzt man in $\tilde{f}$ die Wortfolge $\tilde{f}_s$ durch den Platzhalter V und in $\tilde{e}$ wird $\tilde{e}_s$ durch V ersetzt. Einige Beispiele sind in Abbildung 3.9 zu sehen.

Das folgende Kriterium beschreibt die Menge $\mathcal{PP}$ der Pattern-Pairs eines Satzpaares (f_1^J, e_1^I) mit Alignmentmatrix A. Hierbei wird die Menge $\mathcal{BP}$ der bilingualen Phrasen dieses Satzpaares, wie Abschnitt 3.1.2 beschrieben, benutzt.

$$\begin{aligned} \mathcal{PP}(f_1^J, e_1^I, A) \;=\; & \Big\{ \Big((\tilde{f}_1 V \tilde{f}_2), (\tilde{e}_1 V \tilde{e}_2) \Big) : \\ & \exists (\tilde{f}, \tilde{e}), (\tilde{f}_s, \tilde{e}_s) \in \mathcal{BP}(f_1^J, e_1^I, A) : \tilde{f} = \tilde{f}_1 \tilde{f}_s \tilde{f}_2 \wedge \tilde{e} = \tilde{e}_1 \tilde{e}_s \tilde{e}_2 \; \Big\} \end{aligned} \tag{3.31}$$

Abbildung 3.9: Beispiele für die Erzeugung von Pattern-Pairs.

$(\tilde{f}, \tilde{e})$ $(\tilde{f}_s, \tilde{e}_s)$	am Dienstag den fünften Dienstag	on Tuesday the fifth Tuesday
Pattern-Pair	am V den fünften	on V the fifth

$(\tilde{f}, \tilde{e})$ $(\tilde{f}_s, \tilde{e}_s)$	wir fahren um eins los um eins	we are leaving at one at one
Pattern-Pair	wir fahren V los	we are leaving V

$(\tilde{f}, \tilde{e})$ $(\tilde{f}_s, \tilde{e}_s)$	würde mir gut passen mir gut	would suit me fine me fine
Pattern-Pair	würde V passen	would suit V

Bei den Experimenten werden die Pattern-Pairs, die offensichtliche eine monotone Übersetzung beschreiben, aussortiert. Dies sind Pattern-Pairs der folgenden Gestalt:

$$\bar{f} = (\tilde{f}_1 V) \quad \rightarrow \quad \bar{e} = (\tilde{e}_1 V) \tag{3.32}$$

$$\bar{f} = (V \tilde{f}_2) \quad \rightarrow \quad \bar{e} = (V \tilde{e}_2) \tag{3.33}$$

Die Wahrscheinlichkeit der Anwendung eines Pattern-Pairs $p(\bar{f}|\bar{e})$ wird durch dessen relative Häufigkeit in dem Trainingskorpus bestimmt.

$$p(\bar{f}|\bar{e}) \quad = \quad \frac{N(\bar{f}, \bar{e})}{N(\bar{e})} = \frac{N\left((\tilde{f}_1 V \tilde{f}_2), (\tilde{e}_1 V \tilde{e}_2)\right)}{N\left(\tilde{e}_1 V \tilde{e}_2\right)} \tag{3.34}$$

3.2.3 Pattern-Pair Übersetzungsmodell

In diesem Abschnitt wird das Übersetzungsmodell für die Pattern-Pairs vorgestellt. Auch bei dem Pattern-Pair Übersetzungsmodell wird eine Segmentierung als *hidden variable* verwendet. Hier wird allerdings eine geringfügig andere Segmentierung benötigt. Die Anzahl der Phrasen des Quellsatzes kann größer sein als die Anzahl der Phrase des Zielsatzes. Jeder Phrase des Quellsatzes wird weiterhin genau eine Phrase des Zielsatzes zugeordnet. Jeder Phrase des Zielsatzes wird hier mindestens eine Phrase des Quellsatzes zugeordnet. Einer Zielphrase können also mehrere Phrasen des Quellsatzes zugeordnet sein. Eine Segmentierung enthält nun folgende Informationen:

- Eine Zerlegung des Quellsatzes f_1^J in eine Phrasenfolge $\tilde{f}_1^K$.
- Eine Zerlegung des Zielsatzes e_1^I in eine Phrasenfolge $\tilde{e}_1^{K'}$ mit $K' \leq K$.
- Ein Alignment auf Phrasenebene $\tilde{a}_1^K$.
  Dies ist eine surjektive Funktion $\tilde{a} : \{1, ..., K\} \rightarrow \{1, ..., K'\}$.

Zur Modellierung der Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ wird hier eine stochastische kontextfreie Grammatik verwendet. Diese wird im Folgenden beschrieben.

Es ist zu bedenken, dass bei der Standardrichtung $p(f|e)$ ausgehend von dem Satz der Zielsprache, der Satz der Quellsprache erzeugt wird. Die Terminals der Grammatik sind also die Wörter f der Quellsprache. Die Non-Terminals entsprechen den Phrasen sowohl der Zielsprache als auch der Quellsprache. Daher wird hier $\tilde{E}$ beziehungsweise $\tilde{F}$ als Symbol für diese gewählt. Hierbei soll es für jede Phrase der Zielsprache $\tilde{e}$ ein Non-Terminal $\tilde{E}$ geben. Gleichermaßen gibt es für jede Phrase der Quellsprache $\tilde{f}$ ein entsprechendes Non-Terminal $\tilde{F}$. Für die Quellsprache gibt es noch das besondere Symbol $\tilde{F}_0$. Dies soll einer leeren Phrase entsprechen, also einer Phrase ohne Wörter. Abgesehen von diesen Phrasensymbolen ist natürlich das Startsymbol S ein Non-Terminal.

Bei den beschriebenen Regeln handelt es sich genaugenommen um Regelklassen. So gibt es zum Beispiel für jede bilinguale Phrase $(\tilde{f}, \tilde{e})$ eine Regel der Klasse (3.36). Man hat nun die folgenden Regeln:

$$S \rightarrow \tilde{E}_1^{K'} \tag{3.35}$$
$$\tilde{E} \rightarrow \tilde{F} \tag{3.36}$$
$$\tilde{E} \rightarrow \tilde{F}_1 \tilde{E}' \tilde{F}_2 \tag{3.37}$$
$$\tilde{F} \rightarrow f_1^j \tag{3.38}$$

Ausgehend von dem Startsymbol S wird zunächst der segmentierte Zielsatz erzeugt, beziehungsweise die Folge der entsprechenden Non-Terminals. Die Wahrscheinlichkeit dieser Regelanwendung ist die Segmentierungswahrscheinlichkeit.

Die zweite Regel entspricht der normalen Phrasenübersetzung. Die zugehörige Regelwahrscheinlichkeit ist also die Phrasenübersetzungswahrscheinlichkeit $p(\tilde{f}|\tilde{e})$, wobei die Phrasen $\tilde{f}$ und $\tilde{e}$ den Non-Terminals $\tilde{F}$ und $\tilde{E}$ entsprechen sollen. Bei dieser Regel kann das Non-Terminal $\tilde{F}_0$ nicht auf der

rechten Seite auftreten, da es keine bilinguale Phrase mit leerer Quellphrase gibt.

In der dritten Regel wird das Pattern-Pair $(\bar{f}|\bar{e}) = ((\tilde{f}_1 V \tilde{f}_2), (\tilde{e}_1 V \tilde{e}_2))$ angewendet. Hierbei sollen die Non-Terminals $\tilde{F}_1$ und $\tilde{F}_2$ den Phrase $\tilde{f}_1$ und $\tilde{f}_2$ entsprechen. Falls eine der beiden Teile leer ist, so ist das zugehörige Non-Terminal gleich dem Non-Terminal $\tilde{F}_0$. Die Wahrscheinlichkeit der Anwendung dieser Regel ist die Pattern-Pair Wahrscheinlichkeit $p(\bar{f}|\bar{e})$. Es wurde $\tilde{E}'$ für den Platzhalter V eingesetzt. Demzufolge muss die Phrasenfolge $(\tilde{e}_1 \tilde{e}' \tilde{e}_2)$ der Phrase $\tilde{e}$ entsprechen. Es gilt also $\tilde{e} = (\tilde{e}_1 \tilde{e}' \tilde{e}_2)$. Hierbei ist $\tilde{e}$ die Phrase des Non-Terminals $\tilde{E}$ und $\tilde{e}'$ diejenige des Non-Terminals $\tilde{E}'$.

Zu dieser Pattern-Pair Regel sei noch bemerkt, dass auf das Non-Terminal $\tilde{E}'$ erneut eine Pattern-Pair Regel angewendet werden kann. Die Pattern-Pairs können somit hierarchisch benutzt werden. Prinzipiell kann diese Hierarchie beliebig tief sein. In der Suche in Abschnitt 4.2.2 wird sie jedoch eingeschränkt. Die maximale Anzahl der wiederholten Anwendung der Pattern-Pair Regel wird als Level bezeichnet.

Die letzte Regel beschreibt den Übergang von einer Phrase zu den einzelnen Wörtern dieser Phrase. Dies ist eine deterministische Regel. Sie erhält somit Wahrscheinlichkeit 1. In diese Regelklasse soll auch der Übergang von dem Non-Terminal $\tilde{F}_0$ zu dem leeren Wort gehören.

Mit Hilfe dieser Grammatik lässt sich nun die Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ beschreiben. Die Darstellung ist analog zu [Ney 00, Kapitel 7]. Sei nun $r_1^M : S \rightarrow f_1^J$ eine Ableitung dieser Grammatik. Das heißt, mit den Regelanwendungen $r_1^M = r_1 ... r_m ... r_M$ wird aus dem Startsymbol S die Folge der Terminals f_1^J erzeugt. Die Wahrscheinlichkeit einer solchen Ableitung ist nun das Produkt der einzelnen Regelwahrscheinlichkeiten. Man erhält also:

$$p(r_1^M : S \rightarrow f_1^J|S) \quad = \quad \prod_{m=1}^{M} p(r_m) \tag{3.39}$$

Da es mehrere Ableitungen geben kann, die einen Satz f_1^J erzeugen, wird nun zur Modellierung der Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ über all diese summiert. Dies entspricht der Summierung über alle Segmentierungen.

$$Pr(f_1^J|e_1^I) \quad = \quad \sum_{r_1^M : S \rightarrow f_1^J} \prod_{m=1}^{M} p(r_m) \tag{3.40}$$

Die monotone Übersetzung tritt hier als Spezialfall auf: Wird die Pattern-Pair Regel (3.37) nicht verwendet, so entspricht dies genau der monotonen phrasenbasierten Übersetzung.

3.3 Zweidimensionale n-Gramme

3.3.1 Motivation

Ein Nachteil der phrasenbasierten Übersetzung aus Abschnitt 3.1 ist, dass es keine Abhängigkeit über die Phrasengrenzen hinaus gibt. Eine solche Abhängigkeit wäre jedoch sehr wünschenswert. In diesem Abschnitt wird ein Modell vorgestellt, welches auf kontextabhängigen Lexikonwahrscheinlichkeiten basiert. Die Idee ist, analog zu den n-Grammen der Sprachmodellierung, ein n-Gramm auf dem Alignmentpfad zu definieren.

3.3.2 Definition

Gegeben sei ein Satzpaar (f_1^J, e_1^I) mit dem inversen Alignment B_0^I. Dieses Alignment soll die übliche Einschränkung haben, dass jedem Wort des Quellsatzes genau ein Wort des Zielsatzes (inklusive des leeren Wortes e_0) zugeordnet ist.

Unter einem *zweidimensionalen n-Gramm* versteht man eine Folge von n Wörtern des Zielsatzes e_{i-n+1}^i zusammen mit ihrem Alignment B_{i-n+1}^i und den alignierten Wörtern des Quellsatzes $f_{B_{i-n+1}}^{B_i}$. Hierbei sei f_{B_i} die Menge der Wörter des Quellsatzes welche e_i zugeordnet sind.

$$f_{B_i} \quad := \quad \{f_j | j \in B_i\} \tag{3.41}$$

Auf den ersten Blick ist ein solches zweidimensionales n-Gramm einer bilingualen Phrase, beziehungsweise einem Alignment Template, recht ähnlich. Bei den zweidimensionale n-Grammen ist es aber erlaubt, dass die Wörter des Quellsatzes nicht direkt aufeinanderfolgend sind. Hier können also Lücken vorkommen. Außerdem wird die Verwendung der zweidimensionalen n-Gramme bei der Dekomposition des Übersetzungsmodells in Abschnitt 3.3.3 anders sein. Es wird keine feste Segmentierung gewählt. Die zweidimensionalen n-Gramme können überlappend eingesetzt werden.

Bei der obigen Beschreibung der zweidimensionalen n-Gramme wurde noch ein Detail vernachlässigt. Sind einem Wort des Zielsatzes e_i mehrere Wörter der Quellsatzes zugeordnet, so entsprechen dem auch mehrere zweidimensionale n-Gramme, nämlich genau eines pro zugeordnetem Wort.

Um dies auszudrücken, wird die Definition von $f_{B_{i,<j}}$ benötigt. Dies ist eine Teilmenge von f_{B_i}. Sie enthält alle Wörter die dem Wort e_i zugeordnet sind und deren Position kleiner als j ist.

$$f_{B_{i,<j}} \quad := \quad \{f_{j'} | j' \in B_i \wedge j' < j\} \tag{3.42}$$

Die Menge der zweidimensionalen n-Gramme $\mathcal{NG}$ eines Satzpaares (f_1^J, e_1^I) mit Alignment B_0^I ist nun wie folgt definiert:

$$\begin{aligned} \mathcal{NG}(f_1^J, e_1^I, B_0^I) \quad := \quad & \Big\{ \Big(e_{i-n+1}^i, B_{i-n+1}^i, f_{B_{i-n+1}}^{B_{i-1}}, f_{B_{i,<j}}, f_j \Big) : \\ & j \in B_i \wedge 1 \leq i \leq I \Big\} \end{aligned} \tag{3.43}$$

Später, bei der Verwendung der zweidimensionalen n-Gramme zur Dekomposition des Übersetzungsmodells, soll das Wort f_j vorausgesagt werden. Hierbei wird mittels eines zweidimensionalen n-Grammes folgende Information genutzt:

- die letzten n Wörter des Zielsatzes bis zu dem Wort e_i, also e_{i-n+1}^i,
- das zugehörige Alignment B_{i-n+1}^i,
- die bis zur Position $i-1$ alignierten Wörter des Quellsatzes, also $f_{B_{i-n+1}}^{B_{i-1}}$ und
- die bereits zu dem Wort e_i alignierten Wörter des Quellsatzes. Dies sind alle Wörter die zu e_i aligniert wurden und sich links von f_j befinden, also $f_{B_{i,<j}}$.

Diese Information, die zur Vorhersage genutzt wird, wird analog zur Sprachmodellierung als Historie bezeichnet und mit h abgekürzt. Außerdem wird die verallgemeinerte Historie wieder mit $\bar{h}$ bezeichnet. Diese entsteht, wenn man das hinterste Wort der Zielsprache, also e_{i-n+1}, dessen Alignment B_{i-n+1}, sowie die alignierten Wörter des Quellsatzes $f_{B_{i-n+1}}$ weglässt.

$$h \quad := \quad (e_{i-n+1}^i, B_{i-n+1}^i, f_{B_{i-n+1}}^{B_{i-1}}, f_{B_{i,<j}}) \tag{3.44}$$

$$\bar{h} \quad := \quad (e_{i-n+2}^i, B_{i-n+2}^i, f_{B_{i-n+2}}^{B_{i-1}}, f_{B_{i,<j}}) \tag{3.45}$$

Geschätzt werden die Wahrscheinlichkeiten der zweidimensionalen n-Gramme durch relative Häufigkeiten:

$$p(f|h) \quad = \quad \frac{N(f,h)}{N(h)} \tag{3.46}$$

Für ein solches Modell müssen nun sehr viele Parameter geschätzt werden. Das Problem der Datenknappheit (*data sparseness problem*) tritt um so deutlicher auf. Daher ist eine geeignete Glättung besonders wichtig. Man kann

nun die gleichen Glättungsmethoden wie in der Sprachmodellierung anwenden. Mit dem absoluten Discounting mit Interpolation (2.13) ergibt sich:

$$p(f|h) = \max\left\{\frac{N(f,h)-d}{N(h)}, 0\right\} + d \cdot \frac{V_f - N_0(h)}{N(h)} \cdot p(f|\bar{h}) \tag{3.47}$$

Hierbei ist V_f die Vokabulargröße der Quellsprache. Das Symbol $N_0(h)$ steht, wie in Abschnitt 2.3, für die Anzahl der Wörter die nach der Historie h nicht gesehen wurden.

3.3.3 Übersetzungsmodell mit zweidimensionalen n-Grammen

In diesem Abschnitt wird die Übersetzungswahrscheinlichkeit $Pr(f_1^J|e_1^I)$ so zerlegt, dass die zweidimensionalen n-Gramme zum Einsatz kommen. Hierzu wird das inverse Alignment B_0^I als *hidden variable* eingesetzt.

$$\begin{aligned} Pr(f_1^J|e_1^I) &= \sum_{B_0^I} Pr(f_1^J, B_0^I|e_1^I) && (3.48)\\ &= Pr(J|e_1^I) \cdot \sum_{B_0^I} Pr(B_0^I|e_1^I, J) \cdot Pr(f_1^J|e_1^I, B_0^I, J) && (3.49) \end{aligned}$$

Dies entspricht soweit der üblichen Zerlegung in Gleichung (2.15), es wird nur das inverse Alignment B_0^I verwendet. Diesmal wird jedoch die Lexikonwahrscheinlichkeit $Pr(f_1^J|e_1^I, B_0^I, J)$ kontextabhängig modelliert:

$$\begin{aligned} Pr(f_1^J|e_1^I, B_0^I, J) &= \prod_{i=0}^{I} \prod_{j \in B_i} Pr\left(f_j|e_1^I, B_0^I, f_{B_0}^{B_{i-1}}, f_{B_{i,<j}}, J\right) && (3.50)\\ &= \prod_{i=0}^{I} \prod_{j \in B_i} p\left(f_j|e_1^i, B_1^i, f_{B_1}^{B_{i-1}}, f_{B_{i,<j}}\right) && (3.51) \end{aligned}$$

In Gleichung (3.50) wurde die Lexikonwahrscheinlichkeit faktorisiert. Im zweiten Schritt wurden Modellannahmen gemacht. Die Abhängigkeit wurde folgendermaßen eingeschränkt:

- Die Wörter des Zielsatzes sind nur bis zur aktuellen Position i relevant.
- Das Alignment ist ebenfalls nur bis zur aktuellen Position i relevant.
- Die Satzlänge J sowie die nicht alignierten Wörter des Quellsatzes sind nicht relevant.

Im nächsten Schritt werden nun die zweidimensionalen n-Gramme eingesetzt. Das heißt, die Abhängigkeit wird entsprechend eingeschränkt. Somit erhält man:

$$Pr(f_1^J|e_1^I, B_0^I, J) = \prod_{i=0}^{I} \prod_{j \in B_i} p\left(f_j|e_{i-n+1}^i, B_{i-n+1}^i, f_{B_{i-n+1}}^{B_{i-1}}, f_{B_{i,<j}}\right) \quad (3.52)$$

3.3.4 Kombination mit der phrasenbasierten Übersetzung

In diesem Abschnitt wird beschrieben, wie man die phrasenbasierte Übersetzung aus Abschnitt 3.1 mit den in diesem Abschnitt beschriebenen zweidimensionalen n-Grammen verbinden kann. Es wird eine log-lineare Interpolation der beiden Modelle durchgeführt. Hierbei wird der Skalierungsfaktor λ_1 für das Übersetzungsmodell mit den zweidimensionalen n-Grammen verwendet und λ_2 für das phrasenbasierte Übersetzungsmodell. Dies ist der in Abschnitt 3.1.4 beschriebenen Glättung mit dem Lexikon sehr ähnlich. Hier kommt allerdings ein kontextabhängiges Modell zum Einsatz. Man hat nun wieder zwei *hidden variables*: die Segmentierung S und das Alignment B_0^I. Die Übersetzungswahrscheinlichkeit wird nun wie folgt modelliert:

$$Pr(f_1^J|e_1^I) = \sum_{B_0^I} Pr(f_1^J, B_0^I|e_1^I)^{\lambda_1} \cdot \sum_S Pr(f_1^J, S|e_1^I)^{\lambda_2} \quad (3.53)$$

$$= \sum_{B_0^I} \left(\prod_{i=0}^{I} \prod_{j \in B_i} p\left(f_j|e_{i-n+1}^i, B_{i-n+1}^i, f_{B_{i-n+1}}^{B_{i-1}}, f_{B_{i,<j}}\right)^{\lambda_1} \right)$$

$$\cdot \sum_S \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k})^{\lambda_2} \quad (3.54)$$

Im zweiten Schritt wurden die jeweiligen Formel eingesetzt und im letzten Schritt wurden die Modellannahmen gemacht. Hierbei wurden die Alignmentwahrscheinlichkeit $Pr(B_0^I|e_1^I)$ und die Segmentierungswahrscheinlichkeit $Pr(S|e_1^I)$ weggelassen, da diese nicht modelliert werden.

Diese Variante wird auf dem VERBMOBIL Korpus sowohl für die monotone Suche (Abschnitt 4.1) als auch für die Suche auf dem Umordnungsgraphen (Abschnitt 4.2.1) sehr gute Resultate erzielen (siehe Abschnitt 5.3).

Kapitel 4

Suche

In diesem Kapitel werden verschiedene Suchverfahren vorgestellt. Sie basieren alle auf Dynamischer Programmierung. Zunächst wird in Abschnitt 4.1 ein monotones Verfahren präsentiert welches sehr effizient ist. Für Sprachpaare, deren Wortreihenfolge nahezu gleich ist, wird dieses Verfahren bereits sehr gute Ergebnisse erzielen. Beispiele für solche Sprachpaare sind Italienisch-Englisch und insbesondere Spanisch-Katalanisch. Die Ergebnisse hierzu werden in den Abschnitten 5.4 und 5.5 dargestellt. Es gibt jedoch auch Sprachpaare, wie zum Beispiel Deutsch-Englisch, bei denen die Wortreihenfolge stark von einander abweichen kann. Es werden zwei nicht-monotone Verfahren präsentiert: das erste in Abschnitt 4.2.1 basiert auf einem Umordnungsgraphen und das zweite in Abschnitt 4.2.2 auf Pattern-Pairs.

Bei den Suchverfahren wird typischerweise eine *Rekombination* durchgeführt. Das heißt, falls zwei Hypothesen bezüglich zukünftiger Entscheidungen nicht unterschieden werden können, so wird nur diejenige mit der höheren Wahrscheinlichkeit fortgesetzt. Diejenige mit der geringeren Wahrscheinlichkeit braucht nicht weiter betrachtet zu werden, da sie nicht zu einem globalen Maximum fortgesetzt werden kann. Denn die gleiche Fortsetzung bei der momentan besseren Hypothese führt zu einer insgesamt höheren Wahrscheinlichkeit.

Unbekannte Wörter, also Wörter die im Training nicht gesehen wurden, werden durch sich selbst übersetzt. Da dieses Phänomen häufig bei Namen auftritt, ist diese Heuristik recht brauchbar.

4.1 Monotone Suche

Bei der monotonen Suche wird davon ausgegangen, dass die erste Phrase der Übersetzung $\tilde{e}_1$ von der ersten Phrase des Quellsatzes $\tilde{f}_1$ erzeugt wird, die zweite Phrase $\tilde{e}_2$ von der zweiten Phrase des Quellsatzes $\tilde{f}_2$ und so weiter.

Das monotone Suchproblem kann mit Dynamischer Programmierung effizient gelöst werden. Es werden dabei für jede Position j die Übersetzungshypothesen von f_1^j bestimmt. Diese entstehen, indem man kürzere Hypothesen um eine Phrase erweitert. Bei der Suche mit Maximum Approximation kann die sogenannte Sprachmodellrekombination durchgeführt werden. Das heißt, an einer Position j können alle Hypothesen mit der gleichen Sprachmodellhistorie rekombiniert werden. Für das Suchproblem mit Maximum Approximation (3.12) wird die Hilfsgröße $Q_M(j,h)$ definiert als die maximale Wahrscheinlichkeit einer Phrasenfolge, die die Positionen 1 bis j des Quellsatzes abdeckt und mit Sprachmodellhistorie h endet. $Q_M(J+1,\$)$ ist die Wahrscheinlichkeit der optimalen Übersetzung. Das \$ Symbol ist eine Satzgrenzenmarkierung. Um eine Hypothese zu beenden, muss die vom Sprachmodell gelieferte Satzendewahrscheinlichkeit $p(\$|h)$ angewendet werden. Damit kann man nun die folgenden Rekursionsgleichungen für die Dynamische Programmierung aufstellen:

$$Q_M(0,\$) = 1 \tag{4.1}$$

$$Q_M(j,h) = \max_{\substack{0\leq j'<j,\\ h',\tilde{e}:h=h'\oplus\tilde{e}}} Q_M(j',h')\cdot p(f_{j'+1}^{j}|\tilde{e})^{\lambda}\cdot p(\tilde{e}|h') \tag{4.2}$$

$$Q_M(J+1,\$) = \max_{h} Q_M(J,h)\cdot p(\$|h) \tag{4.3}$$

In Gleichung (4.3) wird die Nebenbedingung $h = h' \oplus \tilde{e}$ gestellt. Dies bezeichne die Sprachmodellhistorie h, die entsteht, wenn man an die Historie h' die Phrase $\tilde{e}$ anhängt. Während der Suche werden Backpointer zu dem jeweils maximierenden Argument gespeichert. Nachdem die Suche durchgeführt wurde kann somit die optimale Übersetzung erzeugt werden. Auf diese Methode wird sich später mit **MonMax** bezogen.

Der Suchalgorithmus `MonSearch` für die Maximum Approximation ist in Abbildung 4.1 dargestellt. Dieser Algorithmus ergibt sich direkt aus den Rekursionsgleichungen. Er bestimmt die Wahrscheinlichkeit der besten Hypothese. Die Funktion `Recombine` entspricht der Sprachmodellrekombination, die Funktion `Prune` dem Pruning. Dieses wird in Abschnitt 4.3 beschrieben.

Für das Suchproblem mit Summenkriterium (3.11) wird nun analog die Hilfsgröße $Q_S(j,e_1^i)$ definiert. Dies ist die maximale Wahrscheinlichkeit einer Phrasenfolge, die die Positionen 1 bis j des Quellsatzes abdeckt und die

Abbildung 4.1: Algorithmus `MonSearch` zur monotonen phrasenbasierten Suche.

```
INPUT: f_1^J
Q(0) = {($, 1)}
FOR j = 1 TO J DO
    FOR j' = 0 TO j − 1 DO
        E = { ẽ | p(f_{j'+1}^j | ẽ) > 0 }
        FOR ALL (h', q') ∈ Q(j') DO
            FOR ALL ẽ ∈ E DO
                q = q' · p(f_{j'+1}^j | ẽ) · p(ẽ | h')
                h = h' ⊕ ẽ
                Q(j) = Q(j) ∪ {(h, q)}
    Recombine(Q(j))
    Prune(Q(j))
bestq = 0
FOR ALL (h', q') ∈ Q(J) DO
    q = q' · p($ | h')
    IF bestq < q THEN bestq = q
OUTPUT bestq
```

Übersetzung e_1^i liefert. Für die Summenbildung ist es erforderlich, dass die vollständige Übersetzungshypothese e_1^i bekannt ist. Es kann deshalb keine Sprachmodellrekombination durchgeführt werden. $Q_S(J+1, \hat{e}_1^I)$ ist die Wahrscheinlichkeit der optimalen Übersetzung $\hat{e}_1^I$.

$$Q_S(0, \$) = 1 \tag{4.4}$$

$$Q_S(j, e_1^i) = \sum_{\substack{0 \leq j' < j \\ 0 \leq i' < i}} Q_S(j', e_1^{i'}) \cdot p(f_{j'+1}^j | e_{i'+1}^i)^\lambda \cdot p(e_{i'+1}^i | e_1^{i'}) \tag{4.5}$$

$$Q_S(J+1, \hat{e}_1^I) = \max_{e_1^I} Q_S(J, e_1^I) \cdot p(\$ | e_I) \tag{4.6}$$

Auf diese Methode wird sich später mit **MonSum** bezogen. Viele statistischen Übersetzungssysteme verwenden die Maximum Approximation. In Kapitel 5 wird jedoch gezeigt, dass das Summenkriterium typischerweise bessere Ergebnisse erzielt. Andererseits ist die Suche mit dem Summenkriterium deutlich aufwendiger, da keine Sprachmodellrekombination durchgeführt werden kann. Die Suche für die in Abschnitt 3.1.4 vorgestellten Varianten der phrasenbasierten Übersetzung wird analog zu den hier präsentierten Methoden durchgeführt.

Die hier vorgestellte Suche mit Maximum Approximation besitzt eine Komplexität von $O(J^2 \cdot {V_e}^2 \cdot |\{\tilde{e}\}|)$ im Worst-Case bei Verwendung eines Trigramm-Sprachmodells. Dabei ist J die Satzlänge des Quellsatzes, V_e die Vokabulargröße der Zielsprache und $|\{\tilde{e}\}|$ ist die Anzahl der Phrasen der Zielsprache. Der Faktor ${V_e}^2$ entspricht der Anzahl der möglichen Sprachmodellhistorien h.

Die hier vorgestellte Suche mit dem Summenkriterium besitzt eine Komplexität von $O(J^2 \cdot {V_e}^I \cdot |\{\tilde{e}\}|)$ im Worst-Case. Da bei dem Summenkriterium keine Sprachmodellrekombination durchgeführt werden kann, ergibt sich hier der Faktor ${V_e}^I$.

4.2 Nicht-Monotone Suche

Bei der Analyse der monotonen Übersetzungen stellt man fest, dass viele Übersetzungsfehler auf die Monotonie zurückzuführen sind. Daher werden in diesem Abschnitt zwei Methoden beschrieben nicht-monotone Übersetzungen zu erzeugen. Das Problem bei nicht-monotonen Übersetzungen ist, dass sich der Suchraum exponentiell vergrößert. Selbst bei einer vorgegebenen Segmentierung eines Satzes in K Phrasen gibt es $K!$ Möglichkeiten die Phrasen der Zielsprache anzuordnen. Werden lange Phrasen verwendet, so ist dies im

Vergleich zu den $J!$ Möglichkeiten auf Wortebene aber schon eine deutliche Reduktion. Bei der ersten Methode wird die Anzahl der möglichen Umordnungen durch einen Umordnungsgraphen eingeschränkt. Die zweite Methode basiert auf Pattern-Pairs. Diese beschreiben explizit Umstellungen von Phrasen. Ausgehend von einer monotonen Übersetzung werden hier Phrasen umgeordnet, um so zu einer nicht-monotonen Übersetzung zu gelangen.

4.2.1 Umordnungsgraph

Im Allgemeinen beschreibt ein Umordnungsgraph, welche Umordnungen der Wörter des Quellsatzes möglich sind. Einem Pfad durch diesen Graphen entspricht eine Permutation π der Zahlen 1 bis J und somit eine Umordnung des Quellsatzes in $f_{\pi(1)}f_{\pi(2)}...f_{\pi(J)}$.

Hier wird ein Umordnungsgraph erzeugt, indem man von einem Wortgraphen der einzelwortbasierten Suche überflüssige Informationen entfernt. Außerdem werden äquivalente Knoten, dies sind Knoten mit dem gleichen Coverage-Vektor (siehe unten), zusammengefasst. Ein derartiger Umordnungsgraph schränkt die möglichen Umordnungen prinzipiell auf die Umordnungen der einzelwortbasierten Suche ein. Hierbei werden nur die vielversprechenden Umordnungen übernommen, da die einzelwortbasierte Suche ein Pruning des Wortgraphen durchführt.

Definition des Umordnungsgraphen

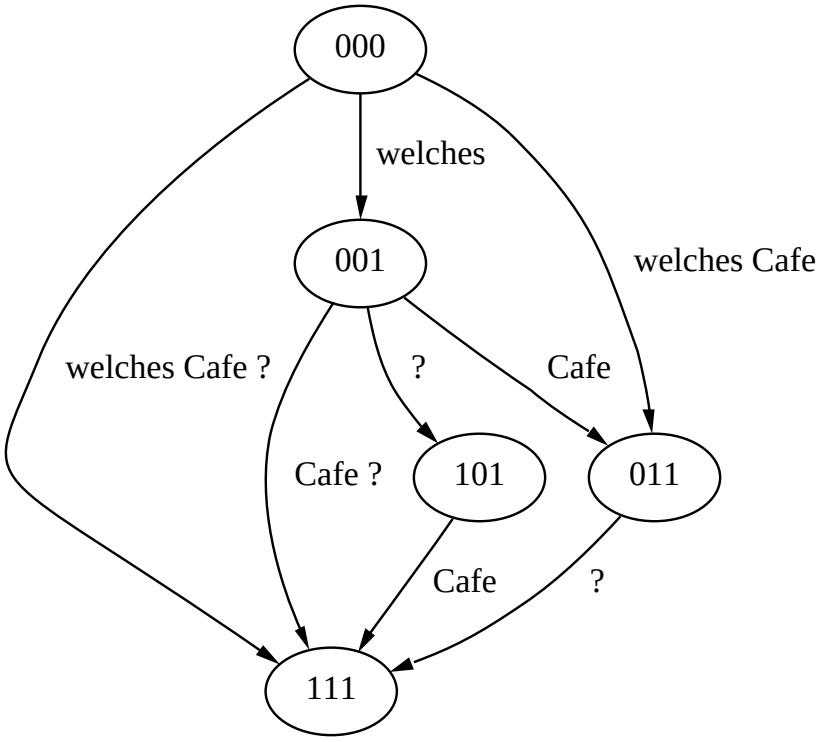
Ein Umordnungsgraph ist ein gerichteter azyklischer Graph mit einem Start- und einem Zielknoten. Die Knoten sind von 0 (Start) bis N (Ziel) nummeriert. Diese Nummerierung muss einer topologischen Sortierung des Graphen entsprechen. Jeder Knoten ist mit einem Coverage-Vektor markiert. Dies ist ein Bitvektor b der Größe J (der Länge des Quellsatzes) mit der Eigenschaft: $b[j] = 1$ genau dann wenn Position j des Quellsatzes bereits übersetzt wurde. Zusätzlich gilt, dass der Coverage-Vektor eines jeden Knoten sich von den Coverage-Vektoren der direkten Vorgängerknoten je um genau ein Bit unterscheidet.

Der Startknoten wird mit 0^J und der Zielknoten mit 1^J markiert. $Pred(n)$ wird definiert als die Menge aller direkten und indirekten Vorgängerknoten des Knoten n. Das Symbol $\tilde{F}(n_1, n_2)$ wird definiert als die Folge von Wörtern des Quellsatzes die vom Knoten n_2 abgedeckt wird, aber nicht vom Knoten n_1. Diese Wortfolge hat die gleiche Reihenfolge wie die Wörter im Quellsatz. Sie ist also unabhängig von der Reihenfolge, in der die Wörter auf einem Pfad von n_1 nach n_2 abgedeckt werden. Diese Definition wurde gewählt,

weil Umordnungen innerhalb einer bilingualen Phrasen $(\tilde{f}, \tilde{e})$ bereits erlaubt sind und aus dem Trainingskorpus gelernt werden. Wichtig sind also die Umordnungen ganzer Phrasen. Dies wird durch diese Definition erreicht.

Ein Beispiel eines Umordnungsgraphen für den Satz "*welches Cafe ?*" ist in Abbildung 4.2 zu sehen. Hierbei wurden die Kanten der Lesbarkeit halber mit den zugehörigen Phrasen beschriftet. Außerdem wurde die Nummerierung weggelassen. Die Coverage-Vektoren sind in jeden Knoten eingetragen. In diesem Beispiel sind Umordnungen eigentlich nicht erforderlich. Es dient der Illustration eines Umordnungsgraphen. Bei längeren Sätzen, in denen Umordnungen Sinn machen, ist der Umordnungsgraph typischerweise zu groß, um ihn sinnvoll darstellen zu können. Prinzipiell wächst der Umordnungsgraph exponentiell mit der Länge des Quellsatzes J. Da es 2^J mögliche Coverage-Vektoren gibt, kann es auch dem entsprechend viele Knoten in dem Umordnungsgraphen geben.

Abbildung 4.2: Umordnungsgraph zu "*welches Cafe ?*".



Suche auf einem Umordnungsgraphen

Die Suche auf einem Umordnungsgraphen kann mit Dynamischer Programmierung durchgeführt werden. Die Idee ist ähnlich wie bei der monotonen Suche. Hier durchläuft man jedoch die Knoten des Umordnungsgraphen von 0 bis N anstelle der Positionen des Quellsatzes von 0 bis J. Wenn man einen Knoten besucht, so wird durch die topologische Sortierung garantiert, dass alle Vorgängerknoten bereits besucht wurden. Für das Suchkriterium mit Maximum Approximation wird die Hilfsgröße $Q_M(n,h)$ definiert als die maximale Wahrscheinlichkeit einer Phrasenfolge, welche $\tilde{F}(0,n)$ übersetzt und in der Sprachmodellhistorie h endet. Die Rekursionsgleichungen lauten:

$$
\begin{aligned}
Q_M(0,\$) &= 1 && (4.7)\\
Q_M(n,h) &= \max_{\substack{n' \in Pred(n),\\ h',\tilde{e}: h = h' \oplus \tilde{e}}} Q_M(n',h') \cdot p(\tilde{F}(n',n)|\tilde{e})^{\lambda} \cdot p(\tilde{e}|h') && (4.8)\\
Q_M(N+1,\$) &= \max_{h} Q(N,h) \cdot p(\$|h) && (4.9)
\end{aligned}
$$

Auf diese Methode wird sich später mit **ExtMax** bezogen. Für das Summenkriterium sehen die Gleichungen analog aus. Die Hilfsgröße $Q_S(n,e_1^i)$ wird definiert als die maximale Wahrscheinlichkeit einer Phrasenfolge, die $\tilde{F}(0,n)$ abdeckt und die Übersetzung e_1^i liefert. Die Rekursionsgleichungen für diese Methode, auf die sich später mit **ExtSum** bezogen wird, lauten:

$$
\begin{aligned}
Q_S(0,\$) &= 1 && (4.10)\\
Q_S(n,e_1^i) &= \sum_{\substack{n' \in Pred(n),\\ 0 \le i' < i}} Q_S(n',e_1^{i'}) \cdot p(\tilde{F}(n',n)|e_{i'+1}^i)^{\lambda} \cdot p(e_{i'+1}^i|e_1^{i'}) && (4.11)\\
Q_S(N+1,\hat{e}_1^I) &= \max_{e_1^I} Q(N,e_1^I) \cdot p(\$|e_I) && (4.12)
\end{aligned}
$$

Die monotone Suche kann als Spezialfall der Suche auf einem Umordnungsgraphen aufgefasst werden. Dazu definiert man den Umordnungsgraphen für einen Satz der Länge J wie folgt: Es gibt $J+1$ Knoten, nummeriert von 0 bis J. Der Knoten j hat den Coverage-Vektor $1^j 0^{J-j}$. Es gibt J Kanten und zwar $(j, j+1)$ für alle $0 \le j < J$. Die Suche auf einem solchen Umordnungsgraphen ist identisch mit der monotonen Suche, wie sie in Abschnitt 4.1 beschrieben ist.

Die hier vorgestellte Suche auf dem Umordnungsgraphen mit Maximum Approximation besitzt eine Komplexität von $O(N^2 \cdot {V_e}^2 \cdot |\{\tilde{e}\}|)$ im Worst-Case bei Verwendung eines Trigramm-Sprachmodells. Dabei ist N die Anzahl der Knoten des Umordnungsgraphen. Die Suche mit dem Summenkriterium besitzt eine Komplexität von $O(N^2 \cdot {V_e}^I \cdot |\{\tilde{e}\}|)$ im Worst-Case.

4.2.2 Pattern-Pairs

Das zweite nicht-monotone Verfahren basiert auf den in Abschnitt 3.2 vorgestellten Pattern-Pairs. Bei dieser Methode werden, ausgehend von einer monotonen Übersetzung, Phrasen umgeordnet, um so zu einer nicht-monotonen Übersetzung zu gelangen. Es wird explizit ein Suchgraph aufgebaut. Bei den beiden bisherigen Verfahren war dieser Suchgraph nur implizit vorhanden.[1]

Die Suche für die Pattern-Pairs läuft folgendermaßen ab:

1. Erzeuge den monotonen, phrasenbasierten Suchgraphen.

2. Wende wiederholt die Pattern-Pairs auf den Graphen an.

3. Extrahiere die beste Hypothese aus dem Graphen.

Monotoner phrasenbasierter Suchgraph

Der Suchgraph ist gerichtet und azyklisch. Er hat für jede Position j des Quellsatzes einen Knoten, sowie einen Endknoten. Es sind also $J+1$ Knoten, nummeriert von 1 bis $J+1$. Für eine Phrase $f_{j_1}^{j_2}$ werden nun Kanten in den Graphen eingetragen. Es wird für jede mögliche Übersetzung $\tilde{e}$ dieser Quellphrase $f_{j_1}^{j_2}$ eine Kante (j_1, j_2+1) in den Graphen eingefügt. Diese wird beschriftet mit der Phrasenübersetzung $\tilde{e}$ und der Wahrscheinlichkeit $p(f_{j_1}^{j_2}|\tilde{e})$. So wird für alle Phrasen $f_{j_1}^{j_2}$ des Quellsatzes verfahren. Durch die Positionen des Quellsatzes (j_1, j_2) ist eindeutig die Klasse der Kanten identifiziert, welche $f_{j_1}^{j_2}$ übersetzt. Dies sind genau die Kanten von Knoten j_1 nach Knoten j_2+1.

Jedem Pfad von Knoten 1 bis Knoten $J+1$ entspricht eine monotone, phrasenbasierte Übersetzungshypothese des Quellsatzes. Die Übersetzungswahrscheinlichkeit dieser Hypothese entspricht dem Produkt der Wahrscheinlichkeiten an den Kanten entlang des Pfades.

[1] Bemerkung: Der Suchgraph sollte nicht mit dem Umordnungsgraphen verwechselt werden.

Anwendung der Pattern-Pairs

In diesen Graphen werden nun durch die Anwendung von Pattern-Pairs zusätzliche Kanten eingetragen. Ein Pattern-Pair $(\bar{f}|\bar{e}) = ((\tilde{f}_1 V \tilde{f}_2), (\tilde{e}_1 V \tilde{e}_2))$ kann auf eine Kantenklasse (j_2, j_3) des Graphen angewendet werden, falls es zwei Positionen des Quellsatzes j_1 und j_4 gibt, für die folgendes gilt:

$$\tilde{f}_1 = f_{j_1}^{j_2} \quad \wedge \quad \tilde{f}_2 = f_{j_3+1}^{j_4} \tag{4.13}$$

Hierbei wird natürlich $j_1 \leq j_2 \leq j_3 \leq j_4$ vorrausgesetzt. Trifft dies zu, so wird für jede Kante dieser Klasse eine neue Kante in den Graphen eingetragen. Eine solche neue Kante führt von Knoten j_1 nach Knoten $j_4 + 1$, sie repräsentiert also eine Übersetzung von $f_{j_1}^{j_4}$. Die Beschriftung der neuen Kante ergibt sich, indem man den Platzhalter des Pattern-Pairs durch die Beschriftung der alten Kante ersetzt. Falls die alte Kante die Beschriftung $\tilde{e}$ hat, so erhält die neue Kante die Beschriftung $(\tilde{e}_1 \tilde{e} \tilde{e}_2)$. Des weiteren ist die Wahrscheinlichkeit dieser neuen Kante das Produkt aus der Pattern-Pair Wahrscheinlichkeit $p(\bar{f}|\bar{e})$ und der Wahrscheinlichkeit p der alten Kante, also $p(\bar{f}|\bar{e}) \cdot p$.

Extraktion der besten Hypothese

Zur Extraktion der besten Hypothese wird auf dem erzeugten Suchgraphen eine monotone Suche durchgeführt. Der Algorithmus ist nahezu identisch zu dem Algorithmus für die monotone Suche in Abbildung 4.1. Der einzige Unterschied ist, dass die Menge E der Hypothesen aus dem Suchgraphen entnommen wird, ebenso die Phrasenwahrscheinlichkeiten für diese Hypothesen. Die Sprachmodellwahrscheinlichkeit muss wie bei der monotonen Suche noch hinzugerechnet werden.

Ausführliches Beispiel

In diesem Abschnitt wird ein Beispiel einer Übersetzung mittels Pattern-Pairs detaillierter beschrieben. Das Beispiel (Abbildung 4.3) zeigt die Übersetzung des Satzes '*wann wollen wir nach Hamburg zurück reisen ?*'.

In der ersten Zeile ist der Quellsatz mit seiner Segmentierung in Phrasen dargestellt. Die Zeile L0 (Level 0) enthält die entsprechende phrasenbasierte Übersetzung ohne Anwendung von Pattern-Pairs. Dies entspricht einer monotonen phrasenbasierten Übersetzung.

Für die nächste Zeile L1 (Level 1) wurde die einmalige Anwendung von Pattern-Pairs erlaubt. Hierbei heißt einmalig nicht, das nur eine einzige Anwendung eines Pattern-Pairs erlaubt wäre, sondern dass die Pattern-Pairs

nicht auf das Ergebnis der Anwendung eines Pattern-Pairs angewendet werden dürfen. Es wird das Pattern-Pair '*V reisen* → *travel V*' angewendet. Hierbei wird *V* durch die bilinguale Phrase (*zurück*,*back*) ersetzt. Auf den Rest des Satzes wurden keine Pattern-Pairs angewendet. Es bleiben also die normalen Phrasenübersetzungen erhalten.

In Zeile L2 (Level 2) wurde die zweimalige Anwendung von Pattern-Pairs erlaubt. Hier ist es also erlaubt, dass ein Pattern-Pair auf das Ergebnis der Anwendung eines Pattern-Pairs angewendet wird. Es wird das Pattern-Pair '*nach Hamburg V* → *V to Hamburg*' benutzt. Für den Platzhalter *V* wird hierbei das Ergebnis der Anwendung des Pattern-Pairs aus Zeile L1 eingesetzt. Wie man sieht, wird dabei die ursprüngliche Phrasenübersetzung '*go to Hamburg*' ignoriert.

Zusätzliche hierarchische Anwendung von Pattern-Pairs (Level 3 und 4) bringen keine Veränderung mehr. Daher ergibt sich die Übersetzung des Quellsatzes als '*when do we want to travel back to Hamburg ?*'. Im Gegensatz zu der monotonen Übersetzung wurde hier die Verbphrase korrekt übersetzt.

Abbildung 4.3: Beispiel für die Anwendung von Pattern-Pairs

	wann	wollen wir	nach Hamburg	zurück	reisen	?
L0	when	do we want to	go to Hamburg	back	travel	?
				V reisen → travel V		
L1				travel back		
			nach Hamburg V → V to Hamburg			
L2			travel back to Hamburg			

4.3 Pruning

Zur Beschleunigung der Suche sowie zur Reduzierung des Speicherplatzbedarfes werden Pruningtechniken angewendet. Durch diese Methoden können jedoch Suchfehler auftreten, das heißt es wird möglicherweise nicht die global beste Hypothese gefunden. Ferner wird durch die Anwendung des Pruning das Summenkriterium nur näherungsweise erfüllt. Dies liegt daran, dass Beiträge von Erweiterungen einer Hypothese, welche aus dem Suchstrahl entfernt wurde, verloren gehen.

Es werden zwei Pruningtechniken angewendet: Threshold Pruning und Histogramm Pruning. Beim Threshold Pruning werden alle Hypothesen eliminiert, die relativ zur bisher besten Hypothese eine kleine Wahrscheinlichkeit haben. Wie stark dieses Pruning ausfallen soll, wird über den sogenannten Threshold Parameter q_t mit $0 \leq q_t \leq 1$ eingestellt. Sei H_0 die aktuell beste Hypothese, so wird eine Hypothese H genau dann eliminiert, wenn gilt:

$$Q(H) < q_t \cdot Q(H_0) \tag{4.14}$$

Um die Anzahl der Hypothesen zu beschränken wird Histogramm Pruning durchgeführt. Die Idee hierbei ist, dass nur die m besten Hypothesen weiterverfolgt werden. Alle anderen werden eliminiert.

Diese beiden Techniken können an verschiedenen Stellen angewendet werden:

- **Observation Pruning.** Hier werden die möglichen Übersetzungen der Phrasen eingeschränkt. Das heißt, es werden nicht alle Kandidaten die im Training vorkamen benutzt, sondern nur diejenigen welche dieses Pruning überstehen. Das Observation Pruning muss nur einmal während des Ladens des Übersetzungsmodells durchgeführt werden. In den Experimenten wurde ein Observation Threshold von 0.05 gewählt. Die Anzahl der Übersetzungskandidaten pro Phrase wurde auf 100 eingeschränkt.

- **Coverage Pruning.** Dieses Pruning wird während der Suche durchgeführt. Es werden jeweils alle Hypothesen an einer bestimmten Position j beziehungsweise einem Knoten n verglichen. Das heißt, immer diejenigen Hypothesen mit gleichem Coverage-Vektor. In den Experimenten wurde ein Coverage Threshold von 0.0005 gewählt. Die Anzahl der Übersetzungshypothesen pro Position j beziehungsweise Knoten n wurde auf 1000 eingeschränkt. Dies entspricht einem sehr konservativem Pruning. Es entstanden hierdurch keine Suchfehler.

- **Cardinality Pruning.** Es wird auf Hypothesen angewendet welche die gleiche Anzahl von Wörtern übersetzen. Dieses Pruning ist nur für die Suche auf dem Umordnungsgraphen sinnvoll. Hier werden auch Hypothesen von verschiedenen Knoten des Umordnungsgraphen mit einander verglichen. Es wird hier nur der Vollstängigkeit halber erwähnt. In den Experimenten wurde dieses Pruning nicht eingesetzt.

In Abschnitt 5.6 werden die Effekte des Prunings bezüglich der Übersetzungszeit und der Suchfehler untersucht. Es wird sich zeigen, dass durch Pruning die Rechenzeit deutlich gesenkt werden kann ohne Suchfehler zu produzieren.

Kapitel 5

Ergebnisse

Mit den vorgestellten Verfahren wurden umfangreiche Experimente durchgeführt. Die Resultate werden in diesem Kapitel beschrieben. Zunächst wird das Training in Abschnitt 5.1 beschrieben. In Abschnitt 5.2 werden die verwendeten Testkriterien vorgestellt. Dann werden die Ergebnisse für die Korpora VERBMOBIL (Deutsch-Englisch), FUB (Italienisch-Englisch) und El Periodico (Spanisch-Katalanisch) präsentiert.

5.1 Training

Das Training der Übersetzungsmodelle ist für die verschiedenen Korpora weitgehend identisch. Allein die Vorverarbeitung ist korpusspezifisch. Das Training wird ausgehend von einem parallelen Text vollautomatisch durchgeführt. Zunächst wird für diesen Text die Vorverarbeitung duchgeführt. Anschließend werden einzelwortbasierte Wortalignments trainiert. Dies sind die in Abschnitt 2.6 vorgestellten Alignmentmodelle sowie das IBM4 Modell. Das geschieht mit dem Tool GIZA++ ([Och & Ney 00]). Dieses Training der Modelle geschieht iterativ. Das heißt, das Alignment eines einfachen Modells dient als Initialisierung eines komplizierteren Modells. Es wurden jeweils 5 Iterationen folgender Modelle durchgeführt: IBM1, HMM, IBM3 und schließlich IBM4. In der üblichen Schreibweise hierfür entspricht dies $1^5H^53^54^5$.

Die Alignments werden für beide Übersetzungsrichtungen (Quellsprache → Zielsprache, Zielsprache → Quellsprache) trainiert. Diese beiden Alignments werden nun symmetrisiert, indem die Vereinigung beider Alignments gebildet wird. Bis hierhin ist das Training identisch mit dem Training für den Alignment Template Ansatz. Mit Hilfe des Alignments werden die bilingualen Phrasen, die zweidimensionalen n-Gramme und die Pattern-Pairs erzeugt,

sowie deren Häufigkeiten bestimmt.

Das grundlegende Übersetzungssystem ist nun trainiert. Es fehlt noch die Einstellung der Skalierungsfaktoren. Diese werden durch eine Heuristik so gewählt, dass die mWER (siehe Abschnitt 5.2) auf dem Developmentkorpus minimal wird. Hierbei wird allerdings nur ein lokales Minimum gefunden.

Diese Heuristik wird nun beschrieben. Es handelt sich um ein iteratives Verfahren. Um einen Parameter zu optimieren, werden zunächst in einem vorgegebenen Intervall für fünf äquidistante Parameterwerte Übersetzungen durchgeführt und die Fehlerraten bestimmt. Das Intervall für die nächste Iteration ist nun der Bereich um das Minimum der Fehlerrate herum, also ab dem Parameterwert links von dem Minimum bis zu dem Parameterwert rechts davon. Hierbei wird die Genauigkeit des Parameters eingeschränkt. Die Kurven in Abbildung 5.1 wurden mit dieser Heuristik bestimmt.

Sollen mehrere Skalierungsfaktoren optimiert werden, so wird das obige Verfahren nacheinander auf jeden dieser Faktoren angewendet. Dieses Vorgehen wird ebenfalls iteriert. Zum Beispiel wird bei der Glättung mit dem Lexikon zuerst das phrasenbasierte Übersetzungsmodell skaliert, dann das Lexikon, anschließend wieder das phrasenbasierte Übersetzungsmodell, daraufhin das Lexikon und so weiter.

Diese Heuristik ist für die Optimierung eines oder zweier Skalierungsfaktoren gut geeignet, bei mehreren läuft man jedoch Gefahr, nur ein schlechtes lokales Optimum zu finden.

Bei einem derartigen Vorgehen ist es von besonderer Wichtigkeit, die bereits berechneten Fehlerraten in einer Tabelle zu speichern. Bei Bedarf können diese dann aus dieser Tabelle abgelesen werden und müssen nicht neu berechnet werden.

5.2 Testkriterien

In der maschinellen Übersetzung gibt es bisher kein allgemein akzeptiertes Kriterium für die Bewertung von Übersetzungen. Daher werden hier verschiedene Kriterien verwendet.

- WER (*word error rate*):
  Die Editierdistanz, auch Levenstheindistanz genannt, ist die minimale Anzahl von Ersetzungen, Einfügungen und Löschungen die notwendig sind um die erzeugte Übersetzung in die Referenzübersetzung zu transformieren. Die Wortfehlerrate entspricht der Editierdistanz pro Wort der Referenzübersetzung. Die WER ist in der Spracherkennung weit verbreitet.

- PER (*position-independent word error rate*):
  Ein Nachteil der WER ist, dass sie eine perfekte Wortreihenfolge verlangt. Die Wortreihenfolge einer akzeptablen Übersetzung kann jedoch von der Reihenfolge der Referenzübersetzung abweichen. Um diesem Problem entgegenzuwirken wird die PER verwendet. Hierbei werden die Wörter der beiden Sätze verglichen ohne die Wortreihenfolgen in Betracht zu ziehen.

- mWER (*multi-reference word error rate*):
  Ein weiterer Nachteil der WER ist, dass ein Satz mehrere mögliche Übersetzungen haben kann. Daher wird für die mWER nicht nur eine Referenzübersetzung verwendet, sondern mehrere. Es wird die Editierdistanz zu der ähnlichsten Referenz verwendet [Nießen & Och$^+$ 00].

- BLEU:
  BLEU wurde in [Papineni & Roukos$^+$ 01] vorgestellt. Es wird die Präzision von Unigrammen, Bigrammen, Trigrammen sowie von 4-Grammen bezüglich mehrerer Referenzübersetzungen gemessen. Außerdem gibt es eine Strafe für zu kurze Sätze. Bei BLEU handelt es sich nicht um eine Fehlerrate, wie bei den anderen Maßen. Daher bedeutet ein höherer BLEU Wert ein besseres Ergebnis.

- SSER (*subjective sentence error rate*):
  Bei der SSER handelt es sich um eine subjektive Bewertung der Übersetzungen. Jeder Satz wurde mit Kosten von 0.0 bis 1.0 bewertet. Kosten von 0.0 bedeuten, der Satz ist semantisch und syntaktisch korrekt, bei 0.5 ist der Satz zwar semantisch korrekt, weist jedoch syntaktische Fehler auf und bei 1.0 ist der Satz sowohl semantisch als auch syntaktisch falsch [Nießen & Och$^+$ 00].

- IER (*information item error rate*):
  Jeder Testsatz ist in Informationseinheiten eingeteilt. Für jede dieser Einheiten wird nun manuell bewertet, ob diese Information in dem Übersetzungskandidaten enthalten ist. Ist dies der Fall und ist der Satz syntaktisch fehlerfrei, so wird er als korrekt gezählt [Nießen & Och$^+$ 00].

Bei den ersten vier Kriterien (WER, PER, mWER, BLEU) handelt es sich um objektive Kriterien. Diese können automatisch berechnet werden. Die letzten beiden Kriterien (SSER, IER) sind subjektive Kriterien. Für diese müssen die Übersetzungshypothesen manuell bewertet werden. Die bewerteten Hypothesen werden in einer Datenbank abgespeichert, um für zukünftige

Auswertungen zur Verfügung zu stehen. Die subjektiven Fehlerraten können auch extrapoliert werden, falls nur ein Teil der Übersetzungshypothesen in der Datenbank enthalten ist. Für nähere Details bezüglich dieser Extrapolation siehe [Nießen & Och$^+$ 00].

Da in dieser Diplomarbeit sehr viele Experimente durchgeführt wurden, konnten nicht alle Übersetzungen manuell bewertet werden. Der maximale Anteil von extrapolierten Sätzen liegt bei etwa 30% und kommt bei den Pattern-Pairs vor. In den meisten Fällen ist der Anteil der extrapolierten Sätze jedoch kleiner als 10%.

5.3 Verbmobil

Während des VERBMOBIL Projektes wurde ein System entwickelt, welches gesprochene Spontansprache erkennt und übersetzt. Es handelt sich um Dialoge aus den Bereichen Terminabsprache, Reiseplanung und Hotelreservierung [Wahlster 00].

5.3.1 Korpus

Tabelle 5.1 zeigt die Korpusstatistiken der verwendeten Korpora. Mit Hilfe des Trainingskorpus wurde das Sprach- und Übersetzungsmodell trainiert. Zusätzlich fließt in das Übersetzungsmodell das manuelle Lexikon mit ein. Auf dem Developmentkorpus wurden die Skalierungsfaktoren bestimmt. Die Evaluierung wurde mittels des Testkorpus durchgeführt.

Tabelle 5.2 zeigt die Phrasenstatistik für das VERBMOBIL-Korpus. Es ist abhängig von der Länge der Quellphrase die Anzahl der bilingualen Phrasen sowie der Singletons (absolut und relativ) dargestellt. Die bilingualen Phrasen werden bezüglich ihrer Länge eingeschränkt. Es werden nur bilinguale Phrasen verwendet, deren Quellphrase nicht länger als 7 Wörter ist. Insgesamt gibt es rund 1.2 Millionen bilinguale Phrasen, davon sind 87.5% Singletons. Der relative Anteil der Singletons steigt mit der Phrasenlänge. Dies ist dadurch zu erklären, dass längere Phrasen im Trainingskorpus seltener auftreten.

Tabelle 5.3 zeigt die gleiche Statistik für die zweidimensionalen n-Gramme für das VERBMOBIL-Korpus. Es ist abhängig von der Länge deren Historie die Anzahl der zweidimensionalen n-Gramme sowie der Singletons (absolut und relativ) dargestellt. Es ist zu beachten, dass die Länge der Historie durch die Wörter der Zielphrase bestimmt wird. Daher stimmt die Anzahl der zweidimensionalen n-Gramme mit Historienlänge Eins nicht exakt mit der An-

Tabelle 5.1: VERBMOBIL Statistik: Trainingskorpus (Train, SZ=Satzzeichen), manuelles Lexikon (Lex), Developmentkorpus (Dev), Testkorpus (Test).

		Ohne Vorverarbeitung		Mit Vorverarbeitung	
		Deutsch	Englisch	Deutsch	Englisch
Train	Sätze	58 073			
	Wörter	519 523	549 921	522 933	548 874
	Wörter ohne SZ	418 979	453 632	421 689	456 629
	Singletons	3 453	1 698	3 570	1 763
	Vokabular	7 940	4 673	8 102	4 780
Lex	Einträge	12 779			
	Erw. Vokabular	11 501	6 867	11 904	7 089
Dev	Sätze	276			
	Wörter	3 159	3 438	3 172	3 445
	Trigramm PP	-	28.1	-	26.3
Test	Sätze	251			
	Wörter	2 628	2 871	2 640	2 862
	Trigramm PP	-	30.5	-	29.9

Tabelle 5.2: VERBMOBIL: Statistik der bilingualen Phrasen in Abhängigkeit von der Länge der Quellphrase.

		Singletons	
Länge	Anzahl	absolut	relativ (in %)
1	55 036	39 811	72.3
2	146 238	109 024	74.6
3	213 881	176 284	82.4
4	236 717	209 365	88.4
5	228 445	210 246	92.0
6	200 574	188 677	94.1
7	150 780	144 264	95.7
insgesamt	1 231 671	1 077 671	87.5

Tabelle 5.3: VERBMOBIL: Statistik der zweidimensionalen n-Gramme in Abhängigkeit von der Länge der Historie.

		Singletons	
Länge	Anzahl	absolut	relativ (in %)
1	55 522	39 233	70.7
2	169 168	125 184	74.0
3	295 015	240 048	81.4
4	349 509	308 635	88.4
5	353 254	327 756	92.8
6	327 962	312 398	95.3
7	292 473	282 235	96.5
insgesamt	1 842 903	1 635 489	88.7

zahl der bilingualen Phrasen mit Quellphrasenlänge Eins überein. Auch bei den zweidimensionalen n-Grammen werden nicht alle möglichen verwendet, sondern nur diejenigen der Historie nicht länger als 7 ist. Insgesamt gibt es rund 1.8 Millionen zweidimensionale n-Gramme, davon sind 88.7% Singletons. Auch hier steigt der relative Anteil der Singletons mit der Länge der Historie. Es gibt etwa 50% mehr zweidimensionale n-Gramme als es bilinguale Phrasen gibt. Ein Grund hierfür ist, dass bei den zweidimensionalen n-Grammen die Quellphrase auch Lücken haben darf. Dies ist bei bilingualen Phrasen nicht erlaubt.

In Tabelle 5.4 ist noch die Statistik für die Pattern-Pairs dargestellt. Wie man sieht, gibt es mit etwa 5.8 Millionen deutlich mehr Pattern-Pairs als bilinguale Phrase oder zweidimensionale n-Gramme. Die Zahl der Singletons ist mit etwa 87% ähnlich wie bei den bilingualen Phrasen oder den zweidimensionalen n-Grammen.

5.3.2 Monotone phrasenbasierte Übersetzung

Maximum Approximation und Summenkriterium

In Abbildung 5.1 wird die Maximum Approximation mit dem Summenkriterium verglichen. Die Kurven zeigen die mWER auf dem VERBMOBIL Developmentkorpus in Abhängigkeit von dem Übersetzungsmodellskalierungsfaktor für die Standardrichtung $p(f|e)$. Wie man sieht, verläuft die Kurve für das Summenkriterium überall unter der Kurve der Maximum Approximation.

Tabelle 5.4: VERBMOBIL: Statistik der Pattern-Pairs in Abhängigkeit von der Gesamtlänge der beiden Quellphrasen.

		Singletons	
Länge	Anzahl	absolut	relativ (in %)
1	128 470	83 728	65.2
2	665 329	490 048	73.7
3	1 334 191	1 096 774	82.2
4	1 596 421	1 409 875	88.3
5	1 402 234	1 312 064	93.6
6	736 669	702 711	95.4
insgesamt	5 863 314	5 095 200	86.9

Der Unterschied beträgt etwa $1.0 - 1.5\%$. Die Verwendung des Summenkriteriums verbessert also die Übersetzungsqualität.

In Tabelle 5.5 sind die Ergebnisse der Baselinevarianten der monotonen phrasenbasierten Übersetzung dargestellt. Man sieht, dass die Varianten mit Summenkriterium auch auf dem Testkorpus bessere Ergebnisse liefern als jene mit Maximum Approximation. Ferner kann man beobachten, dass die inverse Richtung $p(e|f)$ hier bessere Resultate erzielt als die Standardrichtung $p(f|e)$.

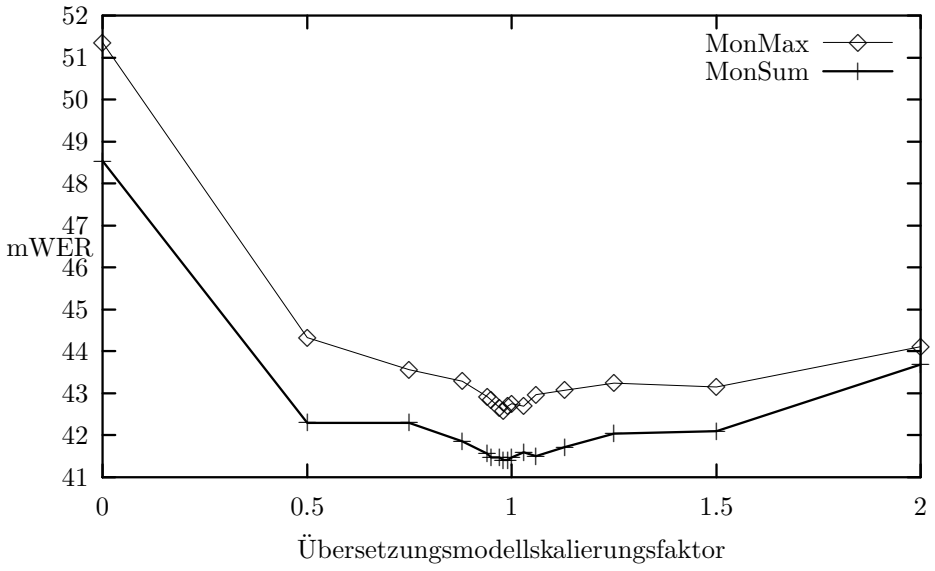
Tabelle 5.5: VERBMOBIL: Baseline Varianten für monotone phrasenbasierte Übersetzung.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	MonMax	46.9	33.3	42.0	42.1	40.8	40.2
	MonSum	45.9	32.2	40.9	43.4	39.7	40.0
$p(e\|f)$	MonMax	43.8	30.8	38.8	43.6	39.1	40.5
	MonSum	43.5	30.4	38.5	44.0	39.3	40.5

Wortstrafe und Phrasenstrafe

In Tabelle 5.6 werden die Auswirkungen der Wortstrafe (WP) und der Phrasenstrafe (PP) gezeigt. Bei diesen Experimenten wurde die Maximum Approximation benutzt, da bei dem Summenkriterium die Anzahl der Phrasen für

Abbildung 5.1: Vergleich der mWER für Maximum Approximation und Summenkriterium auf dem VERBMOBIL Developmentkorpus für die Standardrichtung $p(f|e)$.



eine Übersetzung nicht eindeutig ist. Für die Standardrichtung $p(f|e)$ wurde 0.4 als Wortstrafe c_W verwendet, es handelt sich also um eine echte Strafe, und 7.4 als Phrasenstrafe c_P. Hierbei handelt es sich nun eigentlich um einen Bonus pro Phrase. Wie man in der Tabelle sieht, bewirkt die Wortstrafe, dass die Gesamtzahl der Wörter geringer wird. Dadurch nehmen die Einfügungen ab, aber die Löschungen zu. Die Anzahl der Phrasen bleibt hier konstant.

Durch die Phrasenstrafe, beziehungsweise den Phrasenbonus, bleibt die Zahl der Wörter etwa konstant, aber die Zahl der Phrasen nimmt deutlich zu. Daher sinkt die mittlere Phrasenlänge von 2.1 Wörtern pro Phrase für die Baselinevariante auf 1.3 Wörter pro Phrase für die Variante mit Phrasenstrafe. Durch den Einsatz der Phrasenstrafe nehmen die Einfügugen zu und die Löschungen ab.

Für die inverse Richtung $p(e|f)$ wurde 2.0 als Wortstrafe c_W verwendet, hier handelt es sich also im Gegensatz zur Standardrichtung um einen Bonus pro Wort. Die Auswirkungen hiervon sind allerdings nur gering.

Es wurde 6.7 als Phrasenstrafe c_P benutzt. Dies ist wie bei der Standar-

Tabelle 5.6: Auswirkungen von Wortstrafe und Phrasenstrafe auf dem VERBMOBIL Testkorpus. (W=Wörter, P=Phrasen, W/P=Wörter pro Phrase, W/S=Wörter pro Satz, Ins=Einfügungen, Del=Löschungen, Sub=Ersetzungen)

Variante	W	P	W/P	W/S	Ins	Del	Sub	mWER
MonMax	2753	1345	2.1	11.0	2.7	8.4	22.2	42.1
WP	2630	1342	2.0	10.5	1.6	11.4	21.2	41.0
PP	2779	2105	1.3	11.1	3.0	7.6	20.3	39.8
MonMax	2617	1368	1.9	10.4	1.1	11.2	18.2	38.9
WP	2648	1367	1.9	10.6	1.0	10.4	18.4	38.9
PP	2612	1839	1.4	11.4	1.5	11.1	17.4	38.0

drichtung ein Bonus pro Phrase. Die Ausführung zur Phrasenstrafe bei der Standardrichtung gelten daher hier analog.

In Tabelle 5.7 sind die Fehlerraten für diese Varianten angegeben. Für die Standardrichtung $p(f|e)$ liefert die Wortstrafe geringfügig bessere Übersetzungen, während es für die inverse Richtung $p(e|f)$ genau umgekehrt ist. Außerdem wurde die Variante aufgeführt, bei der die Segmentierungswahrscheinlichkeit modelliert wurde (Seg). Mit dieser Methode werden ähnliche Ergebnisse erreicht wie mit der Wort- oder Phrasenstrafe.

Tabelle 5.7: VERBMOBIL: Varianten monotoner phrasenbasierter Übersetzung.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	MonSum	45.9	32.2	40.9	43.4	39.7	40.0
	+Seg	44.9	31.9	40.3	43.1	38.7	39.4
	+WP	44.7	30.8	39.7	44.8	38.3	39.8
	+PP	44.6	31.5	39.5	44.3	38.1	38.7
	+WP+PP	44.0	31.0	38.9	44.2	38.6	40.2
$p(e\|f)$	MonSum	43.5	30.4	38.5	44.0	39.3	40.5
	+Seg	43.2	30.1	38.4	44.3	38.8	40.3
	+WP	43.6	30.4	38.9	44.6	38.8	39.9
	+PP	42.6	29.3	37.8	44.7	38.4	40.0
	+WP+PP	42.9	30.2	38.5	44.3	38.0	40.5

Lexikonvarianten und zweidimensionale n-Gramme

In Tabelle 5.8 sind die Ergebnisse der beiden Lexikonvarianten aus Abschnitt 3.1.4 dargestellt. Wie zu erwarten war, sind die Unterschiede minimal. Daher wird im folgenden immer die geometrische Mittelung verwendet. Die Ergebnisse der Kombination mit den zweidimensionalen n-Grammen sind ebenfalls in dieser Tabelle. Auch hier sind die Unterschiede zu der Glättung mit dem Lexikonmodell gering. Insgesamt ergeben sich mit diesen Modellen allerdings deutlichere Verbesserungen als mit den Heuristiken Wortstrafe und Phrasenstrafe.

Bei dem Vergleich dieser beiden schneidet das zweidimensionale n-Gramm Modell immer etwas besser ab. Auch hier zeigt sich, dass die kontextabhängige Modellierung gewinnbringend ist. Für die Standardrichtung verbessert das zweidimensionale n-Gramm Modell die mWER gegenüber der einzelwortbasierten Lexikon um 1.6%. Insgesamt verbessert das zweidimensionale n-Gramm Modell die mWER um 3.5% für die Standardrichtung und 2.0% für die inverse Richtung.

In Abbildung 5.2 sieht man die Abhängigkeit der mWER von dem Lexikonskalierungsfaktor auf dem VERBMOBIL Developmentkorpus. Ohne Lexikonmodell, also mit einem Skalierungsfaktor von Null, liegt die mWER bei etwa 44.1%. Wird der Skalierungsfaktor erhöht, so fällt die mWER bis auf 42.5% bei einem Skalierungsfaktor von etwa Eins. Danach steigt die mWER wieder an, sogar über den ursprünglichen Wert hinaus. Man kann also mit einem ungünstig skalierten Modell auch schlechtere Übersetzungen erzeugen. Auf dem Developmentkorpus ist eine Verbesserung von 1.6% erreichbar. Wie bereits gesagt, verbessert sich auch auf dem Testkorpus die Übersetzungsqualität. Die mWER sinkt dort sogar um 2.7%.

5.3.3 Nicht-Monotone phrasenbasierte Übersetzung

Umordnungsgraph

Die Größe der Umordnungsgraphen sind in Tabelle 5.9 dargestellt. Die Umordnungsgraphen für das VERBMOBIL Developmentkorpus haben im Mittel etwa 77 Knoten und 147 Kanten pro Satz. Für das Testkorpus haben sie im Mittel ungefähr 64 Knoten und 119 Kanten pro Satz.

Der Vergleich von monotoner Suche (MonSum) und Suche auf dem Umordnungsgraphen (ExtSum) ist in Tabelle 5.10 dargestellt. Man sieht, dass die Nicht-Monotonie deutliche Verbesserungen bewirkt. Die mWER sinkt zum Beispiel um 4.6% für die Standardrichtung $p(f|e)$ und 3.0% für die inverse

Tabelle 5.8: VERBMOBIL: Lexikonvarianten und zweidimensionale n-Gramme in Kombination mit monotoner phrasenbasierter Übersetzung.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	MonSum	45.9	32.2	40.9	43.4	39.7	40.0
	+Lex (arithm.)	42.6	29.6	37.6	46.5	37.5	37.9
	+Lex (geom.)	42.6	29.8	37.5	46.5	37.9	38.7
	+NG	42.8	30.4	37.7	45.3	37.8	39.8
$p(e\|f)$	MonSum	43.5	30.4	38.5	44.0	39.3	40.5
	+Lex (arithm.)	42.2	29.7	37.0	45.7	36.3	38.5
	+Lex (geom.)	42.0	29.3	37.2	45.6	37.9	38.7
	+NG	42.2	29.3	36.9	45.6	36.7	39.5

Abbildung 5.2: Auswirkung des Lexikonskalierungsfaktors auf die mWER für die monotone phrasenbasierte Übersetzung mit Summenkriterium auf dem VERBMOBIL Developmentkorpus für die Standardrichtung $p(f|e)$.

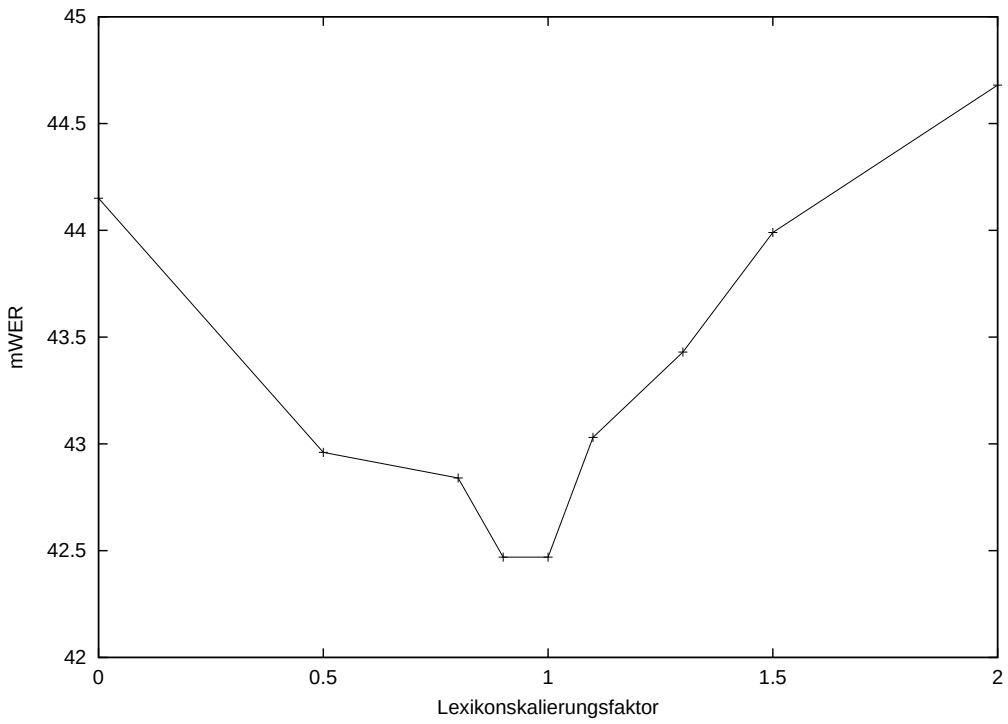


Tabelle 5.9: VERBMOBIL: Größe der Umordnungsgraphen für das Developmentkorpus (Dev) und Testkorpus (Test).

	Knoten		Kanten	
Korpus	insgesamt	pro Satz	insgesamt	pro Satz
Dev	21 359	77.4	40 563	147.0
Test	16 061	64.0	29 797	118.7

Tabelle 5.10: VERBMOBIL: Vergleich monotoner und nicht-monotoner phrasenbasierter Übersetzung.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	MonSum	45.9	32.2	40.9	43.4	39.7	40.0
	ExtSum	42.3	30.1	36.3	50.0	31.9	31.9
$p(e\|f)$	MonSum	43.5	30.4	38.5	44.0	39.3	40.5
	ExtSum	41.1	28.9	35.5	49.3	33.3	34.0

Richtung $p(e|f)$. Die SSER verbessert sich sogar um 7.8% beziehungsweise 6.0%.

Auch für die Suche auf dem Umordnungsgraphen wurden Experimente mit den verschiedenen Varianten durchgeführt. Die Ergebnisse sind in Tabelle 5.11. Unter diesen Fehlerraten sind keine extrapolierten Werte. Für die Standardrichtung stellt man fest, dass Wortstrafe und Phrasenstrafe nur einen geringen oder gar keinen Effekt haben. Die Varianten mit Lexikon oder zweidimensionalen n-Grammen haben in dieser Richtung deutlichere Auswirkungen. Hier ist das zweidimensionale n-Gramm Modell besser als die Lexikonvariante. Es erreicht insgesamt eine Verbesserung der mWER um 3.1%.

Bei der inversen Richtung bringen alle vier Varianten nur leichte Verbesserungen. Die mWER sinkt um $0.4 - 1.1\%$. Hier liefert die Glättung mit dem Lexikon die besten Resultate.

Pattern-Pairs

Tabelle 5.12 zeigt die Auswirkung des maximalen Levels für die Pattern-Pairs. Level 0 bedeutet eine monotone Übersetzung. Wie man sieht verbessert sich die Qualität der Übersetzungen mit steigendem Level.

Tabelle 5.11: VERBMOBIL: Varianten nicht-monotoner phrasenbasierter Übersetzung.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	ExtMax	42.5	30.4	36.7	49.5	32.5	33.0
	ExtSum	42.3	30.1	36.3	50.0	31.9	31.9
	+WP	42.0	30.4	35.8	49.6	32.8	33.8
	+PP	42.3	30.0	36.3	50.0	31.9	32.0
	+Lex	40.4	28.0	34.5	51.4	30.3	32.7
	+NG	39.7	28.0	33.6	52.4	29.0	31.4
$p(e\|f)$	ExtMax	41.6	29.5	35.9	48.4	34.5	35.5
	ExtSum	41.1	28.9	35.5	49.3	33.3	34.0
	+WP	40.6	28.7	34.9	50.5	32.7	33.2
	+PP	40.5	28.1	34.8	49.2	33.2	35.8
	+Lex	40.5	28.4	34.8	49.6	31.8	32.7
	+NG	41.1	28.9	35.5	49.3	33.3	34.0

In Tabelle 5.13 sind die Ergebnisse für die Pattern-Pairs mit Modellskalierung angegeben. Hier wurde allerdings die Suche auf Level 3 eingeschränkt, da für Level 4 die Suche etwa drei Minuten pro Satz dauert (siehe Abschnitt 5.6). Das zusätzliche Lexikon bringt auch hier eine leichte Verbesserung. Für die inverse Richtung zum Beispiel sinkt die mWER von 35.0% um 1.2% auf 33.8%. Bei diesen Verfahren schneidet die Standardrichtung gerinfügig besser ab als die inverse Richtung.

Tabelle 5.12: VERBMOBIL: Pattern-Pair Varianten (keine Modellskalierung).

Kriterientyp		objektiv				subjektiv	
System	Level	WER	PER	mWER	BLEU	SSER	IER
PatSum	0	45.9	32.2	40.9	43.4	39.7	40.0
$p(f\|e)$	1	42.6	30.3	36.8	49.3	32.3	30.1
	2	41.6	29.6	35.6	50.0	32.4	29.4
	3	40.9	29.4	35.0	50.4	32.3	29.3
	4	40.0	29.3	34.2	50.3	31.6	29.1

Tabelle 5.13: VERBMOBIL: Pattern-Pair Varianten (Level 3).

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
$p(f\|e)$	PatSum	39.7	28.7	34.2	50.9	29.9	28.8
	+Lex	39.1	28.6	33.5	51.3	30.9	29.4
$p(e\|f)$	PatSum	40.8	29.5	35.0	48.5	32.1	32.9
	+Lex	39.0	29.6	33.8	50.5	31.7	31.7

5.3.4 Vergleich mit anderen statistischen Ansätzen

Die Ergebnisse der hier vorgestellten Verfahren werden nun mit anderen statistischen Ansätzen verglichen. Dies sind ein einzelwortbasierter Ansatz (SWB, siehe 2.5 und [Tillmann & Ney 00]) und der Alignment Template Ansatz (AT, siehe 2.7 und [Och & Tillmann$^+$ 99]).

Für den einzelwortbasierten Ansatz sind die Ergebnisse für zwei Suchverfahren angegeben. Dies ist ein monotones Verfahren (Mon) und ein quasi-monotones Verfahren (GE). Das quasi-monotone Verfahren erlaubt eine kleine Anzahl von Umordnungen. Diese Umordnungen sind auf die Verbgruppe beschränkt.

Wie man Tabelle 5.14 entnehmen kann liefert die monotone phrasenbasierte Übersetzung deutlich bessere Resultate als die monotone einzelwortbasierte Übersetzung. Die mWER verbessert sich um 5.7% und die SSER sogar um 9.2%. Vergleicht man die nicht-monotone einzelwortbasierte Übersetzung mit den nicht-monotonen phrasenbasierten Varianten, so zeigt sich, dass auch hier die phrasenbasierte Übersetzung deutlich besser abschneidet. Die SSER sinkt von 35.1% auf 29.0% beziehungsweise 30.9%. Die Verwen-

Tabelle 5.14: VERBMOBIL: Vergleich mit anderen Systemen.

Kriterientyp		objektiv				subjektiv	
System	Variante	WER	PER	mWER	BLEU	SSER	IER
SWB	Mon	49.0	35.2	43.4	37.0	47.0	51.7
	GE	41.9	31.4	35.9	47.5	35.1	39.0
PBT	MonSum+NG	42.8	30.4	37.7	45.3	37.8	39.8
$p(f\|e)$	ExtSum+NG	39.7	28.0	33.6	52.4	29.0	31.4
	PatSum+Lex	39.1	28.6	33.5	51.3	30.9	29.4
AT	ME	38.8	28.5	32.6	52.0	30.3	33.5

dung von kontextabhängigen Übersetzungsmodellen führt also zu deutlichen Verbesserungen der Übersetzung.

In Abbildung 5.15 sind die Übersetzungshypothesen der verschiedenen Ansätze für zwei Beispielsätze angegeben.

Tabelle 5.15: VERBMOBIL: Übersetzungsbeispiele.

Quellsatz	ich würde am dreißigsten gern mit dem Zug fahren.
Referenz	I would like to take the train on the thirtieth.
PBT Mon	I would thirtieth like to go by train.
Ext	on the thirtieth I would like to go by train.
Pat	I would like to go by train the thirtieth.
SWB GE	I was on the thirtieth like to go by train.
AT	I was on the thirtieth like to go by train.

Quellsatz	am Mittwoch fahren wir mit dem Zug wieder zurück nach Hamburg.
Referenz	on Wednesday we will go back by train to Hamburg.
PBT Mon	on Wednesday we go by train again back to Hamburg.
Ext	on Wednesday we go by train again back to Hamburg.
Pat	on Wednesday we go back again by train to Hamburg.
SWB GE	on Wednesday we go by train from Hamburg again.
AT	on Wednesday we go by train again back to Hamburg.

5.4 FUB

Bei dem FUB Korpus handelt es sich um ein italienisch-englisches Korpus. Es beinhaltet Anfragen und Beschwerden an eine Hotelrezeption. FUB ist mit 3038 Sätzen ein relativ kleines Korpus. Die Korpusstatistiken sind in Tabelle 5.16 dargestellt. Bei dem FUB Korpus wird keine spezielle Vorverarbeitung durchgeführt.

In Tabelle 5.17 ist die Statistik der bilingualen Phrasen dargestellt. Insgesamt gibt es hier etwa 230 000 bilinguale Phrasen. Davon sind ungefähr 85% Singletons.

Die Ergebnisse auf diesem Korpus sind in Tabelle 5.18 dargestellt. Auch auf diesem Korpus liefert das Summenkriterium bessere Ergebnisse als die Maximum Approximation. Die Glättung mit dem Lexikon ergibt ebenfalls

Tabelle 5.16: FUB Statistik: Trainingskorpus (Train, SZ=Satzzeichen), Testkorpus (Test).

		Italienisch	Englisch
Train	Sätze	3 038	
	Wörter inkl. SZ	55 302	65 446
	Wörter exkl. SZ	47 606	57 588
	Vokabular	2 459	1 701
	Singletons	1 118	662
Test	Sätze	300	
	Wörter	6 121	7 243
	PP (trigr. LM)	19.2	10.1

Tabelle 5.17: FUB: Statistik der bilingualen Phrasen in Abhängigkeit von der Länge der Quellphrase.

		Singletons	
Länge	Anzahl	absolut	relativ (in %)
1	6 943	4 465	64.3
2	19 640	13 374	68.1
3	32 674	25 050	76.7
4	41 337	34 362	83.1
5	45 813	40 333	88.0
6	45 460	41 456	91.2
7	38 060	35 776	94.0
insgesamt	229 927	194 816	84.7

leichte Verbesserungen. Die inverse Übersetzungsrichtung $p(e|f)$ schneidet auf FUB etwas besser ab, als die Standardrichtung $p(f|e)$.

Im Vergleich mit den beiden anderen Übersetzungsverfahren stellt man folgendes fest: die Ergebnisse der phrasenbasierten Übersetzung sind deutlich besser als die der einzelwortbasierten Übersetzung und gleichwertig mit denen der Alignment Templates.

Tabelle 5.18: FUB: Ergebnisse

Kriterientyp		objektiv	
System	Variante	WER	PER
PBT $p(f\|e)$	MonMax	24.5	19.2
	MonSum	24.1	18.7
	MonSum+Lex	23.8	18.3
PBT $p(e\|f)$	MonMax	24.0	19.0
	MonSum	23.5	18.4
	MonSum+Lex	23.0	17.9
SWB		29.3	22.1
AT		23.8	18.0

5.5 El Periodico

Bei dem El Periodico Korpus handelt es sich um Auszüge aus einer elektronisch verfügbaren Zeitung. Diese wird täglich auf Spanisch und Katalanisch veröffentlicht. Das Trainingkorpus besteht aus 10 Monaten der Zeitungsauszüge. Tabelle 5.19 zeigt die Statistik des vorverarbeiteten Korpus. Mit rund 820 000 Sätzen und ungefähr 15 Mio. laufenden Wörtern pro Sprache ist dies ein recht großes Korpus. Detaillierte Ausführungen zur Erzeugung dieses Korpuses findet man in [Tomas & Fabregat$^+$ 01].

Es wurde folgende Art der Vorverarbeitung durchgeführt: Im Originalkorpus sind Namen, Zahlen, Abkürzungen und dergleichen durch ein Label gekennzeichnet. Diese Wörter wurden jeweils durch ihr Label ersetzt. Außerdem wurden aller Wörter die weniger als viermal auftraten durch <unknown> ersetzt.

Die Ergebnisse auf diesem Korpus sind in Tabelle 5.20 dargestellt. In der Standardrichtung bringt hier das Summenkriterium keine Verbesserung. Auch mit einem zusätzlichen Lexikon erreicht man nur minimal bessere Resultate. Für die inverse Richtung ergibt sich mit dem Summenkriterium eine

Tabelle 5.19: El Periodico Statistik: Trainingskorpus (Train, SZ=Satzzeichen), Testkorpus (Test).

		Spanisch	Katalanisch
Train	Sätze	821 516	
	Wörter inkl. SZ	14 415 903	15 075 570
	Wörter exkl. SZ	12 744 329	13 395 288
	Vokabular	60 833	57 900
Test	Sätze	221	
	Wörter	3 382	3 272

leichte Verbesserung. Das zusätzliche Lexikon hat für diese Richtung jedoch keinen Effekt.

Die Vergleichsergebnisse sind aus [Tomas & Casacuberta 01] entnommen. Dort wird ein Übersetzungsverfahren (MonWG) präsentiert, welches der hier vorgestellten phrasenbasierten Übersetzung ähnlich ist. Es basiert ebenfalls auf Wortgruppen. Deren Übersetzungswahrscheinlichkeiten werden mit Hilfe des EM-Algorithmus trainiert. Das Ergebnis mit dem IBM-2 Modell ist ebenfalls aus diesem Artikel entnommen. Wie man sieht, schneiden die beiden phrasenbasierten Methoden (PBT und MonWG) deutlich besser ab als der einzelwortbasierte Ansatz mit IBM-2. Außerdem stellt man fest, dass der hier vorgestellte Ansatz etwa um 1% bessere Resultate liefert das MonWG-System.

Tabelle 5.20: El Periodico Ergebnisse

Kriterientyp		objektiv	
System	Variante	WER	PER
PBT $p(f\|e)$	MonMax	11.4	9.9
	MonSum	11.4	9.9
	MonSum+Lex	11.3	9.7
PBT $p(e\|f)$	MonMax	11.6	9.9
	MonSum	11.2	9.6
	MonSum+Lex	11.2	9.6
MonWG		12.4	
IBM-2		22.3	

Tabelle 5.21: Auswirkungen des Prunings bei monotoner phrasenbasierter Übersetzung auf dem VERBMOBIL Testkorpus (251 Sätze).

	MonMax				MonSum			
maxhyp	SF	SF[%]	mWER	Zeit	SF	SF[%]	mWER	Zeit
1	78	31.1	40.5	0.04	84	33.5	40.4	0.04
5	7	2.8	40.2	0.14	12	4.8	39.4	0.16
10	0	0.0	40.2	0.27	2	0.8	39.6	0.33
30	0	0.0	40.2	0.79	0	0.0	39.6	1.01
60	0	0.0	40.2	1.52	0	0.0	39.6	2.06
100	0	0.0	40.2	2.53	0	0.0	39.6	3.53
300	0	0.0	40.2	6.73	0	0.0	39.6	10.98
1000	0	0.0	40.2	18.88	0	0.0	39.6	40.26
2000	0	0.0	40.2	26.44	0	0.0	39.6	90.30
∞	0	0.0	40.2	26.89	–	–	–	–

5.6 Effizienz

In diesem Abschnitt werden Experimente zum Laufzeitverhalten der verschiedenen Verfahren dargestellt. Zuerst wird für die monotone Suche die Auswirkung des Prunings untersucht, sowie die Maximum Approximation mit dem Summenkriterium verglichen. Anschließend werden die Laufzeiten der verschiedenen Suchverfahren verglichen. Hierbei wird auch der maximale Level der Pattern-Pairs variiert.

In Tabelle 5.21 sind die Auswirkungen des Coverage Histogramm Prunings auf die Laufzeit und die Übersetzungsqualität, gemessen an der mWER, dargestellt. Es wurde kein anderes Pruning angewendet. Die Spalte 'maxhyp' entspricht dem Pruningparameter, also der maximalen Anzahl von Hypothesen pro Position des Quellsatzes. In den Spalten 'SF' ist die Anzahl der Suchfehler absolut und relativ dargestellt.

Erlaubt man nur eine Hypothese pro Position, so entspricht die Suche einem Greedy-Algorithmus. Wie man sieht, werden in diesem Fall bei rund einem Drittel der Sätze Suchfehler gemacht. Wird die Anzahl der Hypothesen pro Position erhöht, fällt die Zahl der Suchfehler sehr schnell auf Null ab. Die Suchverfahren sind also relativ robust gegen diese Art des Prunings. Interessant ist, dass sich die Zahl der Suchfehler nur geringfügig auf die Übersetzungsqualität auswirkt.

Vergleicht man die Übersetzungszeiten der beiden Verfahren miteinander, so sieht man, dass sie sich bei starkem Pruning nur wenig unterscheiden. Bei

moderatem oder schwachem Pruning hingegen gibt es deutliche Unterschiede. Außerdem fällt auf, dass die Laufzeit bei dem Summenkriterium stärker ansteigt als bei der Maximum Approximation. Bei 100 Hypothesen pro Position erfordert das Summenkriterium zum Beispiel einen Mehraufwand von etwa 40%, bei 2000 Hypothesen sind es ungefähr 340%. Diese Differenzen lassen sich mit den unterschiedlichen Komplexitäten (siehe Abschnitt 4.1) begründen.

In Tabelle 5.22 werden die Übersetzungszeiten pro Satz der verschiedenen Suchverfahren auf dem VERBMOBIL Testkorpus verglichen. Die mittlere Satzlänge beträgt bei diesem Korpus etwa 10.5 Wörter. Der Unterschied zwischen der monotonen Suche und der Suche auf dem Umordnungsgraphen liegt etwa bei einem Faktor 10.

Die Pattern-Pair Suche mit Level 0, diese erlaubt nur monotone Übersetzungen, ist genauso schnell wie die monotone Suche. Dies ist interessant, weil bei der Pattern-Pair Suche der Suchgraph explizit aufgebaut wird. Dies ist bei der monotonen Suche nicht der Fall.

Wie man sieht, steigt bei der Pattern-Pair Suche die Übersetzungszeit in Abhängigkeit des Levels stark an. Bei Level 4 dauert die Übersetzung im Mittel drei Minuten pro Satz.

Tabelle 5.22: Die Übersetzungszeiten pro Satz der verschiedenen Suchverfahren bei gleichem Pruning für das VERBMOBIL Testkorpus (mittlere Satzlänge: 10.5 Wörter).

Verfahren		Zeit in s
MonSum		0.13
ExtSum		1.46
PatSum Level	0	0.13
	1	1.04
	2	6.90
	3	39.06
	4	180.22

Kapitel 6

Zusammenfassung und Ausblick

In dieser Diplomarbeit wurden kontextabhängige statistische Übersetzungsmodelle untersucht. Es wurden drei Modelle vorgestellt:

- ein phrasenbasiertes Modell,
- ein Pattern-Pair Modell und
- ein zweidimensionales n-Gramm Modell.

Für das phrasenbasierte Modell wurde eine monotone Suche vorgestellt, sowie zwei nicht-monotone Suchverfahren. Das eine basiert auf einem Umordnungsgraphen, das andere auf Pattern-Pairs. Alle drei Suchverfahren verwenden Dynamische Programmierung.

Es wurden umfangreiche Experimente auf verschiedenen Korpora durchgeführt, insbesondere auf dem VERBMOBIL-Korpus.

Die wesentlichen Resultate lassen sich wie folgt zusammenfassen:

1. Die phrasenbasierte Übersetzung liefert bessere Ergebnisse als die einzelwortbasierte Übersetzung.
2. Nicht-monotone Suche ist wichtig um gute Übersetzungen zu erzeugen.
3. Das Summenkriterium verbessert die Übersetzungen gegenüber der Maximum Approximation.
4. Ein kontextabhängiges Lexikonmodell (zweidimensionale n-Gramme) liefert meist bessere Ergebnisse als das übliche kontextunabhängige Lexikonmodell.

Es gibt sicher noch eine Reihe von Verbesserungs- und Erweiterungsmöglichkeiten. Hierzu sind weitere Untersuchungen notwendig.

- Die Skalierungsfaktoren der einzelnen Modelle könnte man mittels diskriminativem Training bestimmen [Och & Ney 02a].

- Die Phrasenwahrscheinlichkeiten werden zur Zeit durch relative Häufigkeiten geschätzt. Hier könnte man möglicherweise den EM-Algorithmus einsetzen [Tomas & Casacuberta 01, Bisani & Ney 02].

- Das hier beschriebene phrasenbasierte Übersetzungsmodell ist auf Phrasenebene kontextunabhängig. Dies könnte man ändern, indem man zum Beispiel in Gleichung (3.7) eine first-order Abhängigkeit einführt statt der bisherigen zero-order Abhängigkeit. Hierbei würde sich allerdings die Anzahl der zu schätzenden Parameter enorm erhöhen. Dadurch wird das *data sparseness problem* wesentlich deutlicher auftreten. Dies macht eine geeignete Glättung notwendig.

- Es könnten verschiedene Varianten des Pattern-Pair Modells getestet werden. Zur Zeit darf für den Platzhalter V nur eine Phrase eingesetzt werden. Hier könnte man auch Phrasenfolgen erlauben. Dies würde den Suchaufwand allerdings deutlich steigern.

- Die Segmentierungswahrscheinlichkeit $Pr(S|e_1^I)$ wird bei den vorgestellten nicht-monotonen Varianten des phrasenbasierten Übersetzungsmodells nicht berücksichtigt. Eine Modellierung durch ein Phrasenbigramm oder ein Phrasenklassenbigramm könnte sinnvoll sein.

- Anstelle der heuristischen Vorverarbeitung wie sie jetzt durchgeführt wird, könnte eine Vorverarbeitung mit tieferem Verständnis für die jeweilige Sprache treten. Ein Beispiel hierfür ist die morpho-syntaktische Analyse [Nießen & Ney 00].

- Man könnte, wie in [Zubiaga 99], eine zusätzliche Phrasendatenbank hinzunehmen. Ein Problem hierbei besteht allerdings in der Gewichtung dieser Phrasen, also der Zuweisung einer Wahrscheinlichkeit.

- Es könnte eine verfeinerte Methode der Alignment-Symmetrisierung anstelle der Vereinigung verwendet werden. In [Och & Ney 02b] werden verschiedene derartige Methoden vorgestellt.

- Es wird die Heuristik verwendet, unbekannte Wörter durch sich selbst zu übersetzen. Hier könnte man zum Beispiel ein hierarchisches Lexikon, wie in [Nießen & Ney 01] beschrieben, einsetzen.

Anhang A

Modellskalierungsfaktoren

In diesem Anhang sind die Skalierungsfaktoren für die Experimente dargestellt. Die Tabellen zeigen die Faktoren für die unterschiedlichen Verfahren auf den verschiedenen Korpora. Zur Wort- und Phrasenstrafe ist zu sagen: die angegebenen Werte entsprechen den natürlichen Logarithmen von c_W beziehungsweise c_P. Für jedes Experiment gibt es zwei Zeilen. Die erste Zeile enthält die Skalierungsfaktoren für die Standardrichtung $p(f|e)$ und die zweite Zeile diejenige für die inverse Richtung $p(e|f)$. Die Skalierungsfaktoren sind so normiert, dass das Sprachmodell den Faktor 1.0 erhält.

Tabelle A.1: VERBMOBIL: Skalierungsfaktoren der einzelnen Modelle bei nicht-monotoner Übersetzung auf dem Umordnungsgraphen (Sprachmodellskalierungsfaktor=1.0).

Variante	$p(f\|e)$	$p(e\|f)$	WP	PP	Lex	NG
Max	1.10					
		4.22				
Sum	0.98					
		3.46				
Sum+WP	1.00		-0.39			
		3.03	0.70			
Sum+PP	1.00			0.00		
		2.83		2.00		
Sum+Lex	0.94				0.49	
		3.21			0.40	
Sum+NG	1.00					0.60
		2.77				0.23

Tabelle A.2: VERBMOBIL: Skalierungsfaktoren der einzelnen Modelle bei monotoner Übersetzung (Sprachmodellskalierungsfaktor=1.0).

Variante	$p(f\|e)$	$p(e\|f)$	Seg	WP	PP	Lex	NG
Max	0.98						
		3.38					
Sum	0.98						
		2.42					
Sum+Seg	0.88		0.30				
		3.28	0.69				
Sum+WP	1.39			-0.85			
		3.16		0.65			
Sum+PP	0.82				1.95		
		3.85			1.94		
Sum+Lex (geom.)	0.98					0.86	
		2.70				0.50	
Sum+Lex (arithm.)	1.10					0.83	
		2.55				0.54	
Sum+NG	0.60						0.93
		3.92					1.50
Sum+WP+PP	1.05			-0.79	1.38		
		1.84		0.57	2.22		
Sum+NG+WP	0.72			-0.70			0.80
		3.99		0.99			1.48
Sum+NG+WP+PP	0.54			-1.00	1.40		0.60
		2.62		0.70	2.33		0.50

Tabelle A.3: VERBMOBIL: Skalierungsfaktoren der einzelnen Modelle bei phrasenbasierter Übersetzung mit Pattern-Pairs (Level 3) (Sprachmodellskalierungsfaktor=1.0).

Variante	$p(f\|e)$	$p(e\|f)$	Lex	Pat
Sum	1.26			1.06
		2.82		3.90
Sum+Lex	0.82		0.25	1.08
		2.74	0.31	3.68

Tabelle A.4: FUB: Skalierungsfaktoren der einzelnen Modelle bei monotoner phrasenbasierter Übersetzung (Sprachmodellskalierungsfaktor=1.0).

Variante	$p(f\|e)$	$p(e\|f)$	Lex
Max	0.63		
		1.70	
Sum	0.77		
		1.95	
Sum+Lex	0.96		0.84
		1.42	0.55

Tabelle A.5: El Periodico: Skalierungsfaktoren der einzelnen Modelle bei monotoner phrasenbasierter Übersetzung (Sprachmodellskalierungsfaktor=1.0).

Variante	$p(f\|e)$	$p(e\|f)$	Lex
Max	1.27		
		4.42	
Sum	1.38		
		4.98	
Sum+Lex	1.21		0.30
		4.98	0.00

Literaturverzeichnis

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

[Bisani & Ney 02] M. Bisani, H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. 4 pages, 2002. To appear.

[Block 00] H. U. Block. Example-based incremental synchronous interpretation. In W. Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 411–417. Springer Verlag, Berlin, Germany, July 2000.

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

[Ney 00] H. Ney. Machine translation. Course notes, summer term 2000, RWTH Aachen, Germany, September 2000.

[Ney 01] H. Ney. Language modelling. Course notes, winter term 2000/2001, RWTH Aachen, Germany, March 2001.

[Nießen & Ney 00] S. Nießen, H. Ney. Improving SMT quality with morpho-syntactic analysis. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 1081–1085, Saarbrücken, Germany, July 2000.

[Nießen & Ney 01] S. Nießen, H. Ney. Toward hierarchical models for statistical machine translation of inflected languages. In *Data-Driven Machine Translation Workshop*, pp. 47–54, Toulouse, France, July 2001.

[Nießen & Och+ 00] S. Nießen, F. J. Och, G. Leusch, H. Ney. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 39–45, Athens, Greece, May 2000.

[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 02a] F. J. Och, H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 8 pages, July 2002. To appear.

[Och & Ney 02b] F. J. Och, H. Ney. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, 2002. To appear.

[Och & Tillmann$^+$ 99] F. J. Och, C. Tillmann, H. Ney. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, University of Maryland, College Park, MD, June 1999.

[Papineni & Roukos$^+$ 01] K. A. Papineni, S. Roukos, T. Ward, W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, 10 pages, September 2001.

[Tessiore & v. Hahn 00] L. Tessiore, W. v. Hahn. Functional validation of a machine interpretation system: Verbmobil. In [Wahlster 00], pp. 611–631.

[Tillmann & Ney 00] C. Tillmann, H. Ney. Word re-ordering and DP-based search in statistical machine translation. In *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Tillmann 01] C. Tillmann. *Word Re-Ordering and Dynamic Programming based Search Algorithms for Statistical Machine Translation.* Ph.D. thesis, Computer Science Department, RWTH Aachen, Germany, May 2001.

[Tomas & Casacuberta 01] J. Tomas, F. Casacuberta. Monotone statistical translation using word groups. In *Proc. of Machine Translation Summit VIII*, 5 pages, September 2001.

[Tomas & Fabregat$^+$ 01] J. Tomas, F. Fabregat, J. M. del Val, F. Casacuberta, D. Pico, A. Sanchis, E. Vidal. Automatic development of spanish-catalan corpora for machine translation. In *Proc. of The Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*, 5 pages, Jain, Spain, September 2001.

[Vogel & Ney$^+$ 96] S. Vogel, H. Ney, C. Tillmann. HMM-based word alignment in statistical translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pp. 836–841, Copenhagen, Denmark, August 1996.

[Wahlster 00] W. Wahlster, editor. *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, July 2000.

[Zubiaga 99] A. Zubiaga. Einsatz von bilingualen Phrasen für die statistische Übersetzung. Diploma thesis, RWTH Aachen, Germany, January 1999.