Minimum Bayes-Risk Techniques in Automatic Speech Recognition and Statistical Machine Translation

Shankar Kumar

A dissertation submitted to the Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland
October 2004

Abstract

Automatic Speech Recognition (ASR) and Machine Translation (MT) are fundamental language technologies that are emerging as core components of information processing systems. Each of these problems can be evaluated using a variety of metrics that measure different aspects of recognition or translation performance. In contrast, the training and decoding architectures of most of the current ASR and statistical MT systems are optimized with respect to Sentence Error Rate that is rarely used in evaluating these systems. The goal of this thesis is to overcome this mismatch by building automatic systems specialized for each individual evaluation metric. We employ the Minimum Bayes-Risk (MBR) classification framework to construct systems sensitive to specific error criteria. We present the formulation of MBR decoders in speech recognition and in two sub-problems in machine translation: bitext word alignment and translation.

MBR decoding is performed by rescoring a set of likely hypotheses represented as lattices or N-best lists. Statistical ASR systems for generating word lattices have become widely available in the recent years. In contrast, Statistical MT (SMT) has become popular only within the last decade and we did not have access to SMT systems for generating word alignment and translation hypotheses. We therefore formulate and implement a generative, source channel Translation Template Model (TTM) for SMT. The approach we describe allows us to implement each stochastic transformation in this model using a weighted finite state transducer (WFST). This allows translation and bitext word alignment to be realized immediately by standard WFST operations on the component transducers. The TTM is the first phrase-based translation model to be used for bitext word alignment. We describe the construction of a TTM Chinese-to-English translation system that ranked among the top performing systems in the NIST 2004 international MT evaluation.

MBR decoders face computational challenges when applied to large vocabulary speech recognition tasks. We introduce the segmental MBR recognition framework that decomposes a large MBR search problem into a sequence of smaller MBR problems. To achieve this, we develop a risk-driven lattice segmentation procedure to segment large recognition word lattices into smaller sub-lattices over which MBR decoding is performed. Lattice segmentation, in conjunction with MBR decoding, gives consistent gains on a large vocabulary ASR task.

We finally present MBR decoders for bitext word alignment and translation. We show the construction of alignment and translation loss functions from standard evaluation metrics as well as from linguistic analyses such as parse-trees and part-of-speech tags. In both cases we show that MBR decoding can be used to tune statistical MT performance under specific loss functions.

Advisor: Prof. William Byrne
Readers: Prof. William Byrne and Prof. Trac Tran
Thesis Committee: Prof. William Byrne, Prof. Trac Tran,
Prof. Frederick Jelinek, Prof. Gert Cauwenberghs, and Prof. David Yarowsky

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Bill Byrne for his wonderful guidance and strong support. He allowed me an unparalleled level of freedom in pursuing ideas while closely supervising my work. His insistence on mathematical rigor and on clarity in writing went a long way in improving this work. I truly value his outstanding mentorship during these years.

I am expressly grateful to Prof. Frederick Jelinek for giving me the opportunity to work at the Center for Language and Speech Processing (CLSP), encouraging my research, and fostering my professional growth.

My sincere gratitude to the members of my thesis committee, Professors Frederick Jelinek, Trac Tran, Gert Cauwenberghs, and David Yarowsky, for their useful feedback which has undoubtedly improved this work, and for their timely flexibility.

I specially thank Prof. Sanjeev Khudanpur for his valuable advice and assistance with myriad aspects of this work and otherwise. I am also grateful to Prof. David Yarowsky and Prof. Jason Eisner for their insightful discussions and inputs. Thanks are due to Dr. Zak Shafran for many interesting discussions and useful comments on my thesis proposal.

I would like to thank all my senior colleagues at CLSP - Vaibhava, Asela, Ciprian, Murat, Dimitra, Lidia, Hans, Silviu, Jun - for giving me good advice and helping me in various ways during my initial years at Hopkins. I am grateful to Vaibhava, Asela, and Murat for teaching me many practical aspects in building speech recognition systems that I could not have learnt otherwise.

My work was made really enjoyable by my colleagues and friends at CLSP, notably Peng, Veera, Paola, Yonggang, Vlasios, Shantanu, Ahmad, Woosung, Stavros, Jia, Yan, Sourin, Arnab, Ali, Erin, Lambert, Srihari, Yi, Vidya, Charles, Gideon, Noah, David, Roy, Elliott, Markus, Brock, Filip, Pavel, and Petr. Their friendship and enthusiastic support made the center a lively and exciting place to work. In particular, I am grateful to Peng, Veera, Petr, and Yonggang for resourceful discussions that contributed to my work and pointed me in interesting directions.

I thank Paola for inviting me to her dinners and shopping expeditions that pro-

vided me with many a welcome break from monotony. Thanks to Nicole and Roy for entertaining me with many delightful get-togethers.

My thanks to everyone for giving me their constructive feedback on the many drafts of papers and sitting through the dry-runs of talks.

I especially thank Franz Och for influential discussions and for providing me various resources and tools that were helpful in this work. I am grateful to the members of the Syntax for Statistical MT group in the 2003 JHU Summer Workshop - Franz, Dan, Sanjeev, Anoop, Kenji, Alex, Libin, David, Viren, Zhen, Katherine, and Drago - for the interactive sessions, tools, and ideas that contributed to this research.

I thank AT&T Labs - Research for use of the FSM toolkit and Andreas Stolcke for use of the SRILM toolkit.

I am grateful to Laura Graham, Sue Porterfield, Conway Benishek, Wilhelmena Braswell, Amy Berdann, Kimberly Petropoulos, and Janet Lamberti of the CLSP administrative staff, and Gail O'Connor of the ECE department for their professionalism, and for making all administrative matters run smoothly. I am indebted to Jacob Laderman, Dave Smith, Daniel Bell, and Eiwe Lingefors whose hard work kept the CLSP machines up and running.

Finally, and most importantly, I thank my parents and my sister, Jyoti for their constant support throughout these years. I deeply value their endless love, encouragement, and advice. This thesis is dedicated to them.

To my parents and my sister, Jyoti.

Contents

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

Automatic Speech Recognition (ASR) and Machine Translation (MT) are two key language technologies that are emerging as the core components of various information processing systems [85, 26, 19]. The two technologies share the ultimate goal of transforming information from an unfamiliar representation (speech or foreign text) into text in a readable form. They are also similar in that each can be formulated in a generative source-channel modeling framework [40, 9]. This allows the statistical models for ASR (or MT) to be decomposed into two components: a language model that governs the generation of word strings from the source and an acoustic (or translation) model that describes the transformation of word strings into acoustic observation sequences (or foreign language word strings) by the transfer channel. Finally, there is a third aspect in which ASR and MT resemble each other; outputs of both systems are deployed for similar downstream applications such as Information Extraction, Information Retrieval, and Summarization. The objective of this thesis is to develop statistical classification techniques in these two problems.

In Automatic Speech Recognition we focus on Speech Transcription which involves producing a word level transcription of the acoustic signal. In Statistical Machine Translation (SMT) our focus is on two problems. The first problem is the translation of texts from one natural language (such as French) to another (e.g. English). The second problem is the word alignment of bilingual texts (bitexts). Bitext word alignment involves identification of word and phrase correspondences between pairs of

translated sentences. Bitext word alignments are crucial for building SMT systems. In addition, word alignments are valuable for other natural language processing applications, such as automatic dictionary construction [66] and projection of linguistic annotation (e.g. part-of-speech tags) across languages [106].

1.1 Error Criteria and Classification Techniques

We start with an introduction of criteria used in evaluating speech recognition, bitext word alignment, and machine translation systems. The criteria can be broadly divided into task-specific and task-independent metrics.

Task-specific metrics evaluate the ASR or MT system via performance on specific applications where the transcription or translation is utilized. Examples include Keyword error rate in Word Spotting, Precision and Recall in Information Retrieval, and Slot error rate in Named Entity Detection.

Task-independent metrics, on the other hand, measure different aspects of transcription or translation quality independent of any particular application of the automatic system.

An ASR transcription is usually evaluated under Word Error Rate (WER) and Sentence Error Rate (SER) metrics. Word Error Rate measures the number of incorrectly hypothesized words in the transcription relative to a manually generated transcription. Sentence Error Rate measures the number of incorrectly hypothesized sentences in the ASR transcription.

In translation, a bitext word alignment is evaluated with respect to a reference word alignment created by a competent human translator. The alignment performance against the reference alignment is measured using Alignment Precision, Alignment Recall, and Alignment Error Rate (AER) metrics [82].

Evaluation of MT systems is a harder problem than the evaluation of speech recognition or word alignment systems. Conventionally, translations have been evaluated by human translators who weigh many aspects of translation, including adequacy, fidelity, and fluency [39, 105]. However, such evaluations are extremely expensive and time consuming thus creating a bottleneck for MT system development. In re-

System	BLEU (%)	NIST	F-measure	mPER(%)	mWER(%)
1	30.6	9.02	0.77	41.75	65.16
2	27.7	8.58	0.74	43.86	66.83
3	21.9	7.91	0.70	46.46	68.49
4	24.2	7.56	0.75	47.84	70.26
5	19.2	7.25	0.72	49.56	71.18
6	16.8	7.05	0.64	51.03	72.32
7	10.2	4.32	0.54	59.04	75.98

Table 1.1: Translation Performance of seven MT systems as measured under various evaluation criteria. Results are reported on the NIST 2003 Chinese-English evaluation set and the measurements are case-insensitive. Note that better systems correspond to higher BLEU, NIST and F-measure scores, and to lower error rates (mPER and mWER).

cent years there has been active research in developing automatic translation metrics that correlate well with human judgements of translation quality [86, 21, 67]. This has resulted in the development of various translation metrics such as the BLEU score [86], NIST score [21], F-score [67], multi-reference Position-independent Word Error Rate (mPER) [79], multi-reference Word Error Rate (mWER) [79], and many other measures (See [5]). Unlike ASR metrics, these criteria measure performance of the automatic translation against multiple reference translations produced by different translation agencies.

The above criteria illustrate the variety of metrics by which MT systems can be evaluated. Given the difficulty of judging translation quality, it is unlikely that a single global metric will perform better than all other metrics. It is more likely that specialized metrics will be used to measure specific aspects of MT system performance. Our interest in this thesis is not in the development of better evaluation metrics. Instead, we would like to investigate how automatic systems can be tuned for each individual criterion.

Towards this goal we now study how different evaluation metrics can influence the relative rankings of automatic systems. We obtain translations from seven MT systems on the NIST 2003 Chinese-English MT evaluation set [78], and then evaluate each translation under the five translation criteria mentioned above (Table 1.1). We

Hypothesis	BLEU (%)	mPER (%)	mWER (%)
MAP	27.1	44.2	67.3
Oracle-best translation			
Metric			
BLEU	38.4	39.0	62.1
mPER	31.5	34.9	62.0
mWER	32.2	38.8	56.8

Table 1.2: Performance of Oracle-best translations under different evaluation metrics. The oracle-best translation is computer over an N-best list generated by an MT system. We also show the performance of the Maximum A Posteriori (MAP) translation produced by the MT system.

observe that ranks of systems 2, 3, 4, and 5 are sensitive to the evaluation criterion while the ranks of systems 1, 6, and 7 are invariant to the criterion. This study shows that different translation metrics can often rank automatic systems differently. This is of importance in system development because an MT system that is optimized with respect to a given metric might perform poorly when evaluated against a different metric. We therefore see a need to build specialized MT systems optimized for each individual translation criterion.

To further see the value in building specialized decoders, we perform a second experiment. We generate a 1000-best list of translations under a baseline MT system (see Chapter 3). From this list we obtain the translation that is closest to the reference translation(s) under each translation metric; we will refer to this translation as the *oracle-best translation* under the metric. In Table 1.2 we show the performance of the oracle-best translations under the BLEU, mPER, and mWER metrics. We also report the performance of the Maximum A Posteriori (MAP) translation produced by the MT system.

We observe that under each of the three translation metrics, the oracle-best translation can yield substantial gains over the MAP hypothesis. In all instances, the oracle-best hypothesis for a given metric gives the best translation performance under the corresponding criterion. This suggests that it might be useful to develop decoding procedures that are tuned for each evaluation criterion.

In this thesis we will employ a statistical classification framework that enables

us to construct speech recognition, bitext word alignment and translation systems sensitive to specific error criteria. Depending on the problem, the framework attempts to minimize the cost of recognition, alignment, or translation errors. We will show how this framework can compensate for the mismatch between decoding criteria of systems and their evaluation criteria. This is of value in particular because most of the current ASR and MT systems use Maximum A Posteriori (MAP) techniques for decoding [40]. The MAP decoding criterion is optimal under the Sentence Error Rate which is rarely the metric of interest in evaluating ASR or MT systems.

The framework that we propose requires the design of *Loss Functions* to measure the quality of automatically produced hypotheses (e.g. transcriptions) relative to manually produced hypotheses. We will show that loss functions can be obtained in two ways. They can be derived from standard evaluation metrics; alternatively, they can be constructed so as to measure characteristics such as syntactic well-formedness in the hypothesis.

Given a desired loss function, we now introduce the classification framework that will allow us to build a decoder optimized under this loss function. We will introduce this framework in the context of speech recognition.

1.2 Minimum Bayes-Risk Decoding Framework

Automatic Speech Recognition can be viewed as a classification problem in which an acoustic observation sequence $A = a_1, a_2, ..., a_T$ is mapped to a word string $W = w_1, w_2, ..., w_N$, where w_i are words belonging to a vocabulary $\mathcal{V}$.

We assume that a language $\mathcal{W}$ is known; it is a subset of the set of all word strings over $\mathcal{V}$. This language specifies the word strings that could have produced any acoustic data seen by the ASR system. We further assume that the ASR classifier selects its hypothesis from a set $\mathcal{W}_h$ of word strings. This set, called the *hypothesis space* of the classifier, would usually be a subset of the language. The ASR classifier can then be described as the functional mapping $\delta(A) : \mathcal{A} \rightarrow \mathcal{W}_h$.

Let $l(W, W')$ be a real valued loss function that describes the cost incurred when an utterance W belonging to language $\mathcal{W}$ is mistranscribed as $W' \in \mathcal{W}_h$. An example

loss function is Levenshtein distance [62] that measures the minimum string edit distance (word error rate) between W and W'. This loss function is defined as the minimum number of substitutions, insertions and deletions needed to transform one word string into another.

Suppose the true distribution $P(W, A)$ of speech and language is known. It would then be possible to measure the performance of a classifier δ as

$$R(\delta(A)) = E_{P(W,A)}[l(W, \delta(A))]. \tag{1.1}$$

This is the expected loss when $\delta(A)$ is used as the classification rule for data generated under $P(W, A)$. The classification rule that minimizes $R(\delta(A))$ is given by [7]

$$\delta_R(A) = \underset{W' \in \mathcal{W}_h}{\operatorname{argmin}} \sum_{W \in \mathcal{W}} l(W, W')P(W|A). \tag{1.2}$$

We note that while the sum in Equation 1.2 is carried out over the entire language of the recognizer, only those word strings with non zero conditional probability $P(W|A)$ contribute to the sum. Let $\mathcal{W}_e$ denote the subset of $\mathcal{W}$ such that

$$\mathcal{W}_e = \{W \in \mathcal{W} | P(W|A) > 0\}. \tag{1.3}$$

Equation 1.2 can now be re-written as

$$\delta_R(A) = \underset{W' \in \mathcal{W}_h}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_e} l(W, W')P(W|A). \tag{1.4}$$

The sum $\sum_{W \in \mathcal{W}_e} l(W, W')P(W|A)$ in Equation 1.4 is called the *posterior risk* [7]; for convenience, we will drop 'posterior' and refer to it as the risk. We refer to the classifier given by Equation 1.4 as the *Minimum Bayes-Risk* (MBR) classifier.

The set $\mathcal{W}_e$ serves as the evidence for the MBR classifier using which it selects the hypothesis. Therefore, we shall refer to $\mathcal{W}_e$ as the *evidence space* for the acoustic observations A. The distribution $P(W|A)$ describes the evidence space and shall be referred to as the *evidence distribution*.

Our treatment so far assumes that the true distribution over the evidence is available. This distribution is obtained by applying Bayes rule

$$P(W|A) = P(W)P(A|W)/P(A), \tag{1.5}$$

where the component distributions are approximated by models. Of course, $P(W)$ and $P(A|W)$ are not available and must be approximated by models. $P(W)$ is approximated as the language model and $P(A|W)$ is obtained from hidden Markov model acoustic likelihoods.

1.2.1 Lattices and N-best Lists

We approximate the hypothesis and the evidence spaces ($\mathcal{W}_e$ and $\mathcal{W}_h$) of the MBR decoder (Equation 1.4) by *Lattices* or *N-best Lists* generated under a statistical model. We now briefly introduce these two terms; more details on the structure and representation of lattices and N-best lists will be provided within specific contexts in this thesis.

A *Lattice* refers to a large collection of likely hypotheses that can be compactly represented as a directed acyclic graph or an acyclic Weighted Finite State Acceptor (WFSA) [74, 73]. Formally, a WFSA $\mathcal{A} = (Q, \Lambda, q_s, F, \mathcal{T})$ is given by a finite set of states Q, a set of transition labels Λ, an initial state $q_s \in Q$, a set of final states $F \subseteq Q$, and a finite set of transitions $\mathcal{T}$. A transition $t = (p, q, l, s)$ can be represented by an arc from the source state p to the destination state q, with the label l and the weight s. A path in $\mathcal{A}$ is a sequence of consecutive transitions $T = t_1...t_n$ such that $p_1 = q_s$, $p_{i+1} = q_i$, and $q_n \in F$. Figure 1.1 gives an example of a recognition word lattice represented as a WFSA; each path in this lattice is a sentence hypothesis generated by a speech recognizer.

In our applications, the transition weight will represent a negative log probability so that the weight associated with the path T is computed as $\sum_{i=1}^{n} s_i$. Each path in the lattice represents a hypothesis l_1^n along with its log likelihood $-\sum_{i=1}^{n} s_i$ under the statistical model. In this dissertation we will introduce lattices for speech recognition, bitext word alignment, and translation. The lattices in each of these problems share the same underlying WFSA structure but the labels $l \in \Lambda$ on the transitions of the WFSA will be different for each problem. In recognition and translation lattices, labels specify words (as in Figure 1.1) while in bitext word alignment, labels specify links between words in the source language (e.g. English) sentence and the target

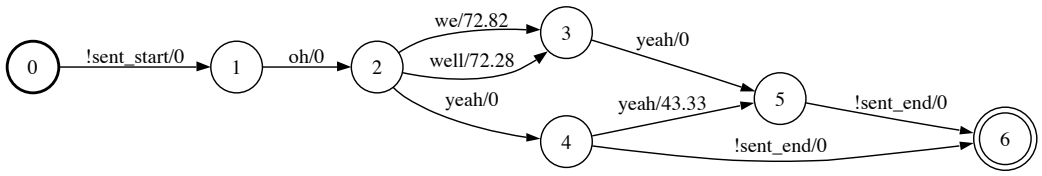

Figure 1.1: An example of a recognition word lattice represented as a Weighted Finite State Acceptor. States are represented by circles. The initial state is represented by a bold circle and the final state by double circles. The label and weight of a transition are marked on the corresponding arc by l/w.

language (e.g. French) sentence.

An *N-best List* is an enumeration of the N most likely hypotheses (transcriptions, translations, word alignments) given a observation. An N-best list can be generated from a lattice as strings with N highest likelihood values [71].

1.2.2 Relationship with MAP decoding

We now describe how the MBR decoder is related to the conventional MAP decoder

$$\delta_{\text{MAP}}(A) = \underset{W \in \mathcal{W}_e}{\operatorname{argmax}} P(W|A). \tag{1.6}$$

The MAP decoder can be seen as a special case of the MBR decoder by considering a hypothesis space that is identical to the evidence space and a 0/1 loss function that assigns equal cost of one to all misclassifications.

$$l_{0/1}(W, W') = \begin{cases} 1 & W \neq W' \\ 0 & \text{otherwise.} \end{cases} \tag{1.7}$$

Under these conditions, the MBR decoder of Equation 1.4 reduces to

$$\delta(A) = \underset{W' \in \mathcal{W}_h}{\operatorname{argmax}} P(W'|A), \tag{1.8}$$

where

$$P(W'|A) = \sum_{W \in \mathcal{W}_e} \delta_W(W') P(W|A). \tag{1.9}$$

This is the MAP decoder of Equation 1.6.

The above derivation illustrates why we are interested in MBR decoder for general loss functions. The MAP decoder is the optimal decoder with respect to a loss function that is overly harsh. This loss function does not differentiate between different types of recognition errors, and all wrong hypotheses are penalized equally. In contrast, we note that the MBR decoder can be thought of as selecting the *Consensus Hypothesis* under a particular loss function [63, 34]. Equation 1.4 selects the hypothesis that is, in an average sense, closest to all other likely hypotheses.

1.3 Related Work

We here review related work in Automatic Speech Recognition, Statistical Machine Translation and other areas of Natural Language Processing that incorporate application-specific or evaluation-specific criteria into estimation and decoding procedures of automatic systems.

In Automatic Speech Recognition, early investigations into the use of minimum Bayes-risk criteria for training speech recognizers were performed by Nadas [76, 77]. Since then other researchers [75, 41, 25, 23, 22] have investigated Bayes-risk criteria in training acoustic models for ASR. However, our primary interest is in minimum risk classification rather than model estimation.

Stolcke et.al. [92] proposed an approximation of the minimum risk classifier to generate the minimum expected word error rate hypothesis from N-best lists of transcriptions. Other researchers [28, 27, 63] have proposed posterior probability and confidence based hypothesis selection strategies for word-error rate reduction that can be shown as approximations of MBR classifiers [34]. Goel and Byrne [33] presented the formulation of MBR classifiers for ASR and developed efficient implementations of MBR decoders via rescoring of N-best lists and word lattices. Our work will closely follow their formulation.

In Statistical Machine Translation, Och [80] developed a training procedure that incorporates various MT evaluation criteria in the training procedure of log-linear MT models. Foster et.al. [29] developed a text-prediction system for translators that maximizes expected benefit to the translator under a statistical user model. In parsing, Goodman [38] developed parsing algorithms that are appropriate for specific parsing metrics.

There has also been recent work that combines 1-best hypotheses from multiple translation systems [3]; this approach uses string-edit distance to align the hypotheses and rescores the resulting lattice with a language model. This approach differs from MBR decoding in that it combines hypotheses from multiple-systems rather than from an N-best list generated from a single system.

1.4 This Thesis

This thesis explores Minimum Bayes-Risk decoding approaches in Automatic Speech Recognition and Statistical Machine Translation.

In Automatic Speech Recognition, we will build on previous work in MBR decoding [33]. MBR decoders have been implemented via A^* search over recognizer word lattices, and rescoring of N-best lists. For most practical ASR tasks, word lattices are very large and the MBR recognizer faces computational problems. We therefore explore lattice segmentation strategies (Chapter 2) that decompose a single large MBR search problem over a lattice into a sequence of smaller MBR search problems on lattice segments. These strategies are termed *segmental MBR recognition*. A previously proposed solution to lattice segmentation [36, 32] relied on word confidence scores and time boundary marks inside a recognition lattice. However this approach lacked a rigorous formulation, and was not adequate in practice because word time boundaries are sometimes missing or inaccurate in recognition lattices. We therefore propose a novel lattice segmentation procedure that is motivated by a risk criterion. This approach does not rely on the time marks or the likelihood values produced by the baseline recognizer. We show that lattice segmentation, in conjunction with MBR decoding, gives consistent improvements as a final stage of a large vocabulary speech

recognition system. We also present an application of this segmentation strategy to combine lattices from multiple ASR systems and perform segmental MBR decoding.

We next address the problem of Statistical Machine Translation (Chapters 3-7). Our goal here is to investigate MBR decoders for two problems: bitext word alignment and translation. However, the requirement for building these MBR decoders is a statistical model for generating word alignment and translation hypotheses. We therefore take a detour to discuss the development of a statistical translation model. In Chapter 3 we formulate the *Translation Template Model* (TTM) and describe how each component of this model can be implemented using a Weighted Finite State Transducer (WFST). We then present word alignment and translation under the TTM (Chapters 4 and 5). We show how bitext word alignment and translation under the TTM can be performed entirely using standard WFST operations on the TTM component transducers. In addition, we describe the use of the WFST framework to generate lattices and N-best lists of word alignment and translation hypotheses. We report and analyze the word alignment and translation performance of the TTM on French-English and Chinese-English tasks. We finally discuss the development of a Chinese-to-English TTM system from large training bilingual texts for the NIST 2004 MT evaluation.

Following the presentation of the translation model, we return to the discussion of MBR decoders for word alignment (Chapter 6) and translation (Chapter 7). In both cases we introduce loss functions and discuss their construction. We then present the formulation of MBR decoders for word alignment and translation. In bitext word alignment, we discuss the implementation of MBR decoders on alignment lattices while in translation, we show their implementation on N-best lists. We finally report the performance of MBR decoders under the various loss functions.

We discuss future research directions in Chapter 8 and conclude in Chapter 9.

1.5 Novel Contributions

We here outline the three main novel contributions of this dissertation.

Our first contribution is the development of a risk-based lattice segmentation

procedure for Segmental Minimum Bayes-Risk speech recognition. The procedure allows us to segment a large recognition word lattice into a sequence of smaller sub-lattices. The core of this procedure is a *lattice-to-string alignment* technique that produces a simultaneous Levenshtein alignment of all word strings in a lattice against any given word string. We show that segmental MBR decoding over lattice segments yields performance improvements over MBR decoding on unsegmented lattices.

Our second contribution is the formulation and implementation of a Weighted Finite State Transducer (WFST) *Translation Template Model* (TTM) for statistical machine translation. The TTM is a generative source channel model of the translation process. We provide a derivation of the TTM that allows us to identify the conditional independence assumptions that underly the WFST implementation. We show that bitext word alignment and translation under the TTM can be performed entirely using standard finite state operations. This avoids the necessity to develop specialized search procedures (such as A^* decoding or beam search strategies) for performing word alignment and translation under the model. We also show that this framework also facilitates generation of alignment and translation lattices without any extra effort in implementation.

Our third contribution is the development of Minimum Bayes-Risk decoding procedures for bitext word alignment and translation. In both cases we show the construction of loss functions from standard evaluation criteria or from linguistic analyses of sentences via parse-trees and part-of-speech tags. For bitext word alignment we derive closed form solutions to MBR decoders; this allows us to perform an exact and efficient implementation over alignment lattices. For translation we implement MBR decoders via rescoring of N-best lists. We demonstrate that the MBR framework can be used to build specialized MT decoders under each individual alignment and translation loss function.

1.6 Reader's Guide

Chapter 2 presents the segmental Minimum Bayes-Risk framework for speech recognition and word lattice segmentation procedures; readers with a speech recogni-

tion background may read this chapter by itself.

Readers interested in machine translation may skip straight to Chapters 3-7. Specifically,

- Chapter 3 provides a detailed introduction to the Translation Template Model, its formulation and implementation using weighted finite state transducers.
- Chapters 4 and 5 discuss bitext word alignment and translation under the Translation Template Model.
- Chapters 6 and 7 present Minimum Bayes-Risk decoders for bitext word alignment and translation respectively.

Finally, readers interested in the application of Minimum Bayes-Risk techniques may skip straight to Chapters 2, 6, and 7. These three chapters illustrate the formulation of min-risk techniques in three different problems within speech and language engineering.

Part II

Automatic Speech Recognition

Chapter 2

Segmental Minimum Bayes-Risk Speech Recognition

We have presented the formulation of Minimum Bayes-Risk (MBR) speech recognizers in Section 1.2. For most large vocabulary speech recognition tasks such as SWITCHBOARD [31], the hypothesis and evidence spaces of the MBR decoder are very large, and the MBR recognizer faces computational problems. Previous research in MBR decoding [33] focussed on efficient search procedures to implement the MBR recognizer (Equation 1.4) via rescoring of N-best lists and word lattices. However, when the recognizer search spaces (e.g. word lattices) are large, we still need to prune the spaces when performing MBR decoding using the above implementations. This pruning of word lattices can lead to search errors in MBR decoding. In this chapter we discuss a set of strategies that segment the hypothesis and the evidence spaces of the MBR recognizer in an attempt to avoid search errors. The segmentation transforms the original MBR search problem into a sequence of smaller MBR search problems that can be solved easily. We will refer to this framework as *Segmental MBR decoding* [37, 35] (Section 2.1).

Our focus is on word lattice segmentation procedures that underly segmental MBR decoding. In Section 2.2, we propose a procedure that segments recognition word lattices into smaller sub-lattices. This procedure is motivated by the observation that under an ideal lattice segmentation the risk (Equation 1.1) of each word

string in the lattice is unchanged after segmentation. We next discuss two segmental MBR decoding procedures over N-best lists (Section 2.3); each of these procedures involves segmentation of N-best lists followed by MBR decoding over the segments. We then present an application of the lattice segmentation procedure to segmental MBR decoding over lattices generated by multiple ASR systems (Section 2.4). We finally report the performance of single-system and multiple-system segmental MBR decoders in Section 2.5.

2.1 Segmental Minimum Bayes-Risk Decoding

We first introduce a segmentation rule $R(W)$ which divides strings in the language $\mathcal{W}$ into N segments of zero or more words each. We denote the i^{th} segment of W as $R^i(W)$. In this way, we impose a segmentation of the space $\mathcal{W}$ into segment sets $\mathcal{W}^1 \cdot \mathcal{W}^2 \cdots \mathcal{W}^N$, where

$$\mathcal{W}^i = \{W' : W' = R^i(W), W \in \mathcal{W}\}.$$

When applied to $\mathcal{W}_e$, R generates N evidence segment sets $\mathcal{W}_e^i, i = 1, 2, ..., N$. We now define the marginal probability (i.e. probability of finite dimensional cylinder sets) of any word string $W \in \mathcal{W}_e^i$

$$P_i(W|A) = \sum_{W' \in \mathcal{W}_e : R^i(W') = W} P(W'|A). \tag{2.1}$$

The application of the segmentation rule to the hypothesis space yields hypothesis segment sets $\mathcal{W}_h^i$. The concatenation of the sets $\mathcal{W}_h^i, i = 1, 2, ..., N$ yields a search space $\mathcal{W}_h^*$ that is the cross-product of the hypothesis segment sets $\mathcal{W}_h^i, i = 1, 2, ..., N$. Concatenating the sets may introduce new hypotheses since suffixes can be appended to prefixes in ways that were not permitted in the original space. However no hypotheses are lost through the concatenation. It is our goal to search over this larger space and, by considering more hypotheses, possibly achieve improved performance.

We now discuss the inclusion of the segmentation rule into MBR decoding. We begin by making the strong assumption that the loss function between any pair of

evidence and hypothesis strings $W \in \mathcal{W}_e, W' \in \mathcal{W}_h$ distributes over the segmentation, i.e.

$$l(W, W') = \sum_{i=1}^{N} l(R^i(W), R^i(W')). \tag{2.2}$$

Under this assumption, we can now state the following proposition [34].

Proposition. An utterance level MBR recognizer given by

$$\delta(A) = \hat{W} = \underset{W' \in \mathcal{W}_h^*}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_e} l(W, W') P(W|A) \tag{2.3}$$

can be implemented as a concatenation

$$\hat{W} = \hat{W}^1 \cdot \hat{W}^2 \cdots \hat{W}^N, \tag{2.4}$$

where

$$\hat{W}^i = \underset{W' \in \mathcal{W}_h^i}{\operatorname{argmin}} \sum_{W \in \mathcal{W}_e^i} l(W, W') P_i(W|A), \ i = 1, 2, ..., N \tag{2.5}$$

This proposition defines the Segmental MBR (SMBR) decoder. Equation 2.5 follows by substituting Equation 2.2 into Equation 2.3.

A special case of segmental MBR recognition is particularly useful in practice. It arises when the strings in the hypothesis and evidence segment sets are restricted to length one or zero, i.e. individual words or the NULL word. We also assume that there is a 0/1 loss function on the segment sets

$$l(w, w') = \begin{cases} 0 \text{ if } w = w' \\ 1 \text{ otherwise.} \end{cases} \tag{2.6}$$

Under these conditions the segmental MBR recognizer of Equation 2.5 becomes

$$\hat{W}^i = \underset{w' \in \mathcal{W}_h^i}{\operatorname{argmax}} \ P(w'|A). \tag{2.7}$$

Equation 2.7 is the maximum a-posteriori decision over each hypothesis segment set. In each segment set the posterior probability of all the words are first computed based on the evidence space, and the word with the highest posterior probability is selected. We call this procedure *segmental MBR voting*. This simplification has been utilized in several N-best list and lattice based hypothesis selection procedures

to improve the recognition word error rate. As discussed in Section 1.3, these procedures [28, 27, 63] can be shown to be approximations of MBR decoders.

This summarizes the relationship between SMBR decoding, MAP decoding and segmental MBR voting. From Equation 2.3, if the hypothesis and the evidence spaces were not segmented, MBR decoding under the 0/1 loss function would lead to the standard MAP rule: $\hat{W} = \text{argmax}_{W' \in \mathcal{W}_h} P(W'|A)$. Introducing hypothesis space segmentation transforms the standard MAP rule to segmental MBR voting as in Equation 2.7.

2.1.1 Induced Loss Functions

For a given loss function, evidence space and hypothesis space, it may not be possible to find a segmentation rule such that Equation 2.2 is satisfied for any pair of hypothesis and evidence strings. However, given any segmentation rule, we can specify an associated *induced* loss function defined as

$$l_I(W, W') = \sum_{i=1}^{N} l(R^i(W), R^i(W')). \tag{2.8}$$

From the discussion of Proposition 1, we see that the segmental MBR recognizer is equivalent to an utterance level MBR recognizer under the loss function l_I. Therefore, the overall performance of the SMBR recognizer under a desired loss function l will depend on how well l_I approximates l.

2.1.2 Trade-offs in Segmental MBR decoding

For ASR, we are particularly interested in the Levenshtein loss function that measures the minimum number of edit-operations (insertions, deletions and substitutions) need to transform a word sequence into another word sequence. Here, a segmentation of the hypothesis and evidence spaces will rule out some string alignments between word sequences. Therefore, under a given segmentation rule, the alignments permitted between any two word strings from $\mathcal{W}_h$ and $\mathcal{W}_e$ might not include the optimal alignment needed to achieve the Levenshtein distance. Therefore, the choice of a

given segmentation involves a trade-off between two types of errors: search errors from MBR decoding on large segment sets and the errors in approximating the loss function due to the segmentation.

The Segmental MBR framework does not provide actual hypothesis and evidence space segmentation rules; it only specifies the constraints that these rules must obey. The construction of segment sets therefore remains a design problem to be addressed in an application specific manner. In the following sections we present procedures to construct the segment sets from recognition lattices under the Levenshtein loss function.

2.2 SMBR Lattice Segmentation

A recognition lattice is essentially a compact representation for very large N-best lists and their likelihoods. Formally, it is a directed acyclic graph, or an acyclic weighted finite state acceptor (WFSA) [73] $\mathcal{W} = (Q, \Lambda, q_s, F, \mathcal{T})$ with a finite set of states (nodes) Q, a set of transition labels Λ, an initial state $q_s \in Q$, the set of final states $F \subseteq Q$, and a finite set of transitions $\mathcal{T}$. The set Λ is the vocabulary of the recognizer. A transition belonging to $\mathcal{T}$ is given by $t = (p, q, w, s)$ where $p \in Q$ is the starting state of this transition, $q \in Q$ is the ending state, $w \in \Lambda$ is a word, and s is a real number that represents a 'cost' of this transition. s is often computed as the sum of the negative log acoustic and language model scores on the transition. Some of the transitions in the WFSA may carry the empty string $w = \epsilon$; these are termed ϵ transitions. A complete path through the WFSA is a sequence of transitions given by $T = \{(p_k, q_k, w_k, s_k)\}_{k=1}^{n}$ such that $p_1 = q_s$, $p_{i+1} = q_i$, and $q_n \in F$. The word string associated with T is w_1^n. For this word string we can obtain the joint acoustic and language model log-likelihood as $logP(w_1^n, A) = -\sum_{k=1}^{n} s_k$. In this work the finite state operations are performed using the AT&T Finite State Toolkit [72].

It is conceptually possible to enumerate all lattice paths and explicitly compute the MBR hypothesis according to Equation 1.4 [92]. However, for most large vocabulary ASR systems it is computationally intractable to do so. Goel et. al. [33] described an A^* search algorithm that utilizes the lattice structure to search for the MBR

word string. Building on that approach, we present lattice node based segmentation procedures in which each segment maintains a compact lattice structure.

2.2.1 Lattice Segmentation using Node Sets

The ASR word lattices are directed and typically acyclic, therefore they impose a partial ordering on the lattice nodes. We say $n_1 \leq n_2$ if either $n_1 = n_2$ or there is at least one path connecting nodes n_1 and n_2 in the lattice and n_1 precedes n_2 on this path.

Let (N_s, N_e) be an ordered pair of lattice node sets such that

P1. For all nodes $n \in \mathcal{Q}$, there is at least one node $n' \in N_s$ such that $n \leq n'$ or $n' \leq n$.

P2. For all nodes $n \in \mathcal{Q}$, there is at least one node $n' \in N_e$ such that $n \leq n'$ or $n' \leq n$.

P3. For any $n \in N_e$, there is no node $n' \in N_s$ such that $n \leq n'$.

Properties P1 and P2 essentially state that all lattice paths from lattice start to lattice end pass through at least one node of N_s and one node of N_e. Property P3 says that nodes of N_s on any lattice path precede nodes of N_e on that path. An example of N_s and N_e is depicted in the top panel of Figure 2.1.

Each lattice path can now be uniquely segmented into three parts by finding its first node that belongs to N_s and its first node that belongs to N_e. The portion of the path from q_s to the first node belonging to N_s is the first segment; from the first node belonging to N_s to the first node belonging to N_e is the second segment; and from the first node belonging to N_e to a node in F is the third segment.

Segmentation of each lattice path, based on node sets $\{q_s\}, N_s, N_e, F$, defines a segmentation rule R to divide the entire lattice into three parts. In general, a rule for segmenting the lattice into $n+1$ segments is defined by a sequence of lattice node sets $\{q_s\}, N_1, N_2, ..., N_n, F$ such that all ordered pairs $(N_i, N_{i+1}),\ i = 1, ..., n-1$ obey P1-P3. The i^{th} lattice segment, $\mathcal{W}_i$, is specified by the node sets N_{i-1} and N_i. We

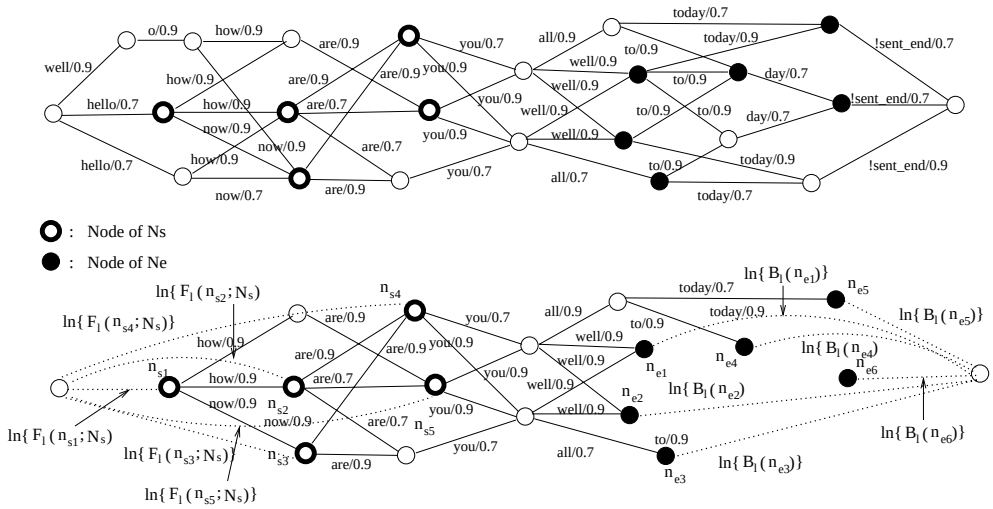

Figure 2.1: Cutting a lattice based on node sets N_s and N_e (top). The lattice segment bounded by these sets is shown in the bottom panel by solid line paths.

shall say it is bounded on the left by N_{i-1} and on the right by N_i. An example lattice segment bounded by N_s and N_e is shown in the bottom panel of Figure 2.1. We call such node based lattice segmentation *lattice cutting* and the lattice cutting node sets as *cut sets*.

We note that lattice cutting yields segment sets $\mathcal{W}_i$ that are more constrained than those that could be obtained by explicitly enumerating all lattice paths and segmenting them. This is due to the sharing of nodes between lattice paths. However, a useful property of lattice cutting is that each segment retains the compact lattice format. This allows for efficient implementation of MBR search on each lattice segment.

We now show that for Levenshtein loss function, fewer lattice segments necessarily result in a better approximation by the induced loss to the actual loss. Suppose we have a collection of cut sets $C = \{\mathcal{N}_i\}_{i=0}^{R}$ where $\mathcal{N}_0 = q_s$ and $\mathcal{N}_R = F$. This collection identifies a segmentation rule such that the induced loss between $W, W' \in \mathcal{W}$ under the segmentation is $l_C(W, W') = \sum_{i=1}^{R} l(W_i, W_i')$.

Suppose we discard a cut set $\mathcal{N}_j$ from C to form $C' = C - \{\mathcal{N}_j\}$. This defines a

new induced loss function

$$l_{C'}(W, W') = \sum_{i=1\ ,i\neq j\ ,i\neq j+1}^{R} l(W_i, W'_i) + l(W_j \cdot W_{j+1}, W'_j \cdot W'_{j+1}) \tag{2.9}$$

By the definition of the Levenshtein distance (Appendix in [33])

$$l(W_j \cdot W_{j+1}, W'_j \cdot W'_{j+1}) \leq l(W_j, W'_j) + l(W_{j+1}, W'_{j+1}). \tag{2.10}$$

Hence $l_{C'}(W, W') \leq l_C(W, W')$. Therefore, if we segment the lattice along fewer cut sets, we obtain successively better approximations to the Levenshtein distance. However as the sizes of the lattice segments increase, SMBR decoding on the resulting segment sets will inevitably encounter more search errors. Our goal is therefore to choose a set C that will yield a "good" cutting procedure. Such a cutting procedure produces small segments that still provide a good approximation of the overall loss.

In the following subsection we describe a risk-driven procedure to identify good cut sets.

2.2.2 Cut Set Selection Based on Total Risk

Our lattice cutting procedure is motivated by the observation that under an ideal segmentation the risk of each hypothesis word string is unchanged after segmentation [46]. The risk after the segmentation is computed under the marginal distribution of Equation 2.1. Consequently the total risk of all lattice hypotheses

$$R_T = \sum_{W' \in \mathcal{W}} \sum_{W \in \mathcal{W}} l(W, W') P(W|A) \tag{2.11}$$

would also be unchanged under this segmentation.

We assume that the posterior probability of the most likely lattice word string dominates the total risk computation. That is

$$R_T \approx \sum_{W' \in \mathcal{W}} l(\tilde{W}, W') P(\tilde{W}|A), \tag{2.12}$$

where $\tilde{W}$ denotes the MAP word string in the lattice

$$\tilde{W} = \underset{W \in \mathcal{W}}{\operatorname{argmax}}\, P(W|A), \tag{2.13}$$

and $\tilde{W} = \tilde{w}_1^K$. Our goal, then, is to find a segmentation rule so that under the ML approximation to the total risk, the following holds

$$P(\tilde{W}|A) \sum_{W' \in \mathcal{W}} l(\tilde{W}, W') = P(\tilde{W}|A) \sum_{W' \in \mathcal{W}} \sum_{i=1}^{K} l(\tilde{W}_i, W'_i). \tag{2.14}$$

Clearly if the rule segments $\tilde{W}$ and each $W' \in \mathcal{W}$ into K substrings so that

$$l(\tilde{W}, W') = \sum_{i=1}^{K} l(\tilde{W}_i, W'_i). \tag{2.15}$$

then Equation 2.14 holds. In the following we describe how such a rule can be derived by first producing a *simultaneous* alignment of all word strings in $\mathcal{W}$ against $\tilde{W}$ and then identifying cut sets in that lattice.

Lattice to Word String Alignment via Finite State Composition

Consider the simple weighted lattice shown in Figure 2.2. We obtain an unweighted acceptor $\mathcal{W}_0$ from this lattice by zeroing the scores on all lattice transitions. We also represent the MAP word string $\tilde{W} = \tilde{w}_1^K$ as an unweighted finite state acceptor whose transitions are given as $t = \{p, q, v\}$ where $v = \tilde{w}_k.k$; this labeling keeps track of both the words and their position in $\tilde{W}$.

To compute the Levenshtein distance between $\tilde{W}$ and the word sequences in $\mathcal{W}$, the possible single-symbol edit operations (insertion, deletion, substitution) and their costs can be readily represented by a simple weighted transducer T [74]. T is constructed to respect the position of words in $\tilde{W}$ (See Figure 2.3). Furthermore, we can reduce the size of this transducer by including only the transductions that map words on the transitions of $\mathcal{W}_0$ to the words in the MAP hypothesis $\tilde{W}$.

We can now obtain all possible alignments between $W \in \mathcal{W}_0$ and $\tilde{W}$ by the weighted finite state composition

$$\mathcal{A} = \mathcal{W}_0 \circ T \circ \tilde{W}. \tag{2.16}$$

Constructed in this way, every path in $\mathcal{A}$ specifies a word sequence $W \in \mathcal{W}$ and a sequence of string-edit operations that transform W to $\tilde{W}$. In its entirety, $\mathcal{A}$ specifies

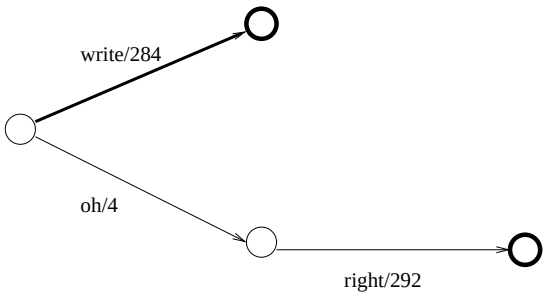

Figure 2.2: A sample word lattice. The MAP hypothesis is shown in bold.

all possible string-edit operations that transform all word strings in $\mathcal{W}$ to $\tilde{W}$ (See Figure 2.4).

$\mathcal{A}$ has transitions $t = (p, q, a, s)$ where a denotes an input-output symbol pair (w, v). There are three types of transitions: (1) $w \neq \epsilon$ and $v = \tilde{w}_i.i$ which indicates a substitution of word w by word $\tilde{w}_i$; (2) $w \neq \epsilon$ and $v = \epsilon$ indicates that word w is an insertion; (3) $w = \epsilon$ and $v = \tilde{w}_i.i$ shows a deletion with respect to $\tilde{w}_i$. The costs s on the transitions of A arise from the composition in Equation 2.16.

Compact Representation of String Alignments

We now wish to extract from $\mathcal{A}$ the Levenshtein alignment between every path $W \in \mathcal{W}$ and $\tilde{W}$. This can be done in two steps. We first perform a sequence of operations that transforms $\mathcal{A}$ into a weighted acceptor $\mathcal{A}'$. $\mathcal{A}'$ contains all the alignments links in $\mathcal{A}$, but represented in simplified form as an acceptor. We next use a variant of dynamic programming algorithm on the acceptor $\mathcal{A}'$ to extract the Levenshtein alignment between $\tilde{W}$ and every word string that was originally in $\mathcal{W}$.

The transformation of $\mathcal{A}$ into $\mathcal{A}'$ is as follows.

1. Project alignment information onto the input labels of $\mathcal{A}$, as follows :

    - Sort the states of $\mathcal{A}$ topologically and insert them in a queue S. Associate with each state q an integer $V(q)$. A value of $V(q) = i$ would indicate that all partial lattice paths ending at state q have been aligned with respect to $\tilde{w}_1^{i-1}$. Set $V(q_s) = 1$.

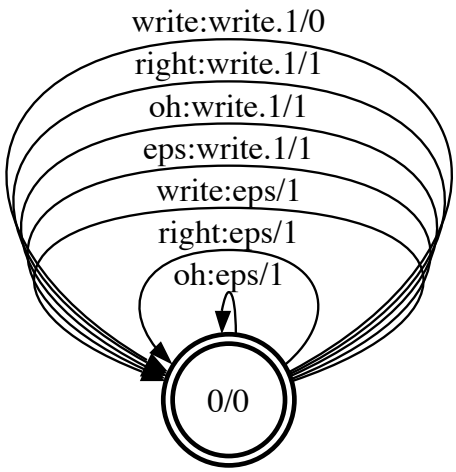

Figure 2.3: The string-edit transducer T for computing Levenshtein distance between word strings in the lattice shown in Figure 2.2 and the MAP hypothesis from the lattice (shown in bold). Each transition in T has the format $x : y/c$, which indicates that the input label is x, output label is y, and the cost of mapping x to y is c.

- While S is non-empty
  (a) $p \leftarrow \text{head}(S)$. DEQUEUE($S$).
  (b) For all transitions $t = (p, q, a, s)$ leaving p, perform one of the following:
    i. *Substitution:* If $a = (w, v)$ has $w \neq \epsilon$, $v = \tilde{w}_i.i$, set $a = (w.i, v)$ and $V(q) = i + 1$.
    ii. *Deletion:* If $a = (w, v)$ has $w = \epsilon$ and $v = \tilde{w}_i.i$, set $V(q) = i + 1$.
    iii. *Insertion:* If $a = (w, v)$ has $w \neq \epsilon$ and $v = \epsilon$, set $w = w_\epsilon.V(p)$ and $V(q) = V(p)$.

2. Convert the resulting transducer from Step 1 into an acceptor by projecting

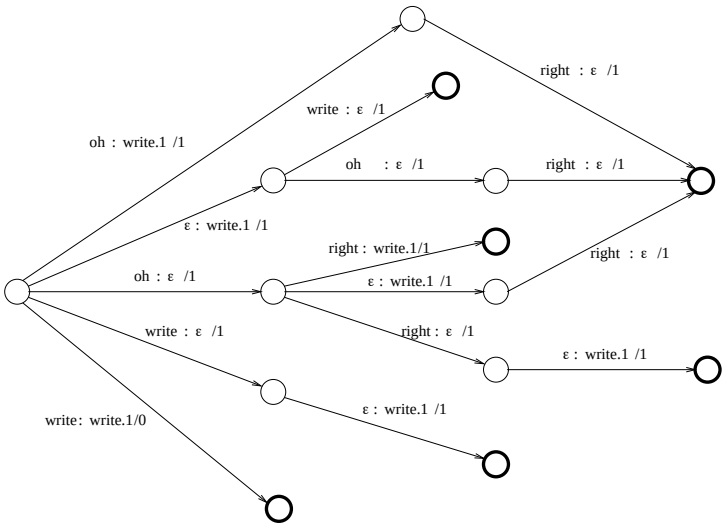

Figure 2.4: The transducer $\mathcal{A}$ for the lattice in Figure 2.2.

onto the input labels.

3. For the weighted automaton generated in Step 2, generate an equivalent weighted automaton without ϵ-transitions [69].

These three operations transform $\mathcal{A}$ into a weighted acceptor $\mathcal{A}'$ that contains the cost of *all alignments* between all lattice word strings and the MAP path (See Figure 2.5). We now relate the properties of the lattice $\mathcal{W}$ and the finite state machines $\mathcal{A}$ and $\mathcal{A}'$. By construction, corresponding to any $\bar{W} \in \mathcal{W}$ where $\bar{W} = \bar{w}_1^n$, there exist paths $T \in \mathcal{A}$ such that

1. $T = \{(p_k, q_k, a_k, s_k)\}_{k=1}^m$, $m \geq n$, $a_k = (w_k, v_k)$ where $w_1^m = \bar{w}_1^n$ if ϵ's in w_1^m are removed and $v_1^m = \tilde{w}_1^K$ if ϵ's in v_1^m are removed. $\sum_{k=1}^m s_k$ is total cost of the alignment specified by T.

s_k is the cost of a transition on T. Furthermore, for each $T \in \mathcal{A}$ there is a corresponding $T' \in \mathcal{A}'$ that specifies the identical alignment. That is,

2. $T' = \{(p'_k, q'_k, v'_k, s'_k)\}_{k=1}^n$ where $\text{Cost}(T) = \sum_{k=1}^m s_k = \sum_{l=1}^n s'_l = \text{Cost}(T')$ is the string edit distance between $\tilde{w}_1^K$ and $\bar{w}_1^n$ along the alignment specified by T and T', and $v_1^{'m} = \bar{w}_1^n$ if each v'_k is stripped of its $\cdot i$ and ϵ subscripts.

s'_k is the cost of a transition on T'.

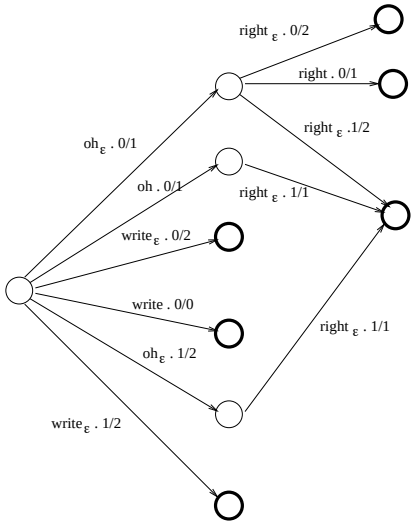

Figure 2.5: The acceptor $\mathcal{A}'$ for the lattice in Figure 2.2.

Optimal Computation of Lattice to Word String Alignment

We now discuss a procedure to extract the optimal alignment between paths $W \in \mathcal{W}$ and $\tilde{W}$. We first note that if $\mathcal{A}'$ contained the alignment of only one word string W against $\tilde{W}$, we could find the desired optimal alignment through a standard dynamic programming procedure [6, 89, 70] that traverses the nodes of $\mathcal{A}'$ in topologically sorted order and retains backpointers to the optimal partial paths to all nodes. However, since $\mathcal{A}'$ contains alignments of multiple strings against $\tilde{W}$, we need to extend the dynamic programming procedure to keep track of the identity of word strings leading into nodes. This is described in the following procedure.

1. Sort the nodes of $\mathcal{A}'$ in reverse topological order (i.e. lattice final nodes first) and insert them in a queue S. For each node $q \in S$, let $b_q(y)$ denote the minimum cost of all paths that lead from node q to the lattice end node and carry the word string y. Let $a_q(y)$ be the immediate successor node of q on the path that achieves $b_q(y)$.

2. For each final node f of $\mathcal{A}'$, set $b_f(\epsilon) = 0$.

3. While S is non-empty

   (a) $p \leftarrow \text{head}(S)$. DEQUEUE($S$).

   (b) Let T denote the set of lattice transitions $t = (p, q, v, s)$ leaving state p. v is either $w.i$ or $v = w_\epsilon.i$. Let Y denote the set of unique word strings on the paths starting from state p. The word string $y = (w \cdot z)$ starts with the word w and has a suffix z.

   (c) For each $y(= w \cdot z) \in Y$,

      i. Compute:

$$\hat{t} = \underset{t \in T : t \text{ has word label } w}{\text{argmin}} \; s + b_q(z)$$

      Denote $\hat{t} = (p, \hat{q}, \hat{v}, \hat{s})$.

      ii. $b_p(y) = \hat{s} + b_{\hat{q}}(z)$. $\quad a_p(y) = \hat{q}$.

   Step 3 prunes all transitions leaving p that are not needed for any optimal alignment passing through p.

4. The procedure terminates upon reaching the start node q_s of $\mathcal{A}'$. The optimal alignment cost of each complete path y can be readily obtained from $b_{q_s}(y)$, and the complete alignment can be obtained by following the backtrace pointers stored in $a_p(y)$ arrays.

An Efficient Algorithm for Lattice to Word String Alignment

The alignment procedure described in the previous section involves the computation of the cost $b_p(y)$ for each state p in $\mathcal{A}'$. This cost is computed for all unique word strings y leaving state q. Therefore, it involves enumerating all the word sub-strings in the word lattice $\mathcal{W}$. While this is definitely impossible for most word lattices of interest, this description does clearly present the inherent complexity of the lattice to string alignment problem. In practice, we do not retain the cost $b_p(y)$ for all word sequences leaving p. For each state p, we approximate $b_p(y)$ as $b_p(y) \approx b_p^* = \min_y b_p(y) \; \forall y$ in Step 3(c)ii. We now present the procedure that results from this approximation.

1. Sort the nodes of $\mathcal{A}'$ in reverse topological order (i.e. lattice final nodes first) and insert them in a queue S. For each node $q \in S$, let b_q denote the minimum cost of all paths that lead from node q to the lattice end node.

2. For each final node f of $\mathcal{A}'$, set $b_f = 0$.

3. While S is non-empty

   (a) $p \leftarrow \text{head}(S)$. DEQUEUE($S$).

   (b) Let T denote the set of lattice transitions $t = (p, q, v, s)$ leaving state p. v is either $w.i$ or $v = w_\epsilon.i$. Let U denote the set of unique words on the transitions starting from state p.

   (c) Initialize $\hat{T} = \{\}$.

   (d) For each $w \in U$,

      i. Compute:

$$\hat{t} = \underset{t \in T : t \text{ has word label } w}{\text{argmin}} \; s + b_q$$

      Denote $\hat{t} = (p, \hat{q}, \hat{v}, \hat{s})$.

      ii. $\hat{T} \leftarrow \hat{t}$.

   (e) Compute:

$$b_p = \min_{\hat{t} \in \hat{T}} \hat{s} + b_{\hat{q}}$$

   (f) Prune transitions $t \in T$ and $t \notin \hat{T}$.

4. The procedure terminates upon reaching the start node q_s of $\mathcal{A}'$.

As a result of the simplification, the information maintained by the $a_p(w)$ and the $b_p(w)$ arrays can be stored with the lattice structure of $\mathcal{A}'$. This is therefore a pruning procedure of $\mathcal{A}'$ and we call the resulting acceptor $\hat{\mathcal{A}}$. For the example $\mathcal{A}'$ of Figure 2.5, $\hat{\mathcal{A}}$ is shown in Figure 2.6.

The transitions of $\hat{\mathcal{A}}$ have the form $t = (p, q, v, s)$. Either (a) $v = w.i$ that indicates the word w has aligned with $\tilde{w}_i$ (substitution) or (b) $v = w_\epsilon.i$ indicates that word w occurs as an insertion before $\tilde{w}_i$. We can insert ϵ-transitions whenever the partial path

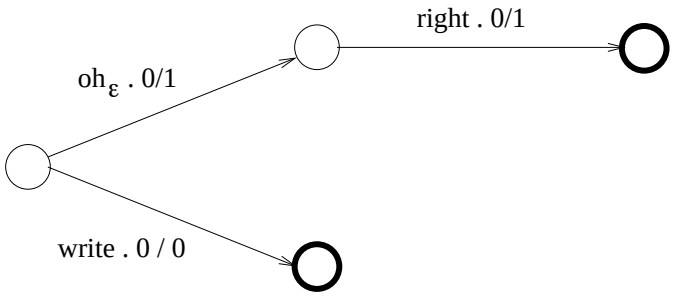

Figure 2.6: The acceptor $\hat{\mathcal{A}}$ for the lattice in Figure 2.5.

ending on state q has aligned with $\tilde{w}_1^i$ and the partial path ending on q', a successor node of q has aligned with $\tilde{w}_1^{i+2}$. This will allow for deletions.

We note that the acceptors $\mathcal{W}$ and $\hat{\mathcal{A}}$ have identical word sequences, therefore, we can get the acoustic and language model scores for $\hat{\mathcal{A}}$ by composing it with $\mathcal{W}$.

Risk Based Lattice Cutting

Referring back to Section 2.2.1, we introduce lattice segmentation as the process of identifying lattice cut sets to satisfy property P1-P3. The process of generating $\hat{\mathcal{A}}$ identifies a correspondence between each word in $\tilde{W}$ and paths in $\hat{\mathcal{A}}$. Each word $\tilde{w}_i$ in $\tilde{W}$ is aligned with a collection of arcs in $\hat{\mathcal{A}}$. These arcs either fall on distinct paths (e.g. hello.0 in Figure 2.7) or form connected subpaths (e.g. well$_\epsilon$.0 $\cdot$ o.0 in Figure 2.7). For each word $\tilde{w}_i$, we define the lattice cut node set $\mathcal{N}_i$ as the terminal nodes of all the subpaths that align with $\tilde{w}_i$. This defines K cut sets if there are K words in $\tilde{W}$. We also define $\mathcal{N}_0$ as $\{q_s\}$.

In this way we use the alignment information provided in $\hat{\mathcal{A}}$ to define the lattice cut sets that segment the lattice into K sublattices. We call this procedure *Risk-Based Lattice Cutting*. This procedure ensures that every lattice path passes through exactly one node from each lattice node cut set. A determinized version of the lattice from Figure 2.1 and its acceptor $\hat{\mathcal{A}}$ are shown in the top and bottom panels of Figure 2.7. The bottom panel also displays the cuts obtained along the node sets.

The segmentation procedure, modulo the errors introduced by the approximate

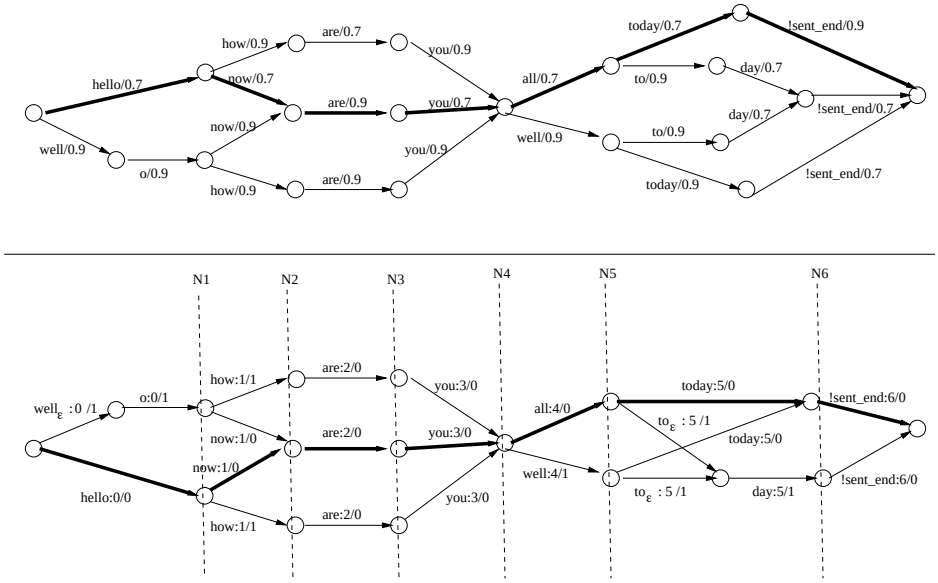

Figure 2.7: (top) A word lattice $\mathcal{W}$ and (bottom) its acceptor $\hat{\mathcal{A}}$ showing the Levenshtein alignment between $W \in \mathcal{W}$ and $\tilde{W}$ (shown as the path in bold). The bottom panel shows the segmentation along the 6 nodesets obtained by the risk-based lattice cutting procedure.

procedure used to generate $\hat{\mathcal{A}}$, is optimal with respect to the MAP word hypothesis. Every path $W' \in \mathcal{A}$ has a corresponding path $\{p_k, q_k, w_k, s_k\}_{k=1}^{N}$ in $\hat{\mathcal{A}}$ such that $l(\tilde{W}, W') = \sum_{i=1}^{K} l(\tilde{W}_i, W'_i) = \sum_{k=1}^{K} s_k$. In this way, the costs in $\hat{\mathcal{A}}$ agree with the loss function desired in Risk-based lattice cutting.

Periodic Risk Based Lattice Cutting

The alignment obtained in Section 2.2.2 is ensured to be optimal only relative to the MAP path. It is not guaranteed that $l(W, W') = l_I(W, W')$ for $W \neq \tilde{W}$. Following the discussion in Section 2.2.1, we note that if we segment the lattice along fewer cut sets, we obtain better approximations to the Levenshtein loss function. However, this leads to larger lattice segments and therefore greater search errors in MBR decoding.

One solution that attempts to balance the trade-off between search errors and errors in approximating the loss function is to segment the lattice by choosing node sets N_i at equal intervals or periods. A period of k specifies the cut sets N_1, N_{k+1},

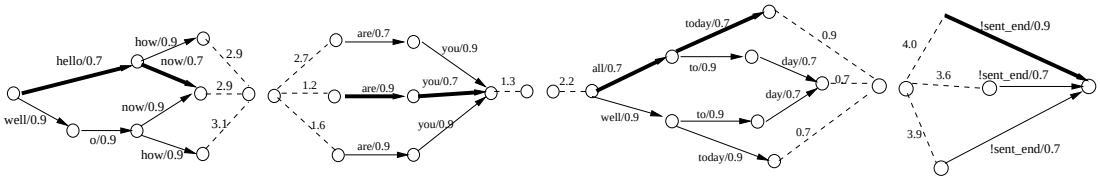

Figure 2.8: Lattice segments obtained by periodic risk-based lattice cutting on the lattice from Figure 2.7 (Period = 2).

N_{2k+1} and so on. Therefore the set $C = \{q_s, N_1, N_{k+1}, ..., N_{nk+1}, F\}$ where n is the largest integer so that periodic cuts can be found. We call this procedure *Periodic risk-based Lattice Cutting* (PLC). If the loss function approximation obtained by cutting $\mathcal{W}$ into K segment sets is good, the optimal cutting period k tends to be smaller and vice-versa. The choice of the cutting period is found experimentally to reduce the word error rate on a development set. We note that the lattice cutting procedure described in the previous section is identical to the PLC procedure with period 1 (PLC-1). Figure 2.8 shows the sub-lattices obtained by periodic risk-based lattice cutting on the lattice from Figure 2.7.

2.2.3 SMBR Decoding of a Lattice Segment

To generate SMBR hypothesis from a lattice segment (Equation 2.5) we require $P_i(W|A)$ for each word string in that segment. $P_i(W|A)$ can be computed by summing over all paths in the lattice whose subpath in $\mathcal{W}_i$ is W. We will now describe the lattice forward-backward procedure [104] to calculate this probability.

Let W be a complete path in the lattice and let W_p be a prefix of W. We use $L_f(W_p)$ to denote the joint log-likelihood of observing W_p and the acoustic segment that corresponds to W_p. $L_f(W_p)$ can be obtained by summing the log acoustic and language model scores present on the lattice links that correspond to W_p. Similarly, for a suffix W_s of W, we use $L_b(W_s)$ to denote the joint log-likelihood of observing W_s together with its corresponding acoustic segment, conditioned on the starting node of W_s. $P(A)$ can be computed as $e^{L_f(n_e)}$.

Let $E^h(W')$ denote the first node of an arbitrary lattice path segment W'. Let

$E^f(W')$ denote the last node of W', and let $E(W')$ be the set of all lattice nodes through which W' passes, including $E^h(W')$ and $E^f(W')$. Let W_i be a path in a lattice segment $\mathcal{W}_i$ bounded by node sets N_s and N_e. Let $n_1 = E^h(W_i)$ and $n_2 = E^f(W_i)$. We first define a *lattice forward probability* of n_1, $F(n_1)$, which is the sum of partial path probabilities of all partial lattice paths ending at n_1. That is,

$$F(n_1) = \sum_{W_p : E^f(W_p) = n_1} e^{L_f(W_p)}. \tag{2.17}$$

We also define *lattice backward probability* of the final node of W_i, using the backward log-likelihood $L_b(W_s)$, as

$$B(n_2) = \sum_{W_s : E^h(W_s) = n_2} e^{L_b(W_s)}. \tag{2.18}$$

Using the forward probability of n_1 and lattice backward probability of n_2, the marginal probability of W_i can be computed as

$$P_i(W_i|A) = \frac{1}{P(A)} F(n_1) P(W_i, A(W_i)|n_1) B(n_2), \tag{2.19}$$

where $A(W_i)$ denotes the acoustic segment corresponding to W_i.

Having obtained $P_i(W|A)$, the SMBR hypothesis can be computed using the A^* search procedure described by Goel et. al. [33]. Alternatively, an N-best list can be generated from each segment and N-best rescoring procedure of Stolcke et. al. [92] can be used. In the following section we describe a third MBR procedure that can be used to rescore the N-best lists generated from each lattice segment.

2.3 SMBR N-Best List Segmentation

We start with a description of N-best list that is an enumeration of N most likely word strings given an acoustic observation; it can be generated from a lattice as word strings with N highest log likelihood values. An N-best list can itself be considered as a special "linear" lattice where each node, except the start and end nodes, has exactly one incoming and one outgoing transition. An example N-best list derived from the lattice of Figure 2.7 is displayed as a linear lattice in Figure 2.9.

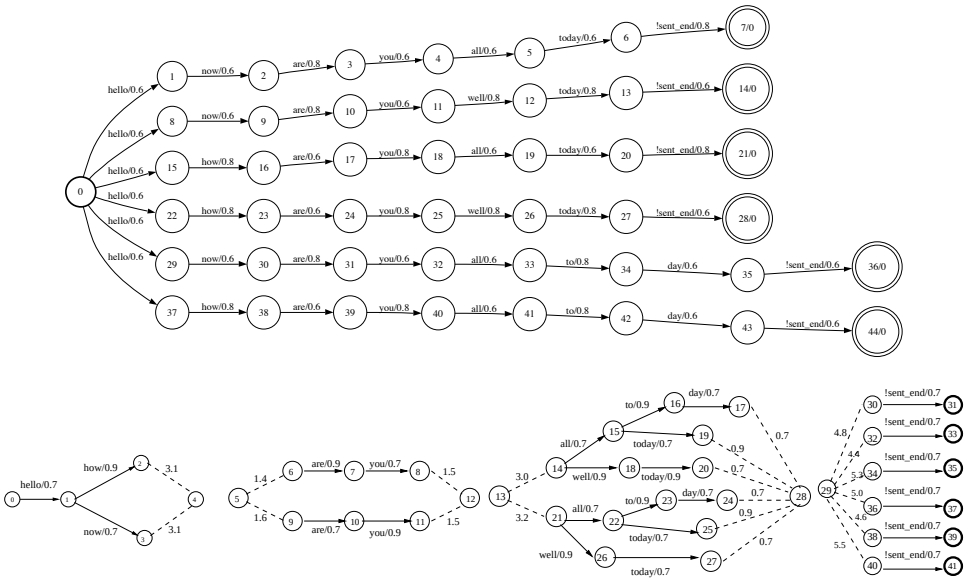

Figure 2.9: N-best List Segmentation using Periodic Lattice Cutting (Period=2). The top panel shows an N-best list generated from the lattice in Figure 2.7. The bottom panel displays the segments obtained by applying the PLC procedure to the N-best list.

For SMBR rescoring of an N-best list we can apply the periodic lattice cutting procedure described in Section 2.2.2 (bottom panel of Figure 2.9), and then apply previously developed MBR implementations [33] to obtain the SMBR hypothesis from each segment. We now discuss two alternate SMBR decoding procedures on N-best lists.

2.3.1 N-best ROVER

Our first SMBR procedure is a variant of ROVER [28], a classifier combination procedure used in speech recognition. ROVER combines the single best outputs produced by multiple speech recognizers to produce a consensus hypothesis. The consensus hypothesis generated by ROVER has been shown to yield a significant reduction in word error rate relative to each of the individual ASR systems that are combined. An extension to the ROVER procedure combines multiple outputs from each ASR system [91]; we call this procedure N-best ROVER.

We here describe the N-best ROVER procedure for combining N-best hypotheses from a single system; this procedure can be easily extended to combine hypotheses from multiple systems (as described in [32]).

1. Construct a Word Transition Network (WTN) from the N-best outputs. The WTN represents a simultaneous alignment of the N best hypotheses. An initial WTN is first produced by performing a Levenshtein alignment of the top-2 hypotheses in the N-best list. This network is grown by iteratively adding each new hypothesis by aligning it to the WTN constructed so far. The procedure is terminated when all the hypotheses in the N-best list have been added to the WTN. Figure 2.10 shows the WTN constructed from the N-best list in Figure 2.9. We use the term *correspondence set* to refer to the set of words in the WTN that align with each other.

2. Using the distribution $P(W|A)$ over the N-best list and the WTN, compute a marginal probability according to Equation 2.1 for each word in each correspondence set.

3. From each correspondence set, select the word with highest posterior probability. Concatenate these words to produce the final hypothesis.

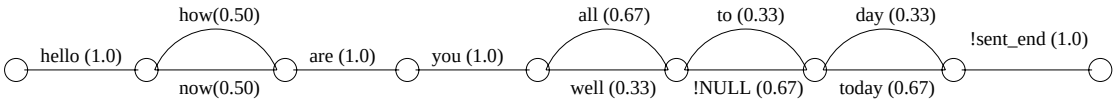

Figure 2.10: Word Transition Network constructed from the N-best list in Figure 2.9.

To show that N-best ROVER is an instance of segmental MBR voting, we must identify the evidence segment sets $\mathcal{W}_e^i$, the hypothesis segment sets $\mathcal{W}_h^i$, and the evidence distribution $P(W|A)$ that underly this procedure.

We first note that the evidence space of N-best ROVER is the N-best list. The evidence distribution is the distribution $P(W|A)$ over the N-best list specified by the ASR system. The correspondence sets play the role of both the evidence and the

hypothesis segment sets. Finally the hypothesis space is the set of all the paths that are contained in the WTN.

The induced loss function in N-best ROVER is

$$l_R(W, W') = \sum_{i=1}^{M} l_{0/1}(w^i, w'^i). \tag{2.20}$$

Since the WTN is constructed to get a good simultaneous alignment between hypotheses, l_R approximates the Levenshtein distance. Hence, N-best ROVER is a segmental MBR voting procedure under an approximate Levenshtein loss function. We also note that N-best ROVER is similar to risk based cutting of N-best lists with the most significant difference being that the risk based lattice cutting allows for multiple consecutive words in each segment set (Figure 2.7); in contrast, ROVER yields at most one consecutive word in a segment set.

2.3.2 Extended ROVER

We now describe a second N-best list SMBR rescoring procedure that generalizes both ROVER and risk based lattice cutting; we call this procedure Extended ROVER (e-ROVER). We first define a process of *joining* two consecutive segment sets. In joining two segment sets we replace those two sets by one *expanded* set that contains all the paths from the original pair of sets. This is illustrated in Figure 2.11.

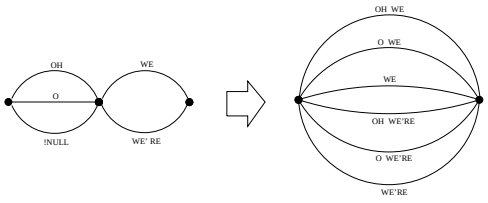

Figure 2.11: The process of joining two segment sets to create an expanded set in Extended ROVER.

The e-ROVER procedure can be described as follows [35].

1. Construct the Word Transition Network (WTN) from the N-best hypotheses of the single system.

2. Using the distribution $P(W|A)$ over the N-best list and the WTN, compute a marginal probability according to Equation 2.1 for each word in each correspondence set.

3. If the largest value of the posterior probability in a segment set is above a threshold, collapse the segment to the most likely word; we call this procedure "pinching". Join all adjacent unpinched segment sets.

The procedure of pinching and expanding the segment sets is shown in Figure 2.12. Hypotheses in e-ROVER are formed sequentially according to Equations 2.4 and 2.5.

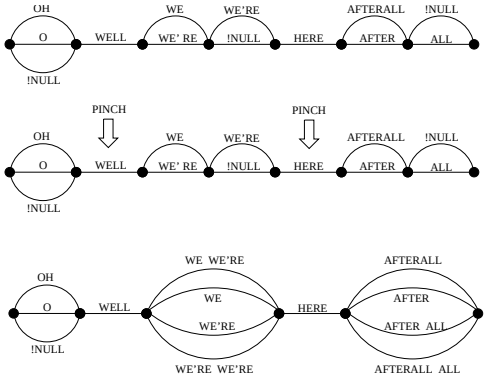

Figure 2.12: Word Transition Network Construction in Extended ROVER.

The hypothesis and evidence spaces in e-ROVER are identical to those in ROVER. However, the e-ROVER procedure results in fewer hypothesis and evidence segments relative to N-best ROVER. From Section 2.2.1, we recall that under the Levenshtein loss function, fewer segments necessarily result in a better approximation by the induced loss to the actual loss. Therefore the loss function in e-ROVER provides a better approximation to the Levenshtein distance. Since they are both instantiations of Equation 2.4, e-ROVER directly extends ROVER and would be reasonably expected to yield a lower word error rate.

In comparing e-ROVER to MBR rescoring of N-best lists [92], we note that both procedures have the same evidence space (hypotheses in the N-best list) but the hypothesis space in e-ROVER is larger due to the expansion of segment sets described

above. It is also worth pointing out that the e-ROVER procedure generalizes risk based lattice cutting by allowing segment sets to contain hypotheses which were not present in the original N-best lists.

2.4 Applications to ASR System Combination

In addition to its role in simplifying MBR decoding, the segmental MBR decoding framework has applications to ASR system combination. These techniques involve combining either word lattices or N-best lists produced by several ASR systems.

Let $\mathcal{W}^k, k = 1, 2, ..., J$ be recognition lattices or N-best lists from J ASR systems. Let $P^k(W|A)$ be the evidence distribution of the k^{th} system over $\mathcal{W}^k$. A common evidence space can be obtained by taking a union or intersection of these K lattices or N-best lists. The evidence distribution over this space can be derived by taking the arithmetic mean

$$P(W|A) = \frac{1}{J} \sum_k P^k(W|A),$$

or a geometric mean

$$P(W|A) = [\prod_k P^k(W|A)]^{\frac{1}{J}}$$

of the J evidence distributions.

We will now describe how lattices from multiple ASR systems can be combined. One possible scheme is described in the following.

1. Select the hypothesis $\hat{W}$ with the overall highest posterior probability among the MAP hypotheses from the J systems. This is obtained as

$$\hat{k} = \underset{k=1,2,...,J}{\operatorname{argmax}} P^k(\hat{W}^k|A) \tag{2.21}$$

$$\hat{W} = \hat{W}^{\hat{k}}. \tag{2.22}$$

2. Segment each lattice with respect to $\hat{W}$ using the periodic risk-based lattice-cutting procedure (Section 2.2.2) into N sections. This gives us $N \times J$ sub-lattices given by $\mathcal{W}_l^k, k = 1, 2, ..., J, l = 1, 2, ..., N$. We note that $\hat{W}$ need not

be present in all of the J lattices since the procedure described in Section 2.2.2 can be used to align the lattice to any word string.

3. For each section $l = 1, 2, ..., N$, we now create new segment-sets by combining the J corresponding sub-lattices $\mathcal{W}_l^k, k = 1, 2, ..., J$. We have considered two combination schemes:

   (a) Perform a weighted finite-state intersection [73] of the corresponding sub-lattices. This is equivalent to multiplying the posterior probability of hypotheses in the individual sub-lattices.

$$\text{For } l = 1, 2, ..., N \tag{2.23}$$

$$\mathcal{W}_l = \cap_{k=1}^{J} \mathcal{W}_l^k \tag{2.24}$$

$$P_l(W|A) = [\prod_{k=1}^{J} P_l^k(W|A)]^{\frac{1}{J}} \ \ \forall\, W \in \mathcal{W}_l. \tag{2.25}$$

   (b) Perform a weighted finite state union of the corresponding sub-lattices followed by a weighted finite state determinization under the $(+, \times)$ semiring [73]. This is equivalent to adding the posterior probability of hypotheses in the individual sub-lattices.

$$\text{For } l = 1, 2, ..., N \tag{2.26}$$

$$\mathcal{W}_l = \cup_{k=1}^{K} \mathcal{W}_l^k \tag{2.27}$$

$$P_l(W|A) = \frac{1}{J} \sum_{k=1}^{J} P_l^k(W|A) \ \ \forall\, W \in \mathcal{W}_l. \tag{2.28}$$

4. Finally, we perform SMBR decoding (Equations 2.5 and 2.4 in Section 2.1) on the sub-lattices $\mathcal{W}_l$ obtained by the above combination schemes.

A schematic of multi-system SMBR decoding using three sets of lattices is shown in Figure 2.13.

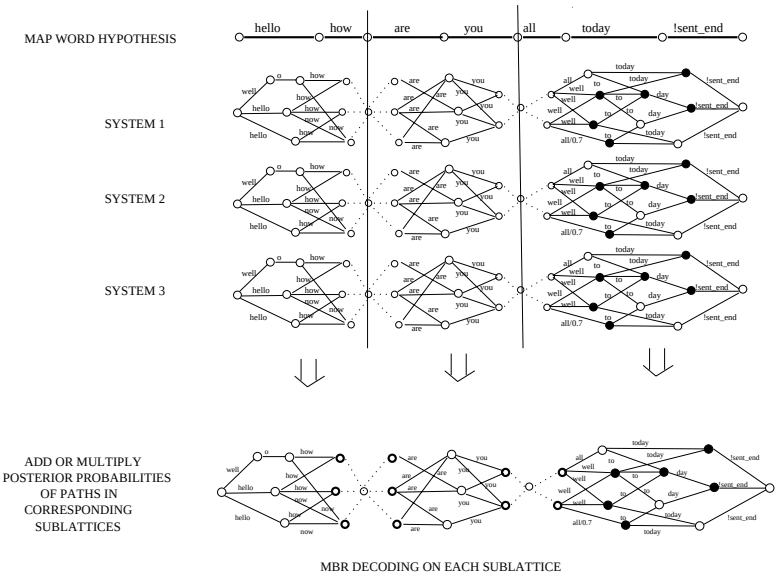

Figure 2.13: Multiple-system SMBR Decoding via Lattice Combination.

2.5 Performance of Lattice Segmentation Procedures

Our SMBR decoding experiments are carried out on Large Vocabulary Conversational Speech Recognition (LVCSR) tasks. We first present performance of SMBR decoders under the risk based lattice cutting procedures described in Section 2.2. We then report performance of the multiple system SMBR decoding schemes presented in Section 2.4.

2.5.1 Single System SMBR Decoding

Our lattice cutting procedures are tested on the Switchboard-2 portion of the 1998 Hub5 evaluation set (SWB2) and Switchboard-1 portion of the 2000 Hub5 evaluation set (SWB1). For both these test sets an initial set of one-best hypotheses is generated using the AT&T large vocabulary decoder [73]. HTK [107] cross-word triphone

acoustic models, trained on VTN-warped data, with a pruned version of SRI 33K trigram word language model [91] are used. The one-best hypotheses are then used to train MLLR transforms, with two regression classes, for speaker adaptive training (SAT) version of the acoustic models. These models are used to generate an initial set of lattices under the language model mentioned above. These lattices are then rescored using the unpruned version of SRI 33K trigram language model and then again using SAT acoustic models with unsupervised MLLR on the test set. Details of the system are given in JHU 2001 LVCSR Hub5 system description [12].

Characterization of Lattice Cutting We here report an experiment to confirm that the risk based lattice segmentation does indeed preserve all the hypotheses in the original lattice. For each utterance we concatenate the sub-lattices produced by lattice segmentation to generate a new search space; we will refer to this new space as the *pinched lattice*.

We measure the Oracle-best Word Error Rate (OWER) of the pinched lattice as a function of the cutting period used in PLC (Figure 2.14). OWER is defined as the word error rate of the hypothesis in the lattice that has the lowest Levenshtein distance relative to the reference transcription. On the SWB2 held out set, the OWER over the original lattices is found to be 24.1%. We observe that the OWER of the pinched lattices increases monotonically as the cutting period in PLC is increased from 1 to 14. At all cutting periods, the OWER of the pinched lattices is lesser than (or equal) to that of the original lattices.

We conclude from this experiment that lattice segmentation does not discard any paths in the original lattice. On the contrary, the pinched lattice contains additional hypotheses relative to the original lattice (as discussed in Section 2.1) and therefore obtains a lower OWER. The number of additional paths introduced in the pinched lattice decreases as the cutting period is increased; this is seen in the monotonic increase of the OWER. At high cutting periods very few lattice segments are generated and almost no new hypotheses added to the pinched lattice; as a result, the pinched lattice yields the same OWER as the original lattice.

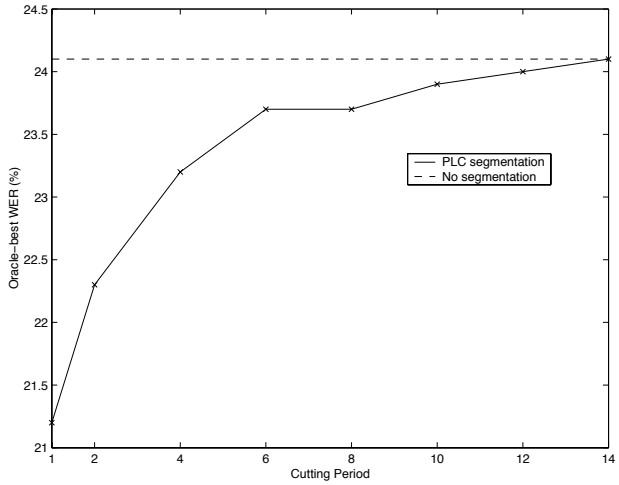

Figure 2.14: Oracle-best Word Error Rate (OWER) of pinched lattices as a function of the cutting period used in Periodic Lattice Cutting. Results are shown on the SWB2 held out set.

SMBR Decoding Experiments In these experiments lattices are segmented using the periodic risk based lattice cutting with periods 1 (PLC-1) and 6 (PLC-6). Once a lattice segmentation was obtained, the following procedures are investigated to compute the SMBR hypothesis. An A^* search over each segment [33] attempts an exact, if heavily pruned, implementation of the MBR decoder. Alternatively, an N-best list is generated from each segment and then rescored using the min-risk procedure [92, 33]. As a third approach, the e-ROVER procedure of Section 2.3 is applied. In the latter two techniques, N-best lists of size 250 are used.

The A^* MBR decoder, in its current implementation [33], is unable to perform MBR decoding over lattice segments that contain deletions relative to the MAP hypothesis. On these segments, we generate N-best lists (of size 250) and rescore the lists under the N-best MBR implementation [92, 33].

For periodic risk based lattice cutting, the optimal segmentation period is determined on two held out sets, one corresponding to each test set. Cutting periods of 1 through 20 are tried and for each segmentation the SMBR hypothesis is generated using one of A^*, N-best list rescoring, or e-ROVER procedures. Figure 2.15 presents the word error rate of the A^* SMBR decoding on the held out set corresponding o the

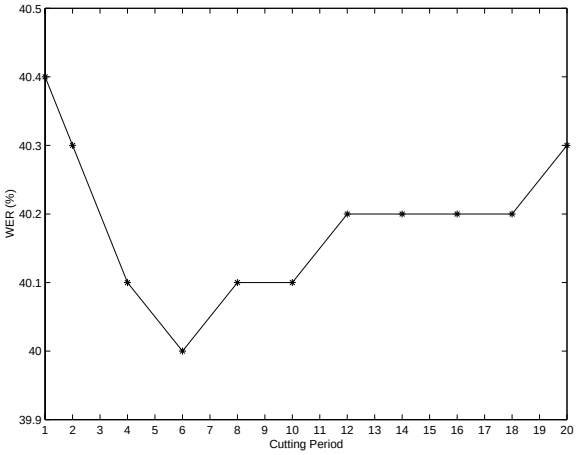

Figure 2.15: Performance of A^* SMBR decoder as a function of the cutting period used in Periodic Lattice Cutting. Results are shown on the SWB2 held out set.

SWB2 test set. As can be seen, the optimal cutting period is 6 on the SWB2 test set. N-best rescoring and e-ROVER also achieve their optimal performance at period 6 on this data set. On the held out set corresponding to the SWB1 test set, the optimal cutting period is found to be 4 under all three hypothesis generation procedures. This suggests that optimal lattice cutting period is relatively insensitive to the hypothesis generation method but should be tuned to the task to which periodic lattice cutting is applied.

Table 2.1 presents a comparison of different lattice segmentation and hypothesis generation procedures. PLC is performed with a cutting period of 6 (on both test sets, even though 4 was found to be optimal for SWB1).

We note that all SMBR procedures yield a gain over the MAP baseline for both test sets. The SMBR decoder under the PLC-1 segmentation does not involve any search errors but obtains nearly the same performance as the MBR decoder on unsegmented lattices. This shows that PLC-1 segmentation does not provide a good approximation of the Levenshtein loss function. At very high cutting periods (e.g. 20) very few lattice segments are created and the performance of the SMBR decoder is almost identical to that of the MBR decoder on unsegmented lattices. The PLC-6 segmentation consistently further improves the word error rate over the no segmentation case which

Decoder		WER(%)	
		SWB2	SWB1
MAP (baseline)		41.1	26.0
SMBR Decoding			
Segmentation Strategy	MBR Decoding Strategy		
No Segmentation	A^* search	40.4	25.5
	e-ROVER	40.5	25.7
	N-best rescoring	40.4	25.6
PLC-1 (Period 1)	A^* search	40.4	25.5
	e-ROVER	40.2	25.5
	N-best rescoring	40.3	25.6
PLC-6 (Period 6)	A^* search	40.0	25.4
	e-ROVER	39.9	25.3
	N-best rescoring	40.1	25.4

Table 2.1: Performance of single system SMBR speech recognizers.

advocates the use of SMBR procedures over MBR decoding without segmentation. Among the various hypothesis generation procedures, PLC with period 6 is found to have the best performance. In all cases, e-ROVER performance is the best among the various MBR procedures.

2.5.2 Multiple System SMBR Decoding

Our experiments with combining lattices from multiple systems and their SMBR decoding are carried out on the development set of the LVCSR RT-02 evaluation. A description of the acoustic and language models used is given in the JHU LVCSR RT-02 system description [11]. In this system, MMI acoustic models are used to generate an initial set of lattices under the SRI 33K trigram language model [91]. These lattices are then rescored with DLLT acoustic models and DSAT acoustic models [96] to yield two other sets of lattices. These three sets of lattices are then used for system combination as described in Section 2.4.

The performance of the lattice combination experiments is reported in Table 2.2. In these experiments, we use a cutting period of 6 for the periodic risk-based lattice cutting. We test these procedures on the Switchboard1 portion of the 2000 Hub5 eval-

uation set (SWB1), Switchboard2 portion of the 1998 Hub5 evaluation set (SWB2) and the Switchboard-Cellular development set released in 2000 (SWB2C). The Table 2.2 is organized as follows. We first report the performance of the MAP hypothesis from each system. We next give results by a simple system combination technique (Lattice-Intersect) that intersects the three lattices and obtains the MAP hypothesis from the resulting lattice [73]. We also report results by the ROVER system combination scheme [28] on the MAP hypotheses from the three systems. We then finally present the results by the two multi-system SMBR decoding schemes (Union-SMBR and Intersect-SMBR) presented in Section 2.4. The e-ROVER procedure is used to compute the MBR hypothesis in both these schemes.

We observe that the multiple system SMBR decoding via either the union or intersection scheme is better than 1) intersecting lattices and obtaining the MAP hypothesis or 2) performing a ROVER on the MAP hypotheses from the three systems. Furthermore, we note that adding posteriors of the paths in the sub-lattices (Union-SMBR) turns out to be better than multiplying them (Intersect-SMBR).

Decoding Strategy	WER(%)		
	SWB1	SWB2	SWB2C
MAP			
Sys 1 (MMIE)	24.5	39.2	39.6
Sys 2 (DLLT)	24.0	38.7	38.8
Sys 3 (DSAT)	24.5	39.3	39.5
Lattice-Intersect	24.0	38.4	38.7
1-best ROVER	23.8	38.1	38.2
SMBR Decoding			
Intersect - SMBR	23.5	37.8	38.0
Union- SMBR	23.3	37.8	37.8

Table 2.2: Performance of Multiple-System SMBR Decoding.

2.6 Discussion

The Segmental Minimum Bayes-Risk Decoding framework allows us to decompose an utterance level Minimum Bayes-Risk Recognizer into a sequence of smaller sub-utterance recognizers. Therefore, a large search problem is decomposed into a sequence of simpler, independent search problems. Though the utterance level MBR decoder is implemented as a sequence of MBR decoders on hypothesis and evidence space segments, the acoustic data is not segmented at all. The marginal probability of a word string within a segment set is computed based on acoustic and language model scores that span the entire utterance; these might have a much greater span than any string in the segment set. In addition, there is no assumption of linguistic independence between word strings belonging to adjacent segments. This is not the case when the entire conversation level decoder is simplified to decoders at the utterance level; by contrast, in that case we do segment acoustic data and assume acoustic and linguistic independence between utterances.

We have described a risk-driven procedure for segmenting word lattices into sub-lattices for SMBR decoding. The strategy attempts to find segments that preserve the total risk of all word strings in the lattice. The procedure identifies node sets that can be used to segment the lattice. However, we have shown that the selection of cut sets must be made considering both SMBR search errors and errors due to poor approximation of the loss function. We have introduced periodic lattice cutting as a cut set selection procedure that finds a balance between these two types of modeling errors. Lattice cutting, in conjunction with SMBR decoding gives consistent improvements as the final stage of an LVCSR evaluation system. In addition, the risk based cutting procedure has been shown to form the basis for novel discriminative training and classification procedures [25, 100].

We note that in the lattice segmentation experiments reported in our original publication [37], some of the hypotheses in the original lattices were inadvertently discarded during segmentation, and this affected MBR performance adversely. In this chapter we have presented the corrected results and also described experiments confirming that the segmentation process does not discard any paths from the original

lattice. The pinched lattices constructed under the PLC procedure always yield a lower oracle error rate than the original lattices; in contrast, other lattice-processing procedures such as confusion network generation [63] have been shown to degrade oracle error rates of lattices [97].

We have presented the N-best ROVER and e-ROVER procedures for rescoring N-best lists. Both these procedures are extensions of ROVER, which is a classifier combination scheme to combine single-best outputs from multiple speech recognizers. N-best ROVER and e-ROVER can be seen as instances of segmental MBR voting. In both procedures, the induced loss function provides a good approximation to the Levenshtein loss function. This accounts for the success of these schemes in reducing the word error rate.

Finally, we have described the application of risk-based lattice segmentation to multiple system SMBR decoding on lattices produced by several ASR systems. The risk-based lattice cutting is more suited to system combination compared to confidence based lattice segmentation strategies [32, 36] since it does not rely on word boundary times which can easily vary across multiple systems. We have presented two schemes to merge posteriors of word strings in sub-lattices and then performing SMBR decoding. The system combination scheme performs better than the output produced by a MAP decoder on each of the individual lattices or on a intersection of the lattices.

Part III

Statistical Machine Translation

Chapter 3

A Weighted Finite State Transducer Translation Template Model

Statistical Machine Translation (SMT) is the application of statistical techniques to the task of translating texts from one natural language (e.g. French) to another (e.g. English). An important sub-problem of SMT is word alignment of bilingual texts (or bitexts). Bitext Word Alignment involves identification of word and phrase correspondences between pairs of translated sentences.

Starting with this chapter, we discuss statistical machine translation and the application of Minimum Bayes-Risk techniques to bitext word alignment and translation. A prerequisite for Minimum Bayes-Risk decoding is a statistical translation model that can assign likelihoods to pairs of translations and generate word alignment and translation hypotheses. We present a Weighted Finite State Transducer *Translation Template Model* (TTM) for statistical machine translation. This is a source-channel model of translation that is inspired by the Alignment Template translation model [79]. This model attempts to overcome the deficiences of word-to-word translation models by considering phrases rather than words as units of translation. The approach we describe allows us to implement each constituent distribution of the model as a weighted finite state transducer or acceptor. We show that bitext word

alignment and translation under the model can be performed with standard finite state operations involving these transducers. We now place this work in the context of recent developments in statistical machine translation. We also show how this work differs from related approaches in translation modeling, notably the Alignment Template translation model developed by Och, Tillmann and Ney [83, 79].

3.1 Previous developments leading to TTM

Statistical machine translation originated with the pioneering work at IBM [9, 10] in modeling the translation process via a string-to-string noisy channel. This noisy channel model consists of a language model (source model) and a translation model (channel model). Each IBM translation model specifies a sequence of operations to transform a word sequence in one language to another; these operations include word duplications, word movements, and word translations. The IBM group developed a series of five translation models of increasing complexity, and developed procedures to estimate the parameters of these models directly from parallel bilingual texts.

There has subsequently been considerable effort devoted to improving the IBM models themselves [102, 82] and developing improved translation search algorithms based on those models [103, 43, 94, 84, 30]. There also have been advancements in the understanding of the nature of these models, notably, due to the work by Knight and Al-Onaizan [43] that describes how Weighted Finite State Transducers (WFSTs) can be used to perform translation using the IBM models, albeit in slightly modified form. In addition to the efficiencies in computation that can be obtained using WFSTs, that formulation provides an accessible, intuitive description of IBM models 1 through 3. Motivated by this work, we developed WFST-based bitext word alignment algorithms and used them to generate alignment lattices for Minimum Bayes-Risk decoding (Chapter 6). However these applications were restricted in power by their reliance on the IBM-3 model, which is the most complex of the IBM models that can easily be treated as a WFST.

The IBM-3 model appears particularly weak in comparison to the Alignment Template Model developed by Och, Tillmann, and Ney [83], which attempts to overcome

the limitations of IBM-style word-to-word translation models by considering whole phrases rather than words as the basis for translation. Under this model, a phrase in the target language (e.g. French) would be translated to a phrase in the source language (e.g. English), and the basic unit of this model is an *alignment template* that specifies the allowable word alignments within a pair of source and target phrases. In our first attempt to implement this model directly using WFSTs, we developed a formulation within which each of the component models can be implemented as a weighted finite state transducer [47]. In doing so, we also generalized the model to support bitext word alignment. That implementation provided a working translation system that we used as a basis for the Chinese-to-English translation system [13] submitted in the NIST 2003 MT evaluations [78]. However, it was flawed in how it incorporated the source language model and in its treatment of phrase insertions and deletions in bitext word alignment. These shortcomings motivated the current model.

We here describe a source-channel model of translation inspired by the WFST implementation of the Alignment Template Model. We have two objectives in doing so. First, by following a careful source-channel formulation we can be certain that all components of the model come together to form a distribution that describes translations. Secondly, each component of the overall model is constructed so that translation and bitext word alignment can be carried out using standard WFST operations.

Our current model departs from the original Alignment Template Model [83, 79] in several ways. In addition to the new formulation of the overall statistical model, the components of the model do not make use of the word alignments within the alignment templates; we model only the translation of phrases. This does not prevent using the model in bitext word alignment, however, and we describe how this can be done. We furthermore allow insertions of target language phrases in the generative translation process; this removes the restriction that the source and target language sentences contain the same number of phrases. To avoid confusion with previous work, we call this model the *Translation Template Model* (TTM) [50, 49], leaving out the reference to word alignment within phrases. In this chapter we present the Translation Template Model and show how it can be implemented component-wise

using WFSTs.

We acknowledge other recent and related work in developing phrase-based models for statistical machine translation. In particular, there are new techniques available for extracting phrase pairs from bitext, either using underlying word alignments [95, 44] or not [110, 65]. Bangalore and Ricardi [4] have also explored the use of WFSTs for machine translation. They implement a two-step translation process in which the foreign sentence is first mapped to an English word sequence, but in foreign word order; that string is then reordered into English word order. Both processing steps are implemented by WFSTs and the overall approach has been applied in a call-routing task. While related in its use of WFSTs for translation, our work (and that of Knight and Al-Onaizan [43]) differs in spirit from Bangalore et al. in that we are mainly focused on the formulation of a source-channel model of translation and its implementation via WFSTs. We finally note that WFSTs have been used for performing translation in small vocabulary limited domain applications [101, 2].

The rest of this chapter is organized as follows. In Section 3.2 we present the derivation of the overall translation model that identifies the conditional independence assumptions among the component variables. The TTM relies on an inventory of target language phrases and their source language translations. In Section 3.3 we describe how such an inventory of phrase-pairs can extracted from aligned bitext. The TTM has six component models, and we describe each along with its WFST implementation in Section 3.4. We finally discuss the TTM in Section 3.5.

3.2 The Translation Template Model

We present here a derivation of the Translation Template model (TTM), and give an implementation of the model using Weighted Finite State Transducers.

The TTM is a source-channel model of translation[1] (Figure 3.1) [9] It defines a joint probability distribution over all possible phrase segmentations and alignments of

[1]In our presentation, the terms *source* and *target* refer to the noisy channel input and output respectively. However, we note that in traditional machine translation literature, the source and the target refer to the output and input of the channel respectively, since the direction of the translation task is the reverse direction of the noisy channel.

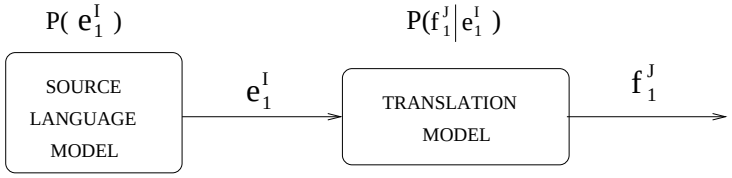

Figure 3.1: A Source Channel Model of Machine Translation.

target language sentences and their translations in the source language. The steps in the translation process are presented with the aid of an example in Figure 3.2, and the conditional dependencies underlying this process are defined in Equation 3.1. Each of the conditional distributions that make up the model is realized independently. In Section 3.4 we define each in turn and present its implementation as a weighted finite state acceptor or transducer.

$$
\begin{aligned}
&P(f_1^J, v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K, e_1^I) = \\
&\quad P(e_1^I)\cdot && \textit{Source Language Model} \\
&\quad P(u_1^K, K|e_1^I)\cdot && \textit{Source Phrase Segmentation} \\
&\quad P(a_1^K|u_1^K, K, e_1^I)\cdot && \textit{Phrase Order} \\
&\quad P(c_0^K|a_1^K, u_1^K, K, e_1^I)\cdot && \textit{Target Phrase Insertion} \\
&\quad P(v_1^R, d_0^K|c_0^K, a_1^K, u_1^K, K, e_1^I)\cdot && \textit{Phrase Transduction} \\
&\quad P(f_1^J|v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K, e_1^I) && \textit{Target Phrase Segmentation}
\end{aligned}
\tag{3.1}
$$

We start with an example (Figure 3.2) showing the generative process through which the TTM transforms a source language sentence into its translation in the target language[2]. In this example, the Source Language Model generates the Source Language Sentence *grain exports are projected to fall by 25 %.* This sentence is segmented into a source phrase sequence: *grain exports are_projected_to fall by_25_%* under the Source Phrase Segmentation Model. This source phrase sequence is reordered into the target language phrase order: *exports grain are_projected_to fall by_25_%* under the Phrase Order Model. The reordered source phrase sequence is then transformed into a

[2]We note that English-French examples in this thesis are taken from the Canadian Hansards corpus as provided to us by the LDC.

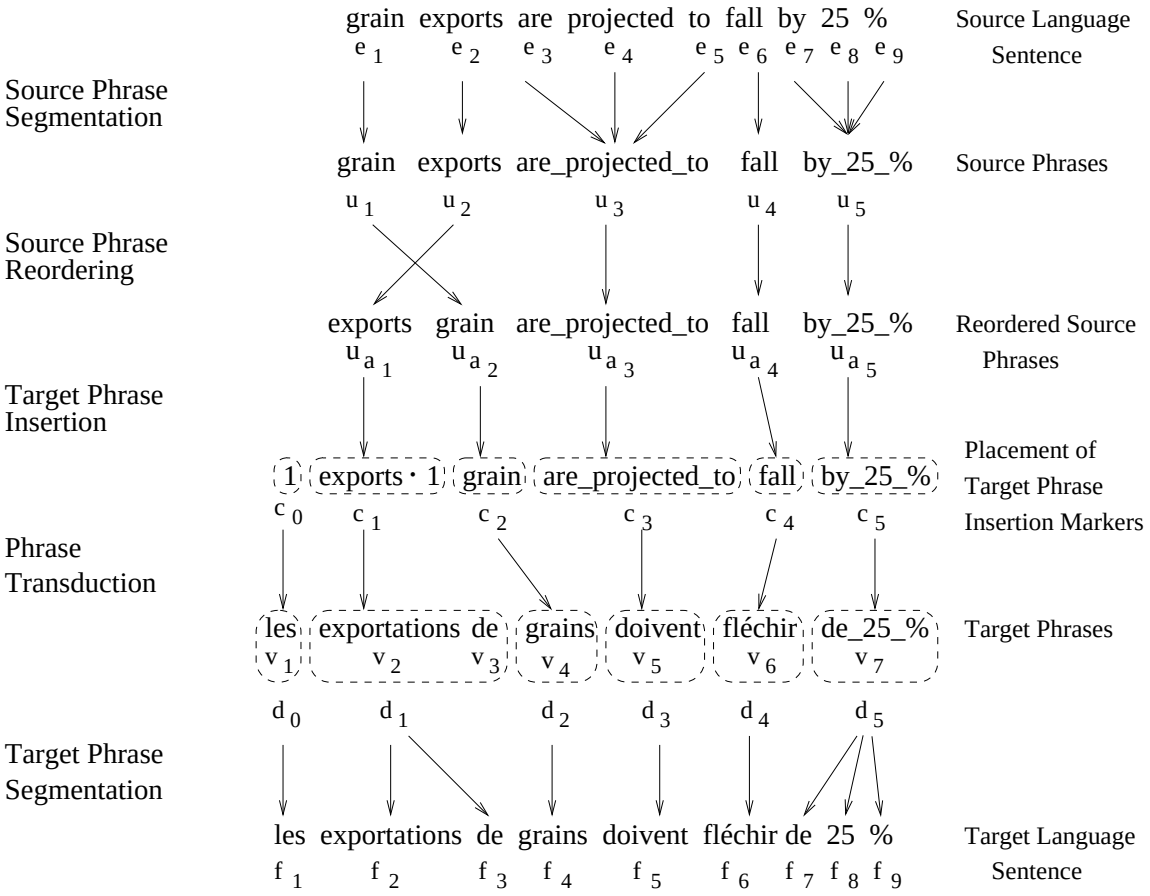

Figure 3.2: An example showing the generative translation process through which the TTM transforms a source language sentence into its translation in the target language. We show the inputs and outputs for each TTM constituent model as well as the TTM variables from Equation 3.1. In this example, $I = 9, K = 5, R = 7, J = 9$.

sequence: *1 exports*·1 *grain are_projected_to fall by_25_%*, where the integers indicate the length of target phrases to be spontaneously inserted; this process is governed by the Target Phrase Insertion Model. The above sequence is next converted into a target language phrase sequence *les exportations de grains doivent fléchir de_25_%* under the Phrase Transduction Model. We note that the words *les* and *de* are spontaneously inserted. Finally the target language phrase sequence is transformed into the target language sentence: *les exportations de grains doivent fléchir de 25 %* under the Target Phrase Segmentation Model. It should be understood that all of the above steps are stochastic, and the example shown is only one possible realization.

We now define some notation. e_1^I refers to a sequence of I elements, and e_i^j refers to the subsequence that begins with the i^{th} element and ends with the j^{th}, *e.g.* if $e_1^I =$ *A B C D*, then $e_2^3 =$ *B C*, where $I = 4$. We next distinguish words and phrases. We assume that u is a phrase in the source language sentence that consists of a variable number of words $e_1, e_2, ..., e_M$. Similarly, v is a phrase in the target language sentence of words $f_1, f_2, ..., f_N$. Throughout the model, if an I word sentence e_1^I is segmented into K phrases u_1^K, we say $u_1^K = e_1^I$ to indicate that the words in the phrase sequence are those of the original sentence.

3.3 The Phrase-Pair Inventory

The Translation Template Model relies on an inventory of target language phrases and their source language translations. These translations need not be unique, in that multiple translations of phrases in either language are allowed. The manner by which the inventory is created does not affect our formulation.

For the experiments that will be presented in this thesis, we utilize the *phrase-extract* algorithm [79] to extract a library of phrase-pairs from bitext word alignments. We first obtain word alignments of bitext using IBM-4 translation models (see Appendix A) [10] trained in both translation directions (IBM-4 F and IBM-4 E), and then form the union of these alignments (IBM-4 $E \cup F$). We will refer to these initial models as the *underlying models*. We next use the algorithm to identify pairs of phrases (u, v) in the source and target language that align well according

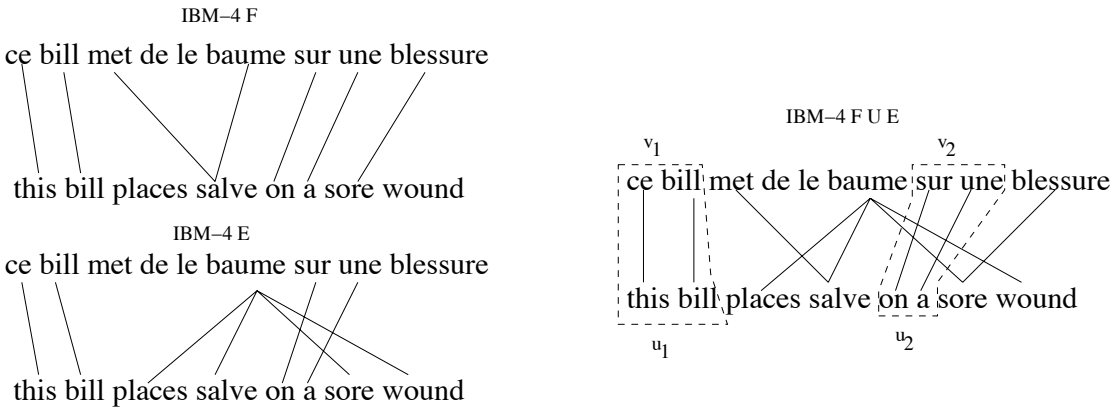

Figure 3.3: Phrase-Pair Collection Process from Bidirectional word alignments of an English-French sentence pair. In this example we extract only those phrase-pairs which have at most 5 words in the French phrase.

to a set of heuristics [79]. These heuristics ensure that the source and target words within a phrase pair are aligned only to each other and not to any words outside the phrase-pair [79, 44]. We now outline the phrase-pair extraction procedure following the presentation of Och [79].

For each source-target sentence pair $(E = e_1^I, F = f_1^J)$ with word alignment B

1. Enumerate all subsequences of the target language sentence f_1^J up to a maximum length. A subsequence $v = f_x^y$ starts at the x^{th} position and ends at the y^{th} position.

2. For each target language subsequence $v = f_x^y$ identified in step 1,

   (a) Determine the subset of words in the source language sentence that are aligned under B to any target language word within v. Locate the leftmost and rightmost source language words in this subset; suppose these are e_p and e_q respectively.

   (b) Determine the subset of words in the target language sentence that are aligned to any source language word within $u = e_p^q$. Locate the leftmost and rightmost target language words in this subset; suppose these are $f_{x'}$ and $f_{y'}$ respectively.

(c) If $x \leq x'$ and $y \geq y'$, add (e_p^q, f_x^y) to the phrase-pair inventory. Add the subsequences e_p^q and f_x^y to the source language and target language phrase inventories respectively.

We emphasize that a target language subsequence $v = f_x^y$ is considered a target language phrase if and only if the above procedure is able to find one or more source language subsequences u that are aligned to v. In Figure 3.3 we show the extraction of phrase-pairs from bidirectional word alignments of an English-French sentence pair.

To restrict the memory requirements of the model, we extract only the phrase-pairs which have at most 5 words in the target phrase. For each pair of target and source phrases, we retain the matrix of word alignments that occurs most frequently in the training corpus. We augment this inventory by the most likely translations of each target (source) word from the IBM-4 translation tables [10] so as to get complete coverage of all single word phrases in either language. We note that a monolingual source (target) phrase inventory can be created by listing the unique source (target) phrases from the phrase-pair inventory.

3.4 TTM Component Models

We here introduce the definitions of the component distributions of the Translation Template Model in Equation 3.1. In presenting these, we first define the component probability distribution, and then describe its implementation using a Weighted Finite State Transducer or an Acceptor.

3.4.1 Source Language Model

We specify this model using a standard monolingual trigram word language model

$$P(e_1^I) = \prod_{i=1}^{I} P(e_i|e_{i-1}, e_{i-2}).$$

Any n-gram or other language model that can be easily compiled as a weighted finite state acceptor could be used [1]. We will use G to denote the language model WFSA.

3.4.2 Source Phrase Segmentation Model

We construct a joint distribution over all phrase segmentations $u_1^K = u_1, u_2, ..., u_K$ of the source sentence e_1^I as

$$P(u_1^K, K|e_1^I) \quad = \quad P(u_1^K|K, e_1^I)P(K|I). \tag{3.2}$$

We choose the distribution over the number of phrases $P(K|I)$ to be uniform

$$P(K|I) = \frac{1}{I}; K \in \{1, 2, ..., I\}. \tag{3.3}$$

For a given number of phrases, the segmentation model is a uniform distribution over the set of K-length phrase sequences of e_1^I

$$P(u_1^K|K, e_1^I) \quad = \quad \begin{cases} C & u_1^K = e_1^I \text{ and} \\ & u_i, i \in \{1, 2, ..., K\} \text{ belongs to the source phrase inventory} \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where C is chosen to ensure that $\sum_{u_1^K} P(u_1^K|K, e_1^I) = 1$. In summary, this distribution assigns a uniform likelihood to all phrase segmentations of the source sentence that can be obtained using the phrase inventory.

We now show the probability computations for two different phrase segmentations of the 6-word source language sentence: *what are its terms of reference*. We first show the portion of our source phrase inventory restricted to the phrases in the above sentence (Table 3.1). Suppose we consider two alternate phrase segmentations of this sentence: U_1 = what_are its terms_of_reference (3 phrases) and U_2 = what_are its_terms of reference (4 phrases). The '_' symbol is used to indicate phrases formed by concatenation of consecutive words.

Under the inventory shown in Table 3.1, we note that the source language sentence can have 4 possible phrase segmentations of length 3 phrases, and 6 possible phrase segmentations of length 4 phrases. To ensure that $P(u_1^K|K, e_1^I)$ is correctly normalized, we therefore set C (in Equation 3.4) to $\frac{1}{4}$ for $K = 3$, and to $\frac{1}{6}$ for $K = 4$.

Phrase Length (in English words)			
1	2	3	4
what are its terms of reference	what_are its_terms terms_of of_reference	its_terms_of terms_of_reference	its_terms_of_reference

Table 3.1: A portion of the source phrase inventory restricted to the phrases in the English sentence : *what are its terms of reference*.

The probabilities assigned to segmentations U_1 and U_2 are given by :

$$P(3, U_1|e_1^I) = P(3|e_1^I)P(U_1|3, e_1^I) = \frac{1}{6} \times \frac{1}{4} = 0.042$$
$$P(4, U_2|e_1^I) = P(4|e_1^I)P(U_2|4, e_1^I) = \frac{1}{6} \times \frac{1}{6} = 0.028.$$

Implementation via WFSTs The WFST implementation of the Source Phrase Segmentation model involves an unweighted segmentation transducer W that maps source word sequences to source phrase sequences. The transducer performs the mapping of source word strings to phrases for every source phrase in our inventory. A portion of the segmentation transducer W is presented in Figure 3.4.

We now describe the procedure to construct a WFST for the distribution $P(u_1^K|K, e_1^I)$. In particular we must ensure that $\sum_{u_1^K} P(u_1^K|K, e_1^I) = 1$ for each source sentence e_1^I and $K \in \{1, 2, ..., I\}$.

1. Build a finite state word acceptor T for the source sentence e_1^I (Figure 3.5). Generate a transducer of segmentations of e_1^I by composing T with W, i.e. $\mathcal{U} = T \circ W$.

2. Partition the transducer $\mathcal{U}$ into I disjoint transducers $\mathcal{U}_K$ so that $\cup_{K=1}^{I} \mathcal{U}_K = \mathcal{U}$; each $\mathcal{U}_K$ consists of those segmentations of the source sentence with exactly K phrases. To construct $\mathcal{U}_K$, create an unweighted acceptor P_K that accepts any phrase sequence of length K; for efficiency, the phrase vocabulary is restricted to the phrases in $\mathcal{U}$. Obtain $\mathcal{U}_K$ by the finite state composition: $\mathcal{U}_K = \mathcal{U} \circ P_K$.

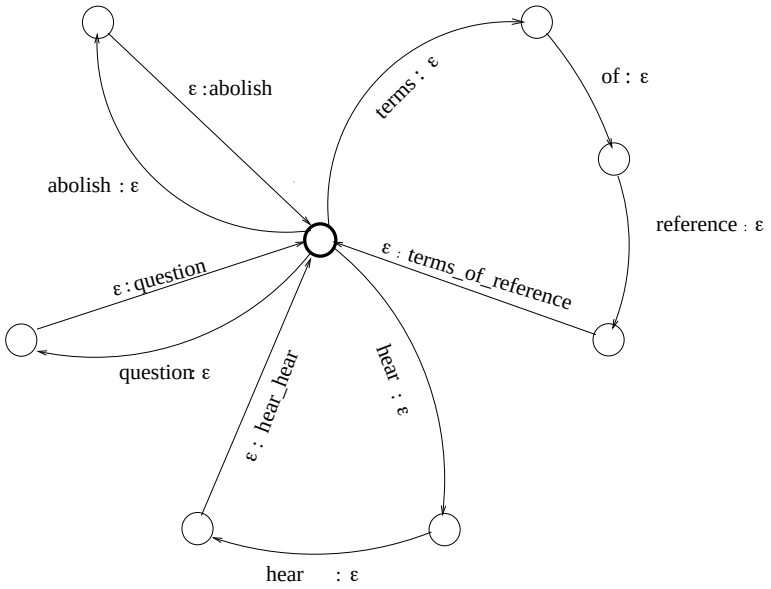

Figure 3.4: A portion of the Source Phrase Segmentation Transducer W that maps word sequences to phrases. Suppose an example input for this transducer is the source language sentence: *What are its terms of reference*, then a possible output of WFST would be the source language phrase sequence: *what_are its terms_of_reference*.

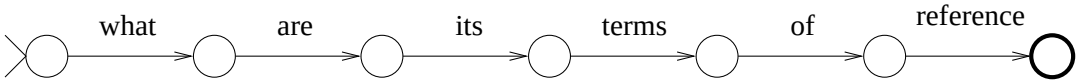

Figure 3.5: An unweighted finite state acceptor for the source language sentence: *What are its terms of reference*.

3. For $K = 1, 2, ..., J$

   Obtain the total number of distinct paths C_K in $\mathcal{U}_K$. This can be computed efficiently using lattice forward probabilities [104]. Set the probability of each path to $\frac{1}{C_K}\frac{1}{I}$ to obtain a new transducer $\mathcal{U'}_K$.

4. Construct a new segmentation lattice $\mathcal{U'} = \cup_{K=1}^{I} \mathcal{U'}_K$.

The segmentation lattice $\mathcal{U'}$ obtained through the above procedure will be normalized so that probabilities of all segmentations of a given length would sum up to one, i.e. $\sum_{u_1^K} P(u_1^K | K, e_1^I) = 1; K \in \{1, 2, ..., I\}$.

We emphasize that these forms of the segmentation distribution are exceedingly

simple, i.e. uniform probabilities, and were chosen for ease of presentation. More complex phrase segmentation models can easily be implemented in this framework.

3.4.3 Phrase Order Model

We here define a model for the reordering of the source phrase sequences that make up the source sentence. The phrase alignment sequence a_1^K specifies a reordering of source phrases into *target language phrase order*; note that the words within the phrases remain in the original order. In this way the phrase sequence u_1^K is reordered into $u_{a_1}, u_{a_2}, ..., u_{a_K}$ under the model $P(a_1^K|u_1^K, K, e_1^I)$. We now discuss several phrase order models.

Markov Phrase Order Model The phrase alignment sequence is modeled as a first order Markov process

$$\begin{aligned} P(a_1^K|u_1^K, K, e_1^I) &= P(a_1^K|u_1^K) \\ &= P(a_1)\prod_{k=2}^{K} P(a_k|a_{k-1}, u_1^K). \end{aligned} \tag{3.5}$$

with $a_k \in \{1, 2, ..., K\}$. The alignment sequence distribution is constructed to assign lower likelihood to phrase re-orderings that diverge from the original word order. Suppose $u_{a_k} = e_l^{l'}$ and $u_{a_{k-1}} = e_m^{m'}$, we set the Markov chain probabilities as follows [83]

$$\begin{aligned} P(a_k|a_{k-1}, u_1^K) &\propto p_0^{|l-m'-1|} \\ P(a_1 = k) &= \frac{1}{K}; k \in \{1, 2, ..., K\}. \end{aligned} \tag{3.6}$$

In the above equations, p_0 is a tuning factor and we normalize the probabilities $P(a_k|a_{k-1})$ so that $\sum_{j=1, j\neq a_{k-1}}^{K} P(a_k = j|a_{k-1}) = 1$.

The finite state implementation of the phrase order model involves two acceptors. We first build a unweighted permutation acceptor Π_U that contains all reorderings of the source language phrase sequence u_1^K [43]. We note that a path through Π_U corresponds to an alignment sequence a_1^K. Figure 3.6 shows the acceptor Π_U for the source phrase sequence *we have run_away_inflation*.

A source phrase sequence U of length K words requires a permutation acceptor Π_U of 2^K states. For long phrase sequences we compute a score $\max_j P(a_k = i|a_{k-1} = j)$ for each arc and then prune the arcs by this score, i.e. phrase alignments containing $a_k = i$ are included only if this score is above a threshold. Pruning can therefore be applied while Π_U is constructed.

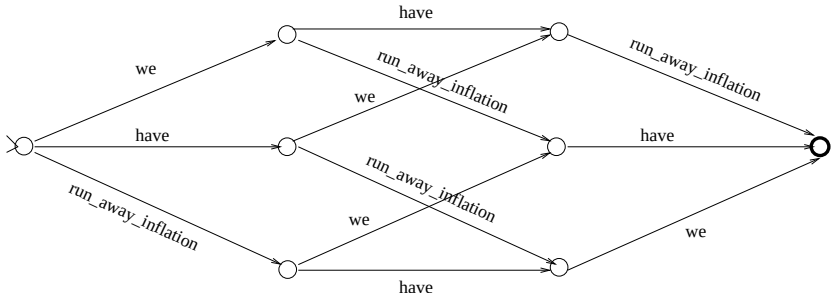

Figure 3.6: The permutation acceptor Π_U for the source-language phrase sequence *we have run_away_inflation*. For this phrase sequence, an example of a reordering allowed by this acceptor is *run_away_inflation we have*, so that the alignment sequence is given by: $a_1 = 3, a_2 = 1, a_3 = 2$.

The second acceptor H in the implementation of the Phrase Order Model assigns alignment probabilities (Equation 3.6) to a given reordering a_1^K of the source phrase sequence u_1^K (Figure 3.7). In this example, the phrases in the source phrase sequence are specified as follows: $v_1 = f_1$ (*we*), $v_2 = f_2$ (*have*) and $v_3 = f_3^5$ (*run_away_inflation*). We now show the computation of some of the alignment probabilities (Equation 3.6) in this example ($p_0 = 0.9$)

$$\begin{aligned} P(a_3 = 1|a_2 = 3) &\propto \quad p_0^{|1-5-1|} = \quad 0.59 \\ P(a_3 = 2|a_2 = 3) &\propto \quad p_0^{|2-5-1|} = \quad 0.66. \end{aligned}$$

Normalizing these terms gives $P(a_3 = 1|a_2 = 3) = 0.47$ and $P(a_3 = 2|a_2 = 3) = 0.53$.

Practical Phrase Order Models The permutation acceptor described above must be constructed for each segmentation u_1^K of the source sentence e_1^I. As a source sentence typically has several segmentations, it is infeasible to construct a separate permutation acceptor for every segmentation. Moreover, during decoding, this process

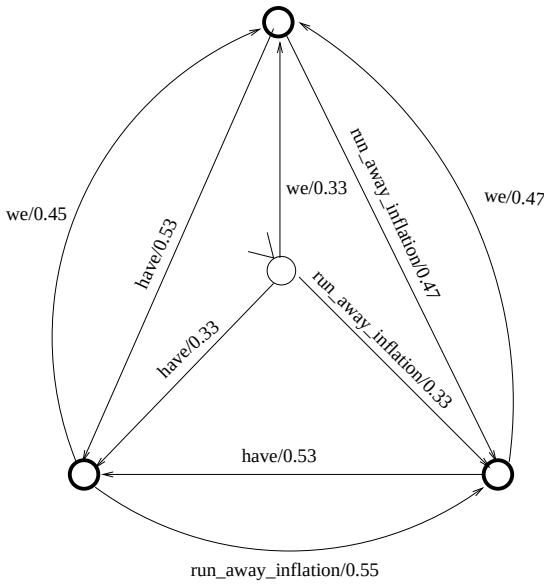

Figure 3.7: Acceptor H that assigns probabilities to reorderings of the source language phrase sequence *we have run_away_inflation* ($p_0 = 0.9$). Given the reordering *run_away_inflation we have* with alignment sequence $a_1 = 3, a_2 = 1, a_3 = 2$, H would assign it a probability: $P(a_1 = 3)P(a_2 = 1|a_1 = 3)P(a_3 = 2|a_2 = 1) = 0.33 \times 0.47 \times 0.53 = 0.08$.

has to be carried out for every source sentence that is allowable by the source language model. As a practical approximation, we therefore consider a degenerate model that does not allow any reordering of the source phrase sequence u_1^K. Therefore the model would be specified as

$$P(a_1^K|u_1^K, K, e_1^I) = \begin{cases} 1 & \{a_1 = 1, a_2 = 2, a_3 = 3, ..., a_K = K\} \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

We will refer to this model as the *Fixed Phrase Order Model*.

3.4.4 Target Phrase Insertion Model

The processes described thus far allow a mapping of a source language sentence into a reordered sequence of source language phrases, whose order is the phrase order of the target language. The constraint that the target language phrase sequence must have the same number of phrases as the source language phrase sequence is overly

restrictive. Our goal is to construct a model to allow insertion of target language phrases anywhere in the reordered source language phrase sequence. This process will be governed by a probability distribution over insertion of target language phrases such that the likelihood of inserting a phrase is inversely proportional to the number of words in the phrase. Therefore there will be a greater penalty for the insertion of longer phrases.

This model transforms the reordered source language phrase sequence $u_{a_1}, u_{a_2}, ..., u_{a_k}$ into a new sequence called c_0^K. The process replaces each source language phrase by a structure that retains the phrase itself and additionally specifies how many target language phrases should be appended to that phrase. Given $u_{a_1}, u_{a_2}, ..., u_{a_k}$, an element in the transformed sequence has the following form

$$c_k = u_{a_k} \cdot p_k \; ; \; p_k \in \{1, 2, ..., M\}^*$$

The term p_k specifies the number and length of the target language phrases that can be spontaneously generated to follow the translation of u_{a_k}. The term has the following form: $p_k = p_k[1] \cdot p_k[2] \cdot ...$ and $p_k[i] \in \{1, 2, ..., M\}$. For example, if $u_{a_k} = terms\_of\_reference$, c_k might equal $terms\_of\_reference \cdot 1 \cdot 3 \cdot 4$, which specifies that the translations of *terms_of_reference* must be followed by three target language phrases of length one word, three words, and four words respectively. We note that these target language phrases must be drawn from the phrase-pair inventory, and therefore are of known maximum word length M. The probability of the element c_k is specified as

$$P(c_k|u_{a_k}) = \begin{cases} \alpha_0 & c_k = u_{a_k} \cdot \epsilon \\ \alpha^{\sum_i p_k[i]} & c_k = u_{a_k} \cdot p_k \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

We will refer to α as the *Phrase Exclusion Probability* (PEP). We note that $c_0, c_1, ..., c_k$ contains one additional term relative to the original sequence $u_{a_1}, u_{a_2}, ..., u_{a_k}$. This term c_0, has the form $c_0 = \epsilon \cdot p_0$, and its probability is given by

$$P(c_0) = \begin{cases} \alpha_0 & c_0 = \epsilon \\ \alpha^{\sum_i p_0[i]} & c_k = p_0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

The total probability of the sequence c_0^K is obtained as

$$P(c_0^K|u_{a_1}, u_{a_2}, ..., u_{a_k}) = P(c_0)\prod_{k=1}^{K} P(c_k|u_{a_k}). \tag{3.10}$$

In the above equations, the value of α_0 is set to ensure that the probability distribution (given in Equation 3.8) is normalized:

$$\begin{aligned}
\sum_{c_k} P(c_k|u_{a_k}) &= P(c_k = u_{a_k} \cdot \epsilon) + \sum_{p_k \neq \epsilon} P(c_k = u_{a_k} \cdot p_k) \\
&= \alpha_0 + \sum_{l=1}^{\infty} \sum_{p_k:|p_k|=l} P(c_k = u_{a_k} \cdot p_k) \\
&= \alpha_0 + \sum_{l=1}^{\infty} \sum_{p_k[1]p_k[2]...p_k[l]} \alpha^{\sum_{i=1}^{l} p_k[i]} \\
&= \alpha_0 + \sum_{l=1}^{\infty} \prod_{i=1}^{l} \sum_{p_k[i] \in \{1,2,...,M\}} \alpha^{p_k[i]} \\
&= \alpha_0 + \sum_{l=1}^{\infty} \prod_{i=1}^{l} \sum_{j=1}^{M} \alpha^j \\
&= \alpha_0 + \sum_{l=1}^{\infty} (\sum_{j=1}^{M} \alpha^j)^l.
\end{aligned}$$

We can set α so that $\sum_{j=1}^{M} \alpha^j < 1$. This imposes a permissible range on α values: $0 \leq \alpha < \alpha_{\max}$, so that $(\sum_{j=1}^{M} \alpha^j)^l$ forms a geometric series in l with sum of its terms given by

$$S = \frac{(\sum_{j=1}^{M} \alpha^j)}{1 - (\sum_{j=1}^{M} \alpha^j)}.$$

Therefore $\sum_{c_k} P(c_k) = \alpha_0 + S$, so that α_0 is fixed by α as $\alpha_0 = 1 - S$.

The WFST Implementation of the Target Phrase Insertion Model involves a transducer Φ shown in Figure 3.8. When a source phrase sequence is composed with Φ, it spontaneously inserts target phrases to generate an output sequence c_0^K according to Equation 3.10.

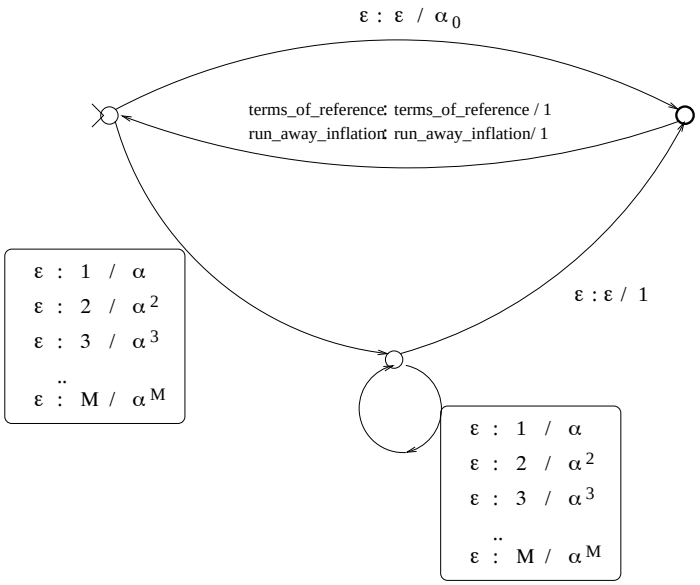

Figure 3.8: A portion of the Weighted Finite State Transducer Φ used to implement the Target Phrase Insertion Model. Suppose an example input for this transducer is the reordered source language phrase sequence *exports grain are projected_to_fall*, then a possible output of the WFST is the sequence 1 *exports* · 1 *grain are projected_to_fall*, which means that two target phrases are spontaneously inserted in the translation of source phrase sequence. The first target phrase is of length one word and inserted at the start of the sentence, and the second target phrase, also of length one, follows the translation of the source phrase *exports*.

3.4.5 Phrase Transduction Model

We have described the segmentation and reordering processes that transform a source language sentence into source language phrases in target language phrase order. The Target Phrase Insertion Model decides the number and length of target phrases that are to be spontaneously inserted within this reordered source phrase sequence. The next step is to map this sequence into a sequence of target phrases.

We assume that the target phrases are conditionally independent of each other and depend only on the source language phrase which generated each of them. Each term c_k is mapped to a sequence of target phrases d_k which are concatenated to obtain

the final target phrase sequence $v_1^R = d_0^K$.

$$\begin{aligned} P(v_1^R, d_0^K | c_0^K, a_1^K, u_1^K, K, e_1^I) &= P(d_0^K | c_0^K) 1\{d_0^K = v_1^R\} \\ P(d_0^K | c_0^K) &= \prod_{k=0}^{K} P(d_k | c_k) \\ &= \prod_{l=1}^{|p_0|} P(d_{0l} | c_{0l}) \prod_{k=1}^{K} \prod_{l=1}^{1+|p_k|} P(d_{kl} | c_{kl}), \end{aligned} \tag{3.11}$$

where $1\{d_0^K = v_1^R\}$ ensures that the target phrase sequence v_1^R agrees with the sequence d_0^K produced by the model. We note that this is the main component model of the TTM. We estimate the phrase translation probabilities by the relative frequency of phrase translations found in bitext alignments. We will implement this model using a transducer Y that maps any reordering of the source language phrase sequence into a target language phrase sequence v_1^R as in Equation 3.11. For every phrase u, this transducer allows only the target phrases v which are present in our library of phrase-pairs. In addition, for each $m \in \{1, 2, ..., M\}$, the transducer allows a mapping from the target phrase symbol m to all the m-length target phrases from our target phrase inventory V_T^m with probability given by

$$P(v|m) = \frac{1}{|V_T^m|}; v \in V_T^m. \tag{3.12}$$

A small portion of the phrase-pair inventory used to build the tranducer Y is shown in Table 3.2.

3.4.6 Target Phrase Segmentation Model

The operations described so far allow a mapping of a source language sentence into a sequence of target language phrases. We now specify a model to enforce the constraint that words in the target sentence f_1^J agree with those in the target phrase sequence v_1^R.

$$P(f_1^J | v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K, e_1^I) = 1\{f_1^J = v_1^R\}.$$

The WFST implementation of this model involves an unweighted segmentation transducer that enforces the above requirement, and maps target phrase sequences to target sentences. We build a weighted finite state transducer Ω for each target language

Source Phrase	Target Phrase	Phrase Transduction Probability
run_away_inflation	inflation_galopante	0.50
run_away_inflation	une_inflation_galopante	0.50
hear_hear	bravo	0.80
hear_hear	bravo_bravo	0.15
hear_hear	ordre	0.05
terms_of_reference	mandat	0.80
terms_of_reference	de_son_mandat	0.20

Table 3.2: A portion of the phrase-pair inventory used to build the Phrase Transducer Y. Y is a trivial single state transducer with number of arcs equal to the size of the inventory.

sentence f_1^J to be translated. The transducer segments the sentence into all possible phrase sequences v_1^R permissible given the inventory of phrases. A portion of the segmentation transducer Ω for the French sentence *nous avons une inflation galopante* is presented in Figure 3.9. When Ω is composed with a valid phrase segmentation, *e.g. nous avons une_inflation_galopante*, it generates the target sentence: *nous avons une inflation galopante*.

3.5 Discussion

We have presented the Translation Template Model (TTM) for statistical machine translation. We have developed this model with two intentions in mind. First the model should be formulated in a way that the conditional dependencies underlying the model are clearly stated. Second we intend to formulate the model in a way that allows bitext word alignment and translation under the model to be implemented using Weighted Finite State Transducer (WFST) operations.

The TTM is a source-channel model of the translation process. It defines a joint distribution over phrase segmentations, reorderings, and phrase-pair translations needed to describe how the source language sentence is translated into the target language.

The model relies on an underlying inventory of target language phrases and their

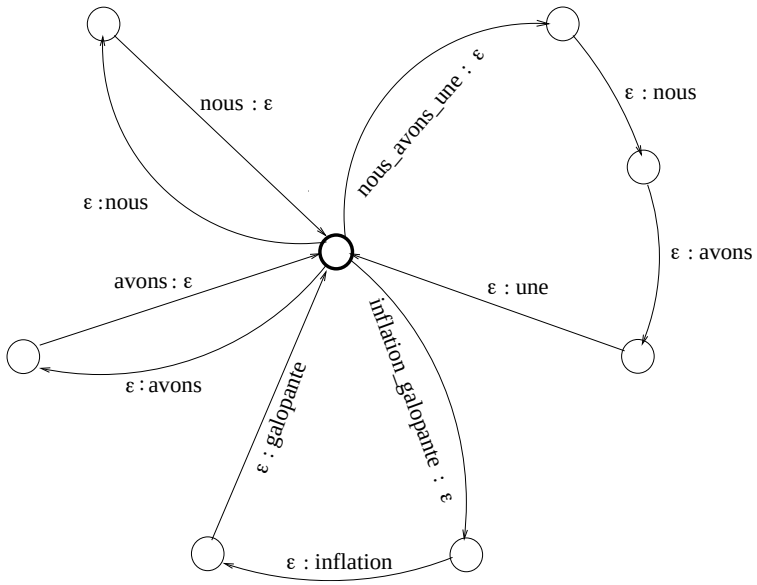

Figure 3.9: A portion of the target phrase segmentation transducer Ω for the target language phrase sequence: *nous avons une_inflation_galopante*. When Ω is composed with this phrase segmentation, it generates the the target language sentence *nous avons une inflation galopante*.

source language translations. The manner by which the inventory is created does not affect the model formulation. In this thesis we have employed IBM-4 word translation models to generate an initial bitext word alignment, and then collected the phrase-pair inventory over this alignment using a set of heuristics [79]. However any word alignment or methodology of collecting phrase pairs could be used.

The TTM consists of six component models each of which can be implemented independently as a weighted finite state acceptor or transducer. The TTM component models are the Source Language Model, Source Phrase Segmentation Model, Phrase Order Model, Target Phrase Insertion Model, Phrase Transduction Model, and the Target Phrase Segmentation Model.

Our derivation of the TTM has allowed us to identify the conditional independence assumptions that underly the WFST implementation. We have shown that this approach leads to modular implementations of the component distributions of the translation model. These components can be refined and improved by chang-

ing the corresponding transducers without requiring changes to the overall search procedure for performing word alignment and translation under the model.

Chapter 4

Bitext Word Alignment under the Translation Template Model

In this chapter we discuss bitext word alignment under the Translation Template Model (TTM) introduced in Chapter 3. Our goal here is to describe how bitext word alignment under the TTM can be performed using Weighted Finite State Transducer (WFST) operations, and evaluate alignment performance of the TTM.

We present extensive experiments analyzing the alignment performance of the TTM system. Our aim is to identify the contribution of each of the model components to different aspects of alignment performance. In doing so, we also analyze some aspects of the performance metrics themselves; these criteria are complex enough that they have behavior of their own. We also study the influence of the bitext used in training the system. The quality of and the amount of available bitext has a strong influence on the quality of the statistical models that result, and we provide an analysis of the influence of both quality and quantity on alignment performance under the TTM.

The experiments in this chapter will be performed on the Hansards French-to-English task [82] and the FBIS Chinese-to-English task [78]. The finite state modeling is performed using the AT&T FSM Toolkit [72].

This chapter is organized as follows. In Section 4.1 we introduce the bitext word alignment problem. We then describe the implementation of word alignment under

the TTM using WFST operations. In Section 4.2 we introduce the French-English and Chinese-English tasks. For each task we provide definitions of the training and test sets, and statistics of the underlying phrase-pair inventories. We present word alignment experiments in Section 4.3. We finally discuss the experiments in Section 4.4.

4.1 WFST Implementation of Word Alignment

4.1.1 Bitext Word Alignment

Given a pair of translations in the source and the target language, the goal of bitext word alignment is to find word-to-word correspondences between these sentences.

Figure 4.1 shows an example of a word alignment for an English-French sentence pair. We now introduce word alignment definitions with the aid of this example. Let $E = e_0^I$ and $F = f_0^J$ denote a pair of translated sentences in the source and the target language. A source word token is defined as an ordered pair $e = (j, w) : w \in V_E, j \in \{0, 1, 2, ..., I\}$, where the index j refers to the position of the word in the source sentence; V_E is the vocabulary of the source language. Similarly, a target word token is written as $f = (i, w) : w \in V_F, i \in \{0, 1, 2, ..., J\}$, where the index i refers to the position of the word in the target sentence; V_F is the vocabulary of the target language.

An *alignment* between E and F is defined to be a link set $B = \{b_1, b_2, ...\}$ whose elements are given by the alignment links b_k. An alignment link $b = (i, j)$ specifies that the source word e_j is connected to the target word f_i under the alignment. Target (Source) words left unaligned are assumed to be connected to the NULL word at position 0 in the Source (Target) sentence. In our example word alignment (Figure 4.1), the French word *chargé* is aligned to the NULL word in the English sentence. We note that this definition of word alignment can describe both automatic alignments and alignments performed by human translators.

We distinguish two types of bitext word alignment [14]. The first type of alignment is referred to as *word-to-word alignment*. The underlying assumption is that each word

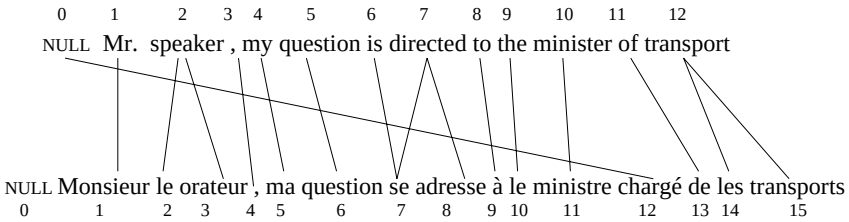

Figure 4.1: An example word alignment for an English-French sentence pair. The link set corresponding to this alignment is given by:
B = (1,1), (2,2), (3,2), (4,3), (5,4), (6,5), (7,6), (7,7), (8,7), (9,8), (10,9), (11,10), (12,0), (13,11), (14,12), (15,12).

in the English sentence can be linked to zero or more words in the French sentence and vice versa. The word alignment shown in Figure 4.1 belongs to this category. The IBM alignment models [10] assume this notion of word alignment but impose an additional constraint that each French (English) word may be linked to at most one English (French) word. A second type of word alignment is phrase-to-phrase alignment. In this style of alignment, a sequence of m English words (i.e. an English phrase) is linked to a sequence of n French words (i.e. a French phrase), where m and n are arbitrary non-negative integers. Furthermore each English phrase may only be unambiguously linked to one French phrase and vice versa. The Translation Template Model assumes this second notion of word alignment.

4.1.2 Use of Phrase-Pairs in Word Alignment

We now describe some issues that arise in the implementation of bitext word alignment using the TTM. We first give an example showing word alignment of a sentence pair under the TTM (Figure 4.2).

Given a source language sentence and its translation in the target language, bitext word alignment under the TTM is performed by considering all segmentations of each sentence and finding the best possible alignment between the phrases under the constraint that all phrases are aligned. However, our inventory of phrase-pairs is not rich enough to cover all possible sentences, and as a result the sentence-pair contains phrase-pairs not in the inventory. Furthermore, this can happen even for

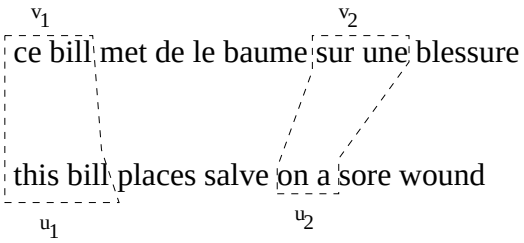

Figure 4.2: An example illustrating the problems in performing bitext word alignment under the TTM. We observe that the inventory of two phrase-pairs is not rich enough to completely cover the words in either the English or the French sentence.

the sentence-pairs in the bitext collection from which the phrase-pair inventory is gathered. We observe this situation in Figure 4.2 where the phrase-pairs extracted from the sentence-pair do not completely cover the words in either the source or the target sentence. When a sentence pair cannot be covered by the inventory, the pair is assigned a probability of zero under the model.

To overcome this limitation, we allow deletion of source phrases during the alignment process. This is done in addition to the insertion of target phrases under the Target Phrase Insertion Model (Equation 3.10). This will make it possible to align sentences containing phrases not found in the phrase pair inventory. The phrase transducer Y is modified by adding extra transitions to allow deletions of source phrases. Therefore each source phrase u can be mapped to an empty string in addition to its regular transductions to target phrases v.

The parameters $P(\epsilon|u)$ for deletions of source phrases u are not estimated; they are tied to the *Phrase Exclusion Probability* (α) introduced in the Target Phrase Insertion Model so that $P(\epsilon|u) = \alpha$ for all source phrases u in our inventory. The parameter α will be tuned to optimize the alignment performance on a development set. We modify the original estimates of phrase transduction probabilities $P(v|u)$ to ensure that the Phrase Transduction Model is correctly normalized while allowing deletions. For each source phrase u in the source phrase inventory, this is done as

follows

$$P'(v|u) = \begin{cases} P(v|u)(1-\alpha) & v \neq \epsilon \\ \alpha & v = \epsilon, \end{cases}$$

This ensures that

$$\sum_{v \in \{V_T, \epsilon\}} P'(v|u) = 1$$

where V_T is the inventory of target language phrases.

Example of Alignment Probability Computations under the TTM With the aid of an example, we now illustrate bitext word alignment under the TTM by by allowing deletions of source language phrases. We present the TTM probability computations for the word alignment shown in Figure 4.3. The top panel of this figure shows the TTM variables (phrase segmentations, reordering, insertions, and deletions) hypothesized in the word alignment of a sentence-pair; the resulting word alignment is shown in the bottom panel.

The probability of this alignment under the TTM can be computed in terms of the probabilities assigned by the TTM component models:

$$\begin{aligned} P(f_1^J, v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K|e_1^I) &= P(K|I)P(u_1^K|K, e_1^I)P(a_1^K|u_1^K) \\ &\quad P(c_0^K|a_1^K, u_1^K)P(v_1^R, d_0^K|c_0^K)P(f_1^J|v_1^R). \end{aligned}$$

The TTM component probabilities for this example are computed as follows:

$$\begin{aligned} P(K|I)P(u_1^K|K, e_1^I) &= \tfrac{1}{8} \times C \\ P(a_1^K|u_1^K) &= 1 \\ P(c_0^K|a_1^K, u_1^K) &= \alpha_0 \times \alpha^{0+1+3+0+1} \\ P(v_1^R, d_0^K|c_0^K) &= P(\text{ce\_bill}|\text{this\_bill}) \times \tfrac{1}{|V_T^1|} \times P(\epsilon|\text{places\_salve}) \times \tfrac{1}{|V_T^3|} \\ &\quad \times P(\text{sur\_une}|\text{on\_a}) \times P(\epsilon|\text{sore\_wound}) \times \tfrac{1}{|V_T^1|} \\ P(f_1^J|v_1^R) &= 1. \end{aligned}$$

In the above equations the value of C is chosen to ensure that the source segmentation model is correctly normalized. We employ the Fixed Phrase Order Model that

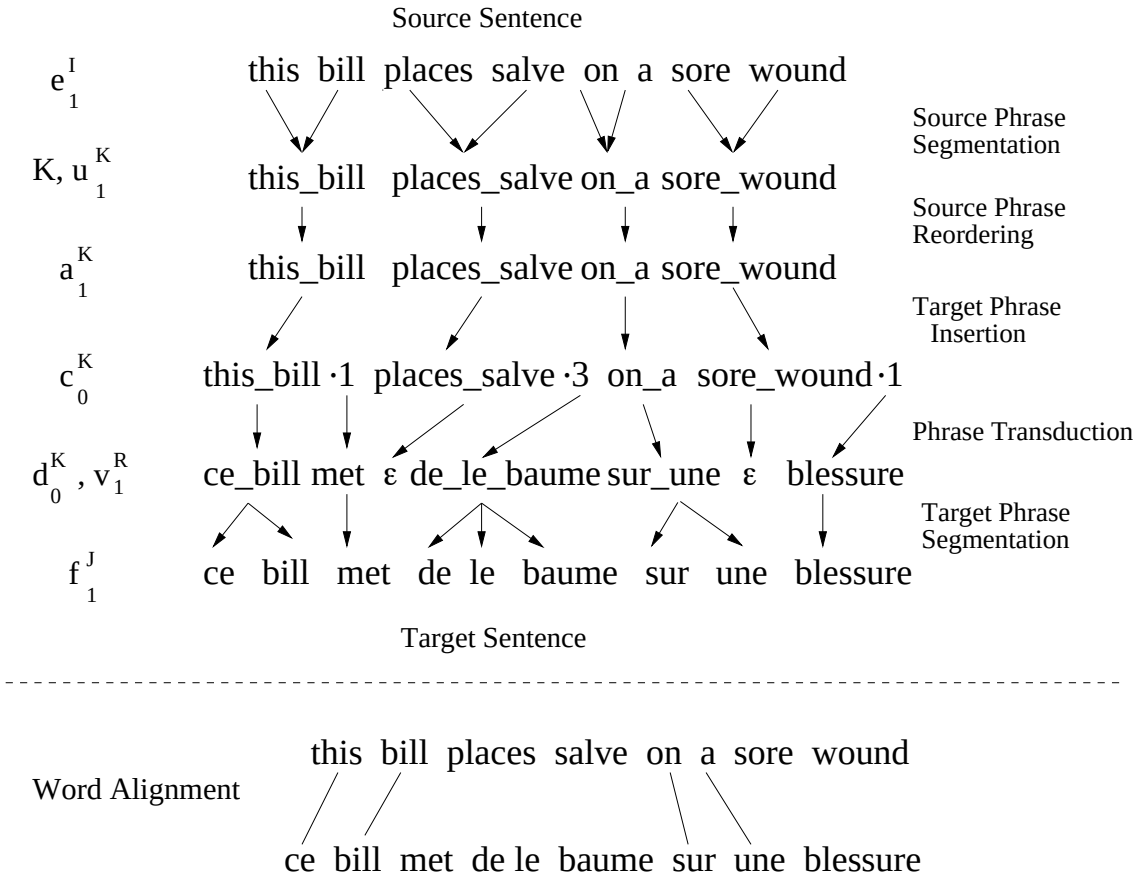

Figure 4.3: Example of bitext word alignment under the TTM. The top panel shows the phrase segmentations, reorderings, insertions, and deletions hypothesized in the word alignment of this sentence-pair. The resulting word alignment is shown in the bottom panel.

disallows any movement of source phrases. α is the Phrase Exclusion Probability and α_0 is determined by α (Section 3.4.4). V_T^3 and V_T^1 denote the subset of 3-length and 1-length phrases from the target language (French) phrase inventory.

4.1.3 WFST Computations

Given a target language sentence f_1^J and its translation e_1^I in the source language, the word-to-word alignment between the sentences can be found using *Maximum A*

Posteriori (MAP) decoding

$$\begin{aligned}\{\hat{v}_1^{\hat{R}}, \hat{d}_0^{\hat{K}}, \hat{c}_0^{\hat{K}}, \hat{a}_1^{\hat{K}}, \hat{u}_1^{\hat{K}}, \hat{K}\} &= \underset{v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K}{\operatorname{argmax}} P(v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K | f_1^J, e_1^I) \\ &= \underset{v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K}{\operatorname{argmax}} \frac{P(f_1^J, v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K | e_1^I)}{P(f_1^J | e_1^I)} \\ &= \underset{v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K}{\operatorname{argmax}} P(f_1^J, v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K | e_1^I),\end{aligned} \tag{4.1}$$

where the last equality holds because (e_1^I, f_1^J) is constant over all alignments of a sentence pair. $\hat{u}_1^{\hat{K}}$ and $\hat{d}_0^{\hat{K}} = \hat{v}_1^{\hat{R}}$ specify the MAP source phrase sequence and target phrase sequence respectively. $\hat{c}_0^{\hat{K}}$ specifies the position and length of the spontaneously generated target phrases within the reordered source phrase sequence. $\hat{a}_1^{\hat{K}}$ describes the MAP phrase-to-phrase alignment between the phrase sequences so that $\hat{c}_i$ is aligned to the target phrase $\hat{d}_i$. The MAP hypotheses are generated at the phrasal level, however using the knowledge that $\hat{c}_i$ is aligned to $\hat{d}_i$, we can obtain the word level alignments within the phrases directly from the phrase pair inventory. In this way we can generate the single MAP alignment. We now describe the WFST computation of the MAP word alignment in three different configurations involving phrase segmentations and phrase sequence reorderings.

All phrase segmentations, No Phrase reordering We first describe how MAP word alignment under the TTM can be obtained when all phrase segmentations of the source sentence are considered and no reorderings of the source phrase sequence are considered. In this case a lattice of possible word alignments between e_1^I and f_1^J can be obtained by the finite state composition

$$\mathcal{B} = T \circ W \circ \Phi \circ Y \circ \Omega \circ S,$$

where T is an acceptor for the source sentence e_1^I, and S is an acceptor for the target sentence f_1^J. An alignment lattice can be generated by pruning $\mathcal{B}$ based on likelihoods or number of states. The MAP alignment $\hat{B}$ (Equation 4.1) is found as the path with the highest probability in $\mathcal{B}$.

One phrase segmentation, No Phrase reordering If only the most probable phrase segmentation of the source sentence is to be considered during alignment, we follow a two-step procedure proposed earlier [47] in place of Equation 4.1. The first step is MAP phrase segmentation of the source sentence, followed by the MAP alignment of the fixed segmentation.

$$\{\tilde{u}_1^{\tilde{K}}, \tilde{K}\} = \underset{u_1^K, K}{\operatorname{argmax}} P(u_1^K, K|e_1^I) \tag{4.2}$$

$$\{\tilde{v}_1^{\tilde{R}}, \tilde{d}_0^{\tilde{K}}, \tilde{c}_0^{\tilde{K}}, \tilde{a}_1^{\tilde{K}}\} = \underset{v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}}{\operatorname{argmax}} P(v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}|\tilde{u}_1^{\tilde{K}}, \tilde{K}, e_1^I, f_1^J) \tag{4.3}$$

$$= \underset{v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}}{\operatorname{argmax}} P(v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}|\tilde{u}_1^{\tilde{K}}, \tilde{K}, f_1^J) \tag{4.4}$$

$$= \underset{v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}}{\operatorname{argmax}} P(f_1^J, v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}|\tilde{u}_1^{\tilde{K}}, \tilde{K}), \tag{4.5}$$

where Equation 4.3 can be simplified to Equation 4.4 because the alignment variables $v_1^R, d_0^{\tilde{K}}, c_0^{\tilde{K}}, a_1^{\tilde{K}}$ are conditionally independent of e_1^I given $\tilde{u}_1^{\tilde{K}}, \tilde{K}$. Also Equation 4.4 can be simplified to Equation 4.5 because f_1^J is constant over all alignments of the sentence pair.

This two-stage procedure is implemented via WFSTs as follows. We first obtain a segmentation lattice of the source sentence: $\mathcal{U} = T \circ W$. The MAP source phrase seqmentation $\tilde{U}$ is obtained as the path with the highest probability in $\mathcal{U}$. Given the MAP segmentation $\tilde{U}$, the alignment lattice can be obtained by the WFST composition:

$$\mathcal{B} = \tilde{U} \circ \Phi \circ Y \circ \Omega \circ S$$

.

One phrase segmentation, N-best Phrase reorderings The above presentation assumes that the source phrase sequence is not reordered while performing alignment. If reorderings of the MAP source phrase segmentation are to be considered when obtaining MAP word alignment, we perform the following procedure. We first obtain the MAP phrase segmentation of the source language sentence as described

above. We next build a permutation acceptor $\Pi_{\tilde{U}}$ that generates reorderings of the source phrase sequence $\tilde{U}$. The N-best reorderings of $\tilde{U}$ are obtained by considering the N most likely paths in the permutation acceptor under the Markov Phrase Order Model (Equation 3.6). Given this set of reorderings of the source phrase sequence, the alignment lattice is found by a WFST composition. These two steps are given by

$$\begin{aligned} \Pi_{\tilde{U}}^{N} &= \text{N-Best Paths}(\Pi_{\tilde{U}} \circ H) \\ \mathcal{B} &= \Pi_{\tilde{U}}^{N} \circ \Phi \circ Y \circ \Omega \circ S. \end{aligned} \tag{4.6}$$

Alignment Lattice Example Figure 4.4 shows a heavily pruned TTM alignment lattice for an English-French sentence pair. Each transition in this lattice has the format $u : v/c$ where u is the English phrase, v is a French phrase, and c is a cost. Word alignments within each phrase-pair are shown in Table 4.1.

4.2 Source Language Texts, Bitexts, and Phrase-Pair Inventories

4.2.1 French-to-English Hansards

The Canadian Hansards are the official records of the Canadian parliament [87] maintained in both English and French. Our corpus consists of a subset of $48,739$ French-English sentence pairs from the Hansards [82]. The French side of the bitext contains $816,545$ words ($24,096$ unique tokens). The English side has a total of $743,633$ words ($18,430$ unique tokens) and is used to train the source language model. The test set consists of 500 unseen French sentences from Hansards for which both reference translations and word alignments are available [82].

On this task our phrase-pair inventory is found as described in Section 3.3 and consists of $772,691$ entries, with $473,741$ unique target phrases and $434,014$ unique source phrases. We restrict the phrase-pairs to the target phrases which have at most 5 words. The distribution of the number of words in the source and target phrases over the inventory is shown in Table 4.2.

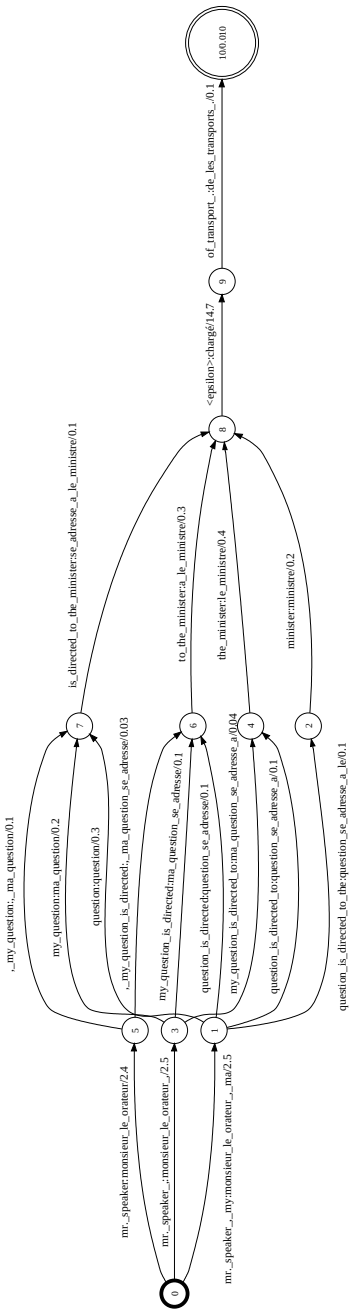

Figure 4.4: A heavily pruned alignment lattice for an English-French sentence pair. English: *Mr speaker , my question is directed to the minister of transport .* French: *Monsieur le orateur , ma question se adresse a le ministre charge de les transports .* Each transition in this lattice has the format $u : v/c$ where u is the English phrase, v is the French phrase, and c is a cost. Word alignments within each phrase pair are shown in Table 4.1.

English Phrase	French Phrase	Word Alignment B (Link set)
Mr. Speaker	Monsieur le orateur	$(0,0),(1,1),(1,2)$
Mr. Speaker ,	Monsieur le orateur ,	$(0,0),(1,1),(1,2),(2,3)$
Mr. Speaker , my	Monsieur le orateur , ma	$(0,0),(1,1),(1,2),(2,3),(3,4)$
, my question	, ma question	$(0,0),(1,1),(2,2)$
my question	ma question	$(0,0),(1,1)$
question	question	$(0,0)$
, my question is directed	, ma question se adresse	$(0,0),(1,1),(2,2)$ $(3,3),(4,3),(4,4)$
my question is directed	ma question se adresse	$(0,0),(1,1),(2,2),(3,2),(3,3)$
question is directed	question se adresse	$(0,0),(1,1),(2,1),(2,2)$
my question is directed to	ma question se adresse a	$(0,0),(1,1),(2,2)$ $(3,2),(3,3),(4,4)$
question is directed to	question se adresse a	$(0,0),(1,1),(2,1),(2,2),(3,3)$
question is directed to the	question se adresse a le	$(0,0),(1,1),(2,1)$ $(2,2),(3,3),(4,4)$
is directed to the minister	se adresse a le ministre	$(0,0),(1,0),(1,1)$ $(2,2),(3,3),(4,4)$
to the minister	a le ministre	$(0,0),(1,1),(2,2)$
the minister	le ministre	$(0,0),(1,1)$
minister	ministre	$(0,0),(1,1)$
of transport	de les transports	$(0,0),(1,1),(1,2)$

Table 4.1: Word alignments within each phrase-pair present in the example Alignment Lattice (Figure 4.4). Each word alignment is shown as a bag of links; a link (i, j) indicates that the English word e_i is linked to the French word f_j within that phrase-pair.

4.2.2 Chinese-to-English FBIS

The Foreign Broadcast Information Service (FBIS) daily reports (available from [17]) consist of broadcasts, news agency transmissions, newspapers, periodicals and government statements from nations around the globe. These media sources are monitored in their original language, translated into English by FBIS, and issued by an agency of the U.S. Government. The FBIS corpus therefore differs from the Hansards French-English corpus in that it is derived from texts of various genres.

The FBIS Chinese-English parallel corpus [57] consists of 9.76M words (49,108 unique tokens) in English and 7.82M words (55,767 unique tokens) in Chinese. The

Target Phrase Length (in French words)	Source Phrase Length (in English words)							
	1	2	3	4	5	6-7	8-10	≥ 11
1	**414.3**	53.1	10.7	2.1	0.5	0.1	0.0	0.0
2	102.8	**190.0**	44.3	12.1	3.2	1.1	0.1	0.0
3	27.8	89.9	**119.5**	35.0	10.8	4.7	0.5	0.0
4	6.8	30.1	73.6	**79.1**	27.7	13.6	1.9	0.1
5	1.7	9.9	29.4	**57.2**	55.5	31.8	5.7	0.4

Table 4.2: Distribution of the number of words in the target and the source phrases over the Phrase-Pair Inventory on the French-English Task. The entries are phrase-pair counts in multiples of 1000, and the bold entries denote the maximum count in each row.

.

Chinese side of the corpus is segmented into words using the LDC word segmenter [51]. The original bitext is aligned at the document level; documents are aligned automatically into chunk-pairs using a statistical chunk model [20] to generate $440,000$ chunk pairs; on an average there are 38 chunk pairs per document pair, 1.72 chunks per English sentence in each document, and 22 sentences per document pair. Our language model training data comes from English news text derived from two sources: online archives (Sept 1998 to Feb 2002) of The People's Daily (16.9M words) [18] and the English side of the Xinhua Chinese-English parallel corpus [54] (4.3M words). The total language model corpus size is 21M words.

Our translation test set is the NIST 2002 MT evaluation set [60] consisting of 878 sentences. Each Chinese sentence in this set has four reference translations. Our alignment test set consists of 124 sentences from the NIST 2001 dry-run MT-eval set [52] that are word aligned manually.

On this task our phrase-pair inventory is found as described in Section 3.3 and consists of 8.05M entries, with 3.12M unique target phrases and 4.98M unique source phrases. We restrict the phrase-pairs to the target phrases which have at most 5 words. The distribution of the number of words in the source and target phrases over the inventory is shown in Table 4.3.

Target Phrase Length (in Chinese words)	Source Phrase Length (in English words)							
	1	2	3	4	5	6	7-8	≥ 9
1	**3,142.3**	1,720.3	775.3	266.2	80.1	24.6	12.2	4.0
2	705.3	**1,461.5**	1,134.8	635.5	295.4	123.2	69.1	18.6
3	149.4	479.1	**781.0**	696.2	461.9	262.6	201.7	64.9
4	34.1	130.7	300.5	**451.3**	441.1	340.7	359.7	162.5
5	9.1	34.2	95.8	196.1	284.0	**300.2**	449.4	314.3

Table 4.3: Distribution of the number of words in the target and the source phrases over the Phrase-Pair Inventory on the Chinese-English Task. The entries are phrase-pair counts in multiples of 1000, and the bold entries denote the maximum count in each row.

.

4.3 Experiments

Given a pair of translations, the goal of bitext word alignment is to find word-to-word correspondences between these sentences. Performance is measured with respect to a reference word alignment created by a competent human translator, and we measure the alignment performance against the reference alignment using Alignment Precision, Alignment Recall, and Alignment Error Rate metrics [82].

We first present definitions of alignment metrics using the word alignment definitions introduced in Section 4.1. Alignment metrics allow us to measure the quality of an automatic word alignment B' relative to a reference alignment B. Alignment Precision is defined as the fraction of links in the automatic alignment which are also in the reference alignment. Alignment Recall is the fraction of links in the reference alignment that are also in the automatic alignment. Alignment Error Rate (AER) is the fraction of links by which the automatic alignment differs from the reference alignment. In all these measurements, links to the NULL word are ignored. This is done by defining modified link sets for the reference alignment: $\bar{B} = B - \{(i,j) : i = 0 \text{ or } j = 0\}$ and the automatic alignment: $\bar{B}' = B' - \{(i',j') : i' = 0 \text{ or } j' = 0\}$.

The reference annotation procedure allows the human transcribers to identify

which links in $\bar{B}$ they judge to be unambiguous. In addition to the reference alignment, this gives a set of *sure links* (S) which is a subset of $\bar{B}$. The alignment metrics are defined as follows [82]

$$\text{Alignment Precision } (S, B; B') = \frac{|\bar{B}' \cap \bar{B}|}{|\bar{B}'|} \tag{4.7}$$

$$\text{Alignment Recall } (S, B; B') = \frac{|\bar{B}' \cap S|}{|S|} \tag{4.8}$$

$$\text{AER } (S, B; B') = 1 - \frac{|\bar{B}' \cap S| + |\bar{B}' \cap \bar{B}|}{|\bar{B}'| + |S|}. \tag{4.9}$$

We present word alignment performance of the WFST translation model on the two alignment tasks in Table 4.4. For comparison, we also align the bitext using IBM-4 word translation models [10][82] trained in both translation directions (IBM-4 F and IBM-4 E), and their union (IBM-4 $E \cup F$). For all the TTM experiments presented here, we will use the Fixed Phrase Order Model (Equation 3.7). We will justify the choice of this model through the experiments in Section 4.3.6.

Model	Alignment Metrics (%)					
	French-English			Chinese-English		
	Precision	Recall	AER	Precision	Recall	AER
IBM-4 F	89.4	90.5	10.1	82.8	48.0	39.2
IBM-4 E	89.6	90.0	10.2	73.9	58.3	34.9
IBM-4 $E \cup F$	84.5	94.5	11.7	66.0	63.1	35.5
TTM	94.5	84.6	9.9	89.0	37.7	47.0

Table 4.4: TTM Alignment Performance on the French-English and the Chinese-English Alignment Tasks.

We note that the alignment error rate of the TTM is comparable to the IBM-4 models on the French-English task, but worse than IBM-4 models on the Chinese-English task. On both tasks the model obtains a very high Alignment Precision but a relatively poor Alignment Recall. The high alignment precision suggests that the word alignments within the phrase-pairs are very accurate. However, the poor performance under the recall measure suggests that the phrase-pair inventory has relatively poor coverage of the phrases in the alignment test set.

Alignment Recall is influenced by the words in the source and the target language sentences which are either spontaneously inserted or deleted during word alignment. In analyzing the French-English word alignments, we found that on average, 32.5% of the target-phrases are inserted and 34.1% of the source phrases are deleted. On the Chinese-English task, 51.6% of the target-phrases are inserted and 52.1% of the source phrases are inserted. Clearly the Alignment Recall in the Chinese-English will therefore be much lower than in French-English, whereas the Alignment Precision degrades only slightly. An additional factor that affects the Alignment Recall is the presence of words in the test set that are unseen in training. These are treated as single word phrases and are left out of the alignment, thus reducing the Alignment Recall.

4.3.1 Phrase Exclusion Probability

MAP word alignment under the TTM is affected by the number of target and source phrases that are excluded during bitext word alignment; this behavior is governed by the Phrase Exclusion Probability (PEP) as described in Section 4.1.2. We will now measure word alignment quality as a function of PEP (α) (Figure 4.5). In Figure 4.5 we observe that Alignment Precision increases monotonically with PEP over most of its permissible range, however there is a critical value above which Alignment Precision decreases. Alignment Recall at first improves slightly with PEP but then decreases. AER closely follows the Alignment Recall.

We now study this behavior more closely. The TTM is constructed so that as PEP (α) increases, the likelihood of excluding phrases increases. To assess this, we measure the percentage of Excluded Phrase Counts (EPC) which is the ratio of the number of source and target phrases excluded under the MAP alignment to the total number of transductions (phrase-pair transductions, spontaneous insertions of target phrases, and deletions of source phrases) in the MAP alignment. In Figure 4.6f, we see that EPC is in fact increasing in PEP. We see furthermore that there is a critical value above which EPC increases rapidly; at this point the model simply finds it more likely to exclude phrases rather than align them. This has a direct influence on

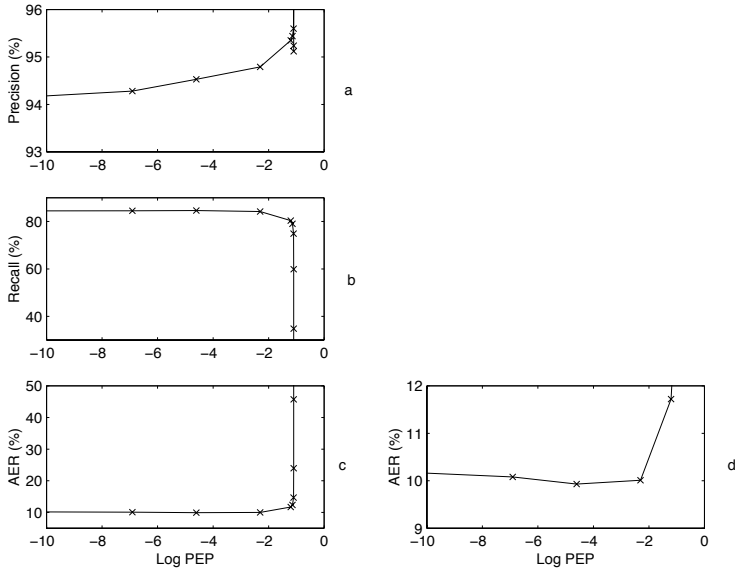

Figure 4.5: Alignment Performance of TTM as a function of Phrase Exclusion Probability (PEP). For each value of PEP, we measure Precision (Panel a), Recall (Panel b), and AER (Panel c). Results are shown on the French-English task. The plot in Panel d focuses in on the values of PEP where AER attains the minimum.

Alignment Recall (Equation 4.8), which is proportional to the number of correctly aligned words. This quantity is necessarily dominated by the number of aligned phrases, so that Alignment Recall falls off sharply with a sharp rise in EPC.

The influence of PEP on Alignment Precision is more complex. As PEP increases, the model is able to avoid aligned phrase pairs whose transduction probability is low. As a result, the phrase pairs that remain in the alignment are those with higher phrase transduction likelihoods. For each aligned phrase pair, this quantity is based simply on the relative frequencies of their occurrences in the bitext word alignments (see Section 3.4.5). As PEP increases, the alignment favors source language phrases that are uniquely aligned to one target phrase. It is plausible that the word alignments within these phrase pairs are of higher quality than found in general. This would explain the increase in Alignment Precision at intermediate values of PEP.

For PEP above the critical point, we observe a decrease in Alignment Precision

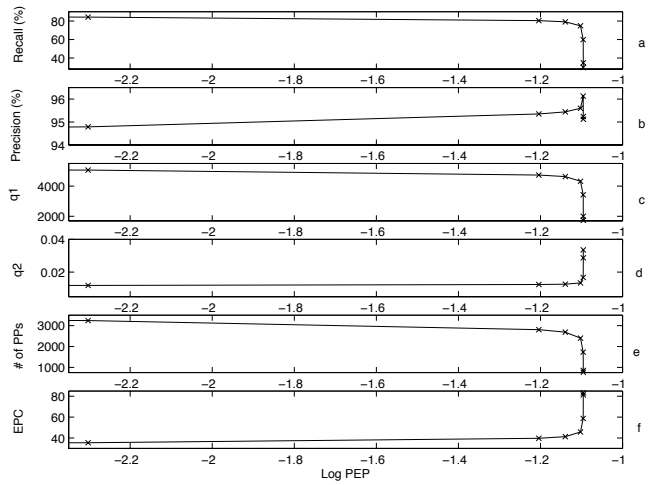

Figure 4.6: Variation of Alignment Precision (Panel b) and Recall (Panel a) for values of Phrase Exclusion Probability (PEP) near the critical value. We also plot four additional quantities derived from the MAP alignment. These include the number of wrongly hypothesized links q_1 (Panel c), penalty per incorrectly hypothesized alignment link q_2 (Panel d), the number of phrase-pair transductions (Panel e), and the percentage of Excluded Phrase Counts (Panel f). Results are shown on the French-English task.

(Figure 4.6e). To analyze this behavior, we write Alignment Precision as

$$\text{Alignment Precision}(S, B; B') = \frac{|\bar{B}' \cap \bar{B}|}{|\bar{B}'|} \\ 1 - q_1 q_2,$$

where $q_1 = |\bar{B}'| - |\bar{B}' \cap \bar{B}|$ and $q_2 = \frac{1}{|\bar{B}'|}$. Considered in this way, q_1 is the number of incorrectly hypothesized alignment links, and q_2 is the penalty associated with each wrong alignment link; this penalty decreases inversely with the number of hypothesized links. The interaction between q_1 and q_2 as PEP varies will determine the Alignment Precision. In Figure 4.6, we see that as EPC increases (Figure 4.6f) the absolute number of phrase-pairs in the alignment decreases (Figure 4.6e). The quantity q_2 (Figure 4.6d) can be expected to vary inversely with the number of aligned phrase pairs, and we in fact observe this behavior. We separately measure q_1, the number of incorrectly hypothesized alignment links, and find that this number does decrease for PEP above the critical value (Figure 4.6c), suggesting that the relatively

few phrase pairs that remain in the alignments are of high quality. However we see that the Alignment Precision (Figure 4.6b) is dominated by q_2 so that performance falls for PEP above the critical value.

Conclusion Alignment Recall decreases with Phrase Exclusion Probability (PEP) due to an increase in the percentage of excluded phrases (Figure 4.5b). Alignment Precision increases at intermediate values of PEP because of a decrease in the number of incorrectly hypothesized alignment links within the phrase-pairs (Figure 4.5a). However, beyond a critical value of PEP, Alignment Precision decreases due to a sharp decrease in the number of hypothesized alignment links (Figure 4.6b).

4.3.2 Richness of the Phrase-Pair Inventory

It has been established [82] that alignment performance of IBM-4 models improves as the size of the bitext training set grows. In contrast, the alignment performance of the TTM is more complex. The phrase-pair inventory is created from a set of word alignments generated by underlying IBM-4 models so that the TTM alignment performance depends, in part, on the quality of the underlying word alignments. In addition, the TTM alignment performance also depends on the richness of the phrase-pair inventory which determines coverage of the test set. We here perform experiments to tease apart these two factors.

In this section we study the effect of richness of phrase-pair inventory on word alignment quality. For this purpose, we train IBM-4 translation models on the 48K French-English Hansards bitext collection (Section 4.2) and obtain word alignments over this set. We then construct four subsets of the bitext word alignments consisting of 5K, 12K, 24K, and 48K sentence-pairs respectively. From each subset, we extract a phrase-pair inventory (using the procedure described in Section 3.3). Statistics over the four phrase-pair inventories are shown in Table 4.5. We measure coverage by each inventory of the test set in the following way. We first obtain all the target language phrases (up to a maximum length) in the test set, and then measure the percentage of these phrases that also occur in the phrase-pair inventory. A higher value indicates

a better coverage of the test set under the particular inventory.

Subset ID	# of Sentence Pairs	Phrase-Pair Inventory Statistics			Coverage (%)
		# of Target Phrases	# of Source Phrases	# of Phrase Pairs	
PPI-1	5K	81,640	73,283	122,621	20.79
PPI-2	12K	167,752	151,534	259,010	26.75
PPI-3	24K	288,395	261,685	456,946	31.35
PPI-4	48K	473,741	434,014	772,691	36.02

Table 4.5: Statistics over Phrase-Pair Inventories extracted from four subsets of the French-English Hansards. IBM-4 models are trained on 48K sentence-pairs from Hansards and word alignments are obtained on the training set. Four phrase-pair inventories are then constructed from four nested subsets of these word alignments. The coverage by each inventory of the test set is also reported.

Using the PPIs as described in Table 4.5, we construct four different TTMs and use each to obtain MAP word alignments of the test set (Equation 4.1). In Figure 4.7 we show the Alignment Precision, Alignment Recall, and AER as a function of Phrase Exclusion Probability (α), for values below the critical value. Examining these results shows that Alignment Precision changes only slightly with an increase in the size of the phrase-pair inventory (Figure 4.7a). However Alignment Recall decreases dramatically as the size of the phrase-inventory is reduced (Figure 4.7b). AER is dominated by the decrease in Alignment Recall and increases with a reduction in the size of the inventory (Figure 4.7c). The variation in all three alignment metrics with respect to Phrase Exclusion Probability (PEP) is identical for all the four subsets.

We first explain the variation in Alignment Precision as the size of the phrase-pair inventory is reduced. We note that the four phrase-pair inventories are extracted from word alignments generated by the same IBM-4 models. Therefore the word alignments within the phrase-pair inventories are of uniform quality; this in turn suggests that the word alignments generated by the TTM will yield nearly identical Alignment Precision regardless of the size of the inventory employed. We explain the variation in Alignment Recall across the four inventories by measuring the coverage of the inventories on the test set. As the size of the underlying phrase-pair inventory is reduced, the coverage of test set drops as seen in Table 4.5. Alignment Recall

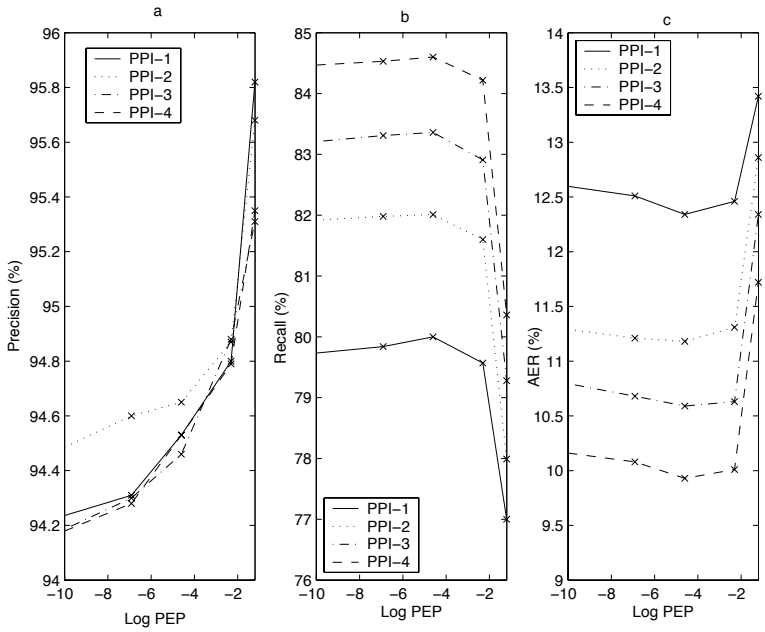

Figure 4.7: Effect of phrase-pair inventory size on TTM word alignment quality. IBM-4 models are trained on 48K sentence-pairs from French-English Hansards and word alignments are obtained over the collection. Four subsets are constructed from this set of word alignments and phrase-pair inventories were collected over each subset. For each inventory, MAP word alignments under the TTM are obtained, and Alignment Precision (Panel a), Alignment Recall (Panel b), and AER (Panel c) are measured as functions of Phrase Exclusion Probability (PEP). Inventories are shown in Table 4.5.

(Equation 4.8) is proportional to the number of correctly aligned words on the test set and is therefore dependent on the coverage by the inventory of the test set. This suggests that as the size of the phrase-pair inventory is reduced, Alignment Recall will decrease due to a decrease in test set coverage.

Conclusion We hold word alignment quality of IBM-4 translation models constant when constructing phrase-pair inventories from different sizes of bitext. Increasing the bitext size improves the coverage by the inventory of the phrases on the test set (Table 4.5), and consequently improves Alignment Recall of the TTM (Figure 4.7). However, the Alignment Precision of the TTM remains invariant to the size of the bitext because the word alignments within the phrase-pairs are of uniform quality.

4.3.3 Word Alignment Quality of Underlying IBM-4 Models

In the previous experiment, the quality of the underlying word alignments is held constant while we vary the size of the bitext from which the phrase-pair inventories are extracted. As an alternative, we would like to fix the size of the phrase-pair inventory and allow the underlying word alignments to vary in quality. However this is not possible, since the phrase-pair inventories themselves are extracted from word alignments. We take the following approach. We constructed systems over varying amounts of bitext and adjust the PEP (α) so that two different sized systems have the same Alignment Recall; this implies that they have comparable coverage. At these points we will measure Alignment Precision and AER.

For this experiment, we construct four nested subsets of the Hansards bitext collection containing 5K, 12K, 22K, and 48K sentence pairs respectively; these are the same four subsets used in the previous experiment. On each subset, we trained IBM-4 translation models and used these models to obtain word alignments over the smallest (5K) subset. From each set of word alignments over the 5K subset, we construct a phrase-pair inventory using the procedure described in Section 3.3. Statistics over these four phrase-pair inventories are shown in Table 4.6. We note that this experiment allows us to hold the coverage of target phrases nearly constant across the four inventories; the variation in coverage across the four inventories (Table 4.6) is lower compared to the corresponding variation in coverage values for the inventories used in the previous experiment (Table 4.5).

Using the four phrase-pair inventories as described in Table 4.6, we construct four different TTMs and use each to obtain a MAP word alignment of the test set. In Figure 4.8, we study Alignment Precision, Alignment Recall, and AER as a function of PEP for values below the critical value. Contrary to the previous experiment in which we alignment quality was held constant, we observe that as the size of bitext increases, the Alignment Precision improves (Figure 4.8a). We see that Alignment Recall also improves with the size of the bitext (Figure 4.8b). AER reflects the combined Alignment Precision and Recall, and improves consistently as the bitext size is increased (Figure 4.8c). The variation in alignment performance (precision,

Subset ID	# of Sentence Pairs	Phrase-Pair Inventory Statistics			AER (%) of IBM-4 models	Coverage (%)
		# of Target Phrases	# of Source Phrases	# of Phrase Pairs		
PPI-1	5K	58,266	51,318	80,256	20.6	18.14
PPI-2	12K	67,242	59,138	95,953	15.9	19.39
PPI-3	24K	74,526	65,856	108,952	13.9	20.23
PPI-4	48K	81,800	73,442	123,314	12.1	20.86

Table 4.6: Statistics over four different Phrase-Pair Inventories collected from a 5K subset of the French-English Hansards. IBM-4 models are trained on four nested subsets of the French-English Hansards bitext and word alignments are obtained over the smallest (5K) subset. A phrase-pair inventory is collected over each word alignment. The alignment quality (in AER) of each underlying IBM-4 model and the coverage by each inventory of the test set are reported.

recall and AER) with respect to Phrase Exclusion Probability is seen to be identical for all the four subsets.

Subset ID	Bitext Size	$log(PEP)$	Precision (%)	Recall (%)	AER (%)
PPI-1	5K	-2.30	94.0	75.0	15.5
PPI-2	12K	-1.25	94.8	75.0	15.1

Table 4.7: Analysis of the effect of Word Alignment Quality on TTM Alignment Performance. We select two systems from Figure 4.8 with constant Alignment Recall, and measure Alignment Precision and AER for these systems.

To understand this behavior, we note that as the size of bitext is increased, the alignment performance of the IBM-4 models improves (Table 4.6). We therefore attribute the increase in Alignment Precision to the improvement of the underlying word alignments. We attempt to measure Alignment Precision for constant values of Alignment Recall. Table 4.7 presents PPI-1 at $log(PEP) = -2.30$ and PPI-2 at $log(PEP) = -1.25$. We observe that although the Alignment Recall values for the two systems are equal, the PPI-1 system has a lower Alignment Precision than PPI-2. Since the underlying models for PPI-1 are trained on approximately half the number of sentence pairs as the underlying models for PPI-2, we conclude that if Alignment Recall can be held constant, the effect of increasing bitext is to improve Alignment Precision of the TTM.

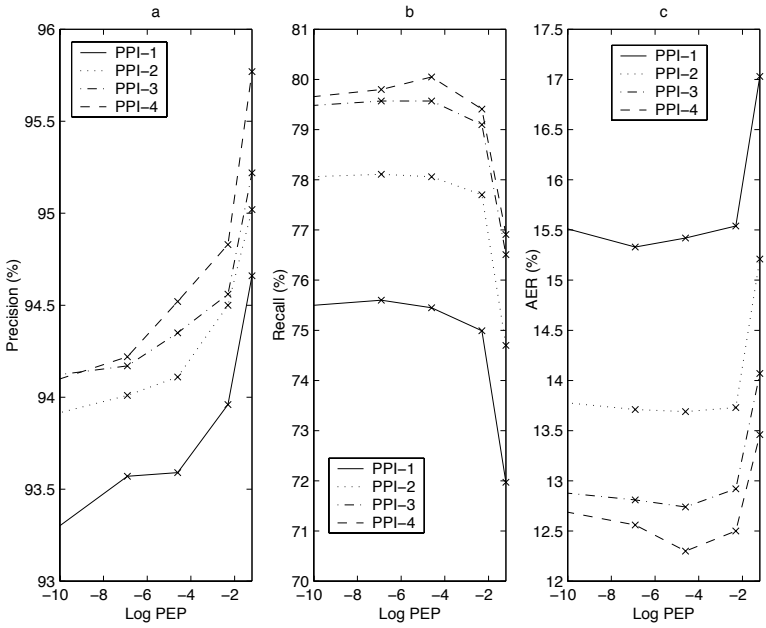

Figure 4.8: Effect of word alignment quality of underlying IBM-4 models on alignment performance of TTM. IBM-4 models are trained on four nested subsets of the French-English Hansards bitext and word alignments are obtained over the smallest subset (5K sentence pairs). A phrase-pair inventory are constructed over each word alignment. For each inventory, MAP word alignments under the TTM are obtained, and Alignment Precision (Panel a), Alignment Recall (Panel b), and AER (Panel c) are measured as functions of Phrase Exclusion Probability. Inventories are shown in Table 4.6.

Conclusion We build phrase-pair inventories from IBM-4 word alignments of varying quality while keeping coverage of phrases on a test set nearly constant (Table 4.6). In this case increasing the bitext size for IBM-4 model training improves the word alignment quality of IBM-4 models and consequently improves the Alignment Precision of the TTM (Figure 4.8). However, Alignment Recall stays nearly constant because the coverage of the test set does not change much across the inventories.

4.3.4 Multiple Source Phrase Segmentations

Ideally the word alignment of sentence pairs under the TTM is obtained after considering all possible phrase segmentations of the source sentence (Equation 4.1).

An alternative, approximate approach could be done following the two-step procedure (Equation 4.2) that consists of MAP phrase segmentation of the source sentence, then followed by the MAP alignment of the fixed source sentence phrase segmentation. Figure 4.9 compares the performance of the two approaches as a function of the Phrase Exclusion Probability for values above the critical value. We find that the two-step approach (Equation 4.2) is markedly inferior relative to the exact MAP word alignment (Equation 4.1).

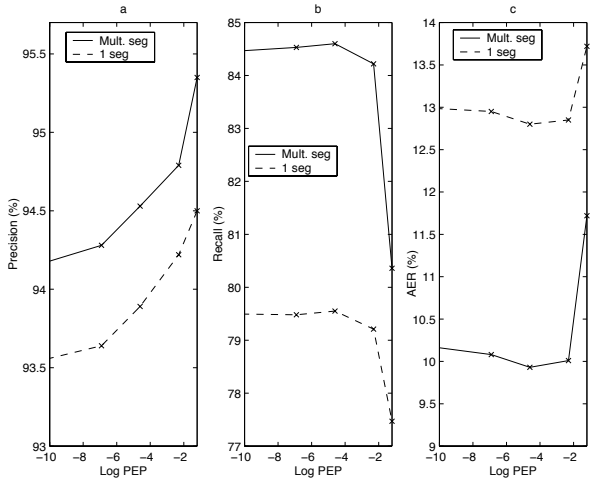

Figure 4.9: Effect of multiple phrase segmentations of the source sentence on TTM word alignment quality. MAP Word Alignments Under the TTM are obtained using the two-step alignment process (Equation 4.2) that considers only a single phrase segmentation of the source sentence These are compared to MAP word alignments obtained using all segmentations of the source sentence (Equation 4.1). In both cases, Alignment Precision (Panel a), Alignment Recall (Panel b), and AER (Panel c) are measured are functions of Phrase Exclusion Probability.

Conclusion Considering multiple phrase segmentations of the source sentence during TTM alignment yields large improvements in alignment performance (Precision, Recall, and AER) relative to the case where only a single phrase segmentation is used (FIgure 4.9).

4.3.5 Unweighted Source Phrase Segmentation Model

In Section 3.4, we described an exact procedure to ensure that the probabilities over all segmentations of a given length sentence are correctly normalized. As this procedure is expensive in practice, we consider excluding the source phrase segmentation model in the following way. We obtain word alignments under the TTM using an unweighted source phrase segmentation model, i.e. a source phrase segmentation transducer W is constructed as in Section 3.4.2 but no weights are assigned to the transitions. The MAP word alignments without the source phrase segmentation model are compared to the exact MAP word alignments in Table 4.8.

We observe that excluding the segmentation model has almost no impact on the alignment quality. We can therefore avoid this expensive step in practice.

Source Phrase Segmentation Model	Alignment Metrics (%)		
	Precision	Recall	AER
Weighted	94.5	84.6	9.9
Unweighted	94.4	84.6	10.0

Table 4.8: Effect of an Unweighted Source Segmentation Model on TTM Alignment Quality. Results are shown on the French-English Hansards Task.

Conclusion The source phrase segmentation model likelihoods do not influence the alignment performance of the TTM (Table 4.8). We therefore conclude that this particular instance of the segmentation model is so weak that the overall alignment performance is dominated by the phrase transduction probabilities.

4.3.6 Source Phrase Reorderings

In the experiments described thus far we have used the Fixed Phrase Order Model (Equation 3.7) that does not reorder the source phrase sequence while performing word alignment (Equation 4.1). We now measure the effect of reorderings of the MAP source phrase segmentation on alignment performance of the TTM.

We follow the procedure described earlier (Section 4.1) and obtain an N-best list of reorderings under the Markov Phrase Order model (Equation 3.6). Word alignment

of each sentence-pair under the TTM (Equation 4.1) is then performed given the N-best reorderings of the source phrase sequence.

We first derive a quantity that characterizes the tendency of the model to relocate phrases in order to achieve the MAP word alignment. This quantity, called Average Phrase Movement (APM) [79], measures the degree of non-monotonicity in the MAP word alignment (Equation 4.1). Suppose any two consecutive phrases in the reordered source phrase sequence $\hat{u}_{\hat{a}_1}, ..., \hat{u}_{\hat{a}_k}$ are given by $\hat{u}_{\hat{a}_k} = e_l^{l'}$ and $\hat{u}_{\hat{a}_{k-1}} = e_m^{m'}$, the movement between these phrases is measured as $d_k = |l - m' - 1|$. The total phrase movement over the sentence pair is taken as the sum of the individual movements: $d = \sum_{k=1}^{K} d_k$. The Average Phrase Movement is obtained by averaging the total movement over the sentences in the test set. We emphasize that the target phrase order is unchanged during the alignment process, so the Average Phrase Movement measures variation in the source phrase order relative to both the original source phrase order and the target phrase order.

We perform two experiments to study the effect of reorderings on TTM word alignments. In the first experiment, we fix the number of reordered source phrase sequences (an N-best list of size 400) and obtain MAP word alignments under the TTM as a function of PEP (α) (Figure 4.10). For each PEP we also measure the percentage of Excluded Phrase Counts (EPC). We observe that there is only a slight improvement of AER by allowing reorderings relative to the no reordering case. When reorderings are allowed, the Average Phrase Movement drops monotonically as PEP is increased. We also note the AER peaks at the same value of PEP whether or not reordering of the source phrase sequence is allowed.

Our conclusion is that to induce phrase reorderings in the MAP word alignment, PEP must be set to a value that leads to a degradation in AER. In contrast, at the optimal value of AER, we observe that the Average Phrase Movement of the MAP word alignment is less than one word; this suggests that we could obtain similar gains in AER by increasing the maximum word length of the source phrases in the phrase-pair inventory instead of allowing source phrase reorderings during alignment.

In the second experiment we fix the Phrase Exclusion Probability at its optimal value from the the first experiment (PEP = 0.005), and then obtain MAP word

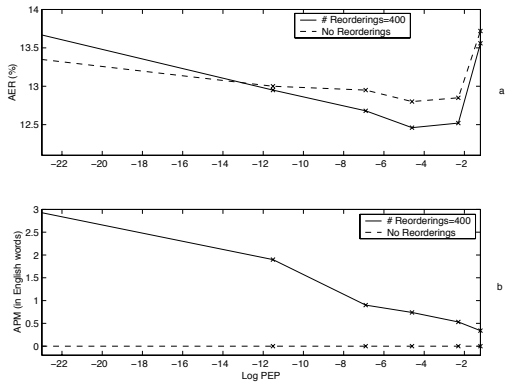

Figure 4.10: Effect of Reorderings of the Source Phrase Sequence on TTM alignment quality. MAP Word Alignments under the TTM are obtain using a fixed number of reorderings ($N = 400$) of the single phrase segmentation of the source sentence. Performance is compared with MAP word alignments obtained without reordering the source phrase sequence. We measure AER (Panel a) and Average Phrase Movement (Panel b) as functions of the Phrase Exclusion Probability (PEP). Results are shown on the French-English Task.

alignments under the TTM as the number of reordered source phrase sequences is varied (Table 4.9). For comparison we also show the performance when the source phrase is not reordered in computing the MAP word alignment. As the number of reordered source phrase sequences is increased from 1 to $1,000$, we note that the Average Phrase Movement increases slightly. When reorderings are allowed, there is a slight reduction in EPC relative to the no-reordering case. AER decreases only slightly by allowing more reorderings of the source phrase sequence during alignment.

We conclude from the second experiment that allowing more reorderings leads to a greater Average Phrase Movement in the MAP alignment. In addition this also allows more phrase pairs to be aligned as seen by the reduction in the percentage of Excluded Phrase Counts (EPC). However, there are only small gains in AER by allowing reorderings of the source phrase sequence.

Conclusion Alignment Performance of the TTM does not improve much by allowing movement of source phrases during word alignment (Figure 4.10). Most the gains in performance can be obtained using a 100-best list of reorderings. This experiment

# of reordered source phrase sequences	Alignment Metrics (%) Precision	Recall	AER	Average Phrase Movement	EPC (%)
No Reordering	93.9	79.5	12.8	0.0	34.8
1	93.9	79.5	12.8	0.2	34.8
100	94.0	80.1	12.5	0.7	34.2
200	94.0	80.1	12.5	0.7	34.2
400	94.0	80.1	12.5	0.7	34.2
600	94.0	80.1	12.5	0.8	34.2
800	94.0	80.1	12.5	0.8	34.2
1000	94.0	80.1	12.5	0.8	34.2

Table 4.9: Effect of number of reorderings of the source phrase sequence on TTM alignment quality. MAP word alignments under the TTM is obtained as a function of the number of reorderings of the source phrase sequence in the French-English Task. In each case, we measure Alignment Quality (Precision, Recall and AER), Average Phrase Movement and the percentage of Excluded Phrase Counts (EPC).

provides evidence that we can avoid reordering the source phrase sequence without much degradation in alignment performance.

4.4 Discussion

In this chapter we have discussed word alignment of bitexts under the Translation Template Model. Once the component models of the TTM are implemented as weighted finite state transducers, we have shown how MAP word alignment can be obtained immediately using standard weighted finite state operations involving these transducers. In addition, these WFST operations facilitate generation of alignment lattices without any extra effort in implementation.

This is the first time that phrase-based models of this variety have been employed for bitext word alignment. The ability to do this is crucial in order to implement iterative parameter estimation procedures such as Expectation Maximization (EM) for this model. In general we note that a finite inventory of phrase-pairs is not rich enough to cover all possible sentences in any given bitext collection. As a result sentence-pairs from the collection can contain phrase-pairs not in the inventory; unless addressed, these sentence-pairs are therefore assigned a probability of zero under the

model. We have described how modeling the deletion of source phrases during the alignment process can overcome this limitation, and thus make it possible for the TTM to be used to align any bitext.

We have presented a detailed experimental analysis of the TTM, and analyzed several factors that influence alignment performance. Our experiments are aimed at throwing light on the strengths and weaknesses of the model. We will now highlight some of the key results and conclusions that we draw from these experiments.

We observe that the Alignment Error Rate (AER) of the TTM is comparable to the IBM-4 models on the French-English task, but worse than that of the IBM-4 models on the Chinese-English task. On both tasks the model obtains a very high Alignment Precision but a relatively poor Alignment Recall. The lower recall on the Chinese-English task can be attributed to the greater number of source language and target language phrases that are excluded during word alignment under the TTM.

Source and target phrases excluded during word alignment affect alignment performance of the TTM. This behavior is governed by varying the Phrase Exclusion Probability (PEP). Alignment Recall at first improves slightly with PEP but then decreases. Alignment Precision increases monotonically with PEP over most of its permissible range, however there is a critical value above which Alignment Precision decreases. The initial increase in Alignment Precision suggests that as PEP increases the model favors phrase-pairs that yield higher quality word alignments than found in general. However as PEP is increased above the critical value, the percentage of excluded phrases increases sharply. As a result, the Alignment Precision drops even though the relatively few phrase-pairs that remain in the alignments are of high quality. We conclude from this behavior that we cannot 'game' Alignment Precision by arbitrarily decreasing the number of hypothesized alignment links.

The quality of underlying word alignments and the richness of the phrase-pair inventory both influence alignment performance of the TTM. If the underlying word alignment quality is held constant, the main influence of increasing bitext size is to increase phrase-pair coverage and consequently improve Alignment Recall. By contrast, if Alignment Recall can be held constant, the effect of increasing bitext size is to improve Alignment Precision of the TTM.

All phrase segmentations of the source sentence are generally considered when obtaining the MAP word alignment of sentence pairs under the TTM. An alternate approach is a two step procedure that consists of MAP phrase segmentation of the source sentence, followed by the MAP alignment of the fixed source phrase segmentation. We find that this two-step approach is markedly inferior relative to the exact MAP word alignment.

Excluding the source segmentation model has almost no impact on the alignment quality of the TTM. We conclude that this particular instance of the segmentation model is so weak that the overall alignment process is dominated by the phrase translation probabilities.

Reorderings of the source phrase sequence can be allowed during TTM word alignment. However there is only a slight improvement in AER by allowing any reordering. We observe that to induce phrase reorderings in the MAP word alignment, PEP must be set to a value that leads to a degradation in AER. In contrast, at the optimal value of AER, the Average Phrase Movement of the MAP word alignment is less than one word; this suggests that we can obtain the benefits of phrase reordering by increasing the maximum word length of the source phrases in the phrase-pair inventory. Allowing more reorderings leads to a greater phrase movement in the MAP word alignment. However, this does not result in a large improvement in AER; we find no gains in AER beyond a 100-best listing of reorderings.

Chapter 5

Translation under the Translation Template Model

In this chapter we discuss translation under the Translation Template Model (TTM). We describe how translation under the TTM can be implemented using Weighted Finite State Transducer (WFST) operations involving the TTM component transducers described in Chapter 3. We then evaluate the translation performance of the TTM on the Hansards French-English and FBIS Chinese-English tasks introduced in Section 4.2. We present experiments to study the contribution of model components to the translation performance of the TTM system. We also investigate the influence of both quality and quantity of the training bitext on the translation performance.

This chapter is organized as follows. We first show how translation under the TTM can be performed with standard WFST operations involving the TTM component transducers (Section 5.1). We present investigatory translation experiments in Section 5.2. We then describe the development of a Chinese-to-English TTM system constructed from large training bitexts for the NIST 2004 MT evaluation (Section 5.3). We finally discuss the experiments in Section 5.4.

5.1 WFST Implementation of Translation

Given a target language sentence f_1^J, a translation $\hat{e}_1^{\hat{I}}$ in the source language can be generated via MAP decoding as:

$$\{\hat{e}_1^{\hat{I}}, \hat{K}, \hat{u}_1^{\hat{K}}, \hat{a}_1^{\hat{K}}, \hat{c}_0^{\hat{K}}, \hat{d}_0^{\hat{K}}, \hat{v}_1^{\hat{R}}\} = \underset{e_1^I, K, u_1^K, a_1^K, c_0^K, d_0^K, v_1^R}{\operatorname{argmax}} P(e_1^I, K, u_1^K, a_1^K, c_0^k, d_0^K, v_1^R | f_1^J) \quad (5.1)$$

$$\underset{e_1^I, K, u_1^K, a_1^K, c_0^K, d_0^K, v_1^R}{\operatorname{argmax}} P(e_1^I) P(f_1^J, v_1^R, d_0^K, c_0^K, a_1^K, u_1^K, K | e_1^I).$$

$\hat{u}_1^{\hat{K}}$, $\hat{a}_1^{\hat{K}}$, $\hat{d}_0^{\hat{K}} = \hat{v}_1^{\hat{R}}$ and $\hat{c}_0^{\hat{K}}$ are the corresponding source phrase sequence, source phrase reordering sequence, target phrase sequence, and the sequence that specifies the position and length of spontaneously inserted target phrases within the reordered source phrase sequence; values for all these variables are hypothesized in the decoding process.

In translation we do not consider reorderings of the source phrase sequence due to limitations in the current WFST translation framework. In this case the set of possible translations of f_1^J is obtained using the weighted finite state composition:

$$\mathcal{T} = G \circ U \circ \Phi \circ Y \circ \Omega \circ S.$$

A translation lattice [98, 47] can be generated by pruning $\mathcal{T}$ based on likelihoods or number of states [74]. The translation with the highest probability (Equation 5.1) can be computed by obtaining the path with the highest score in $\mathcal{T}$. Figure 5.1 shows a heavily pruned translation lattice for the French sentence: *Monsieur le orateur , ma question se adresse a le ministre charge de les transports* ·. The N-best (N=5) list of translations from this lattice is shown in Figure 5.2.

5.2 Experiments

We now measure the translation performance of the TTM. In implementing translation under the TTM we use the same components analyzed in our word alignment experiments (Section 4.3). We used the unweighted Source Phrase Segmentation Model (Section 3.4.2). Allowing phrase movement in FSM-based implementations

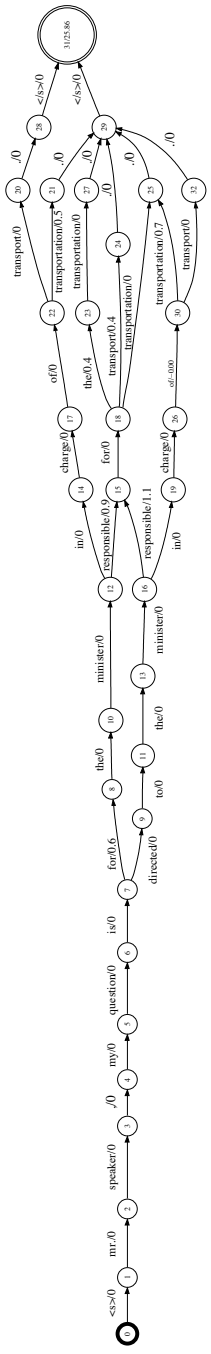

Figure 5.1: A heavily pruned translation lattice for the French sentence: *Monsieur le orateur , ma question se adresse a le ministre charge de les transports .* Each transition in this lattice has the format w/c where w is a word and c is a cost.

Hypothesis	Log Probability
<s> mr. speaker , my question is directed to the minister in charge of transport . </s>	−25.9
<s> mr. speaker , my question is for the minister in charge of transport . </s>	−26.5
<s> mr. speaker , my question is directed to the minister in charge of transportation . </s>	−26.6
<s> mr. speaker , my question is directed to the minister responsible for transportation . </s>	−26.9
<s> mr. speaker , my question is for the minister in charge of transportation . </s>	−27.0

Figure 5.2: An N-best (N=5) list of English translations for the French sentence: *Monsieur le orateur , ma question se adresse a le ministre charge de les transports .* The list is generated from the translation lattice shown in Figure 5.1. <s> and </s> denote the sentence beginning and the sentence end symbols respectively.

such as this is expensive in memory usage [47, 43]. We use the Fixed Phrase Order Model (Section 3.4.3). Translation is performed in monotone phrase order, as has been done by others [109].

Unlike word alignment, translation requires as source language model (Section 3.4.1). Here we use a trigram word language model estimated using modified Kneser-Ney smoothing as implemented in the SRILM tookit [90]. As described in Section 4.2, separate source (English) language models are trained for the French-English and Chinese-English tasks.

Translation performance is measured using the BLEU and NIST MT-eval metrics, and Multi-Reference Word Error Rate (mWER). The NIST and mWER metrics are described at length elsewhere [21] [79], and we will not review them. However we wish to provide a detailed analysis of translation performance under BLEU, so we will review its formulation.

The BLEU score [86] measures the agreement between a hypothesis translation E' and its reference translation E by computing the geometric mean of the precision of their common n-grams. The score also includes a 'Brevity Penalty' $\gamma(E, E')$ that is applied if the hypothesis is shorter than the reference. The functional form is

$$\text{BLEU}(E, E') = \gamma(E, E') \times \text{BPrecision}(E, E') \quad (5.2)$$

$$\text{BPrecision}(E, E') = exp\left(\frac{1}{N}\sum_{n=1}^{N} log\, p_n(E, E')\right) \quad (5.3)$$

$$\gamma(E, E') = \begin{cases} 1 & |E'| \geq |E| \\ e^{(1-|E|/|E'|)} & |E'| < |E| \end{cases} \quad (5.4)$$

In the above equations, $p_n(E, E')$ is a modified precision of n-gram matches in the hypothesis E', and is specified as

$$p_n(E, E') = \frac{\sum_{g \in \mathcal{V}^n} \min(\#_E(g), \#_{E'}(g))}{\sum_{g \in \mathcal{V}^n} \#_{E'}(g)}, \quad (5.5)$$

where $\mathcal{V}^n$ denoted all n-grams (order n), and $\#_E(g)$ and $\#_{E'}(g)$ are the number of occurrences of the n-gram g in the reference E and in the hypothesis E' respectively. We will use the notation BLEUrXnY to refer to BLEU score measured with respect to X reference translations and a maximum n-gram length $N = Y$ in Equation 5.3. The BLEU score (Equations 5.2-5.5) is defined over all sentences in the test set, i.e. E' and E are concatenations of hypothesis (reference) translations over sentences in a test set. We can also define a sentence-level BLEU score between the hypothesis and reference translations of each individual sentence using Equations 5.2-5.5.

We note that it has not been the standard practice in the MT community to measure statistical significance when reporting translation performance under the automatic evaluation metrics. NIST currently provides the MT evaluation software [78] but does not supply any tools for significance testing. We therefore do not perform any tests of significance in the experiments reported in this thesis. We expect NIST to supply the MT community with these tools in the near future.

To serve as a baseline translation system, we use the ReWrite decoder [64] with the French-English and Chinese-English IBM-4 translation models used in creating the phrase-pair inventories. We see in Tables 5.1 that in both the Chinese-English and French-English tasks, the performance of the TTM compares favorably to that of the ReWrite decoder.

Model	French-English		Chinese-English	
	BLEUr1n4 (%)	NISTr1n4	BLEUr4n4 (%)	NISTr4n4
IBM-4	17.09	5.02	9.67	3.57
TTM	22.29	5.52	22.45	7.73

Table 5.1: Translation Performance of the TTM on the French-English and Chinese-English Translation Tasks. For comparison, we also report performance of ReWrite Decoder with the French-English and Chinese-English IBM-4 translation models used to create the Phrase-Pair inventories.

5.2.1 Phrase Exclusion Probability

In Section 4.3, we have seen that the Phrase Exclusion Probability (PEP) strongly influences bitext word alignment quality. We now evaluate the effect of this parameter on translation. The role of PEP in translation is to control spontaneous insertions of target phrases. This is in contrast to word alignment where PEP affects both the spontaneous insertions of target phrases and the deletions of source phrases. We would like to allow the model the flexibility of deleting phrases in sentence to be translated. Within the source-channel model, this is achieved through the insertion of target language phrases. We could also allow the generative model to delete source language phrases, but this would correspond to the insertion of English phrases in translation independent of any evidence in the Chinese or French sentence; in other words, they would be hypothesized entirely by the source language model. We do not consider this scenario.

We now discuss several aspects of Phrase Exclusion Probability in translation. We first observe that there is sensitivity in the BLEU score to the number of reference translations. In the French-English task, we have only one reference per sentence to be translated, while in the Chinese-English task we have four references. In Figure 5.3 we measure BLEU and WER metrics as functions of PEP when one reference is considered in measuring performance. We see that BLEU decreases as the PEP increases to allow target (French/Chinese) phrases to be deleted in translation. As in bitext word alignment, there is a critical value of PEP above which BLEU and WER quickly degrade. We note that performance under WER does improve slightly with PEP, unlike BLEU.

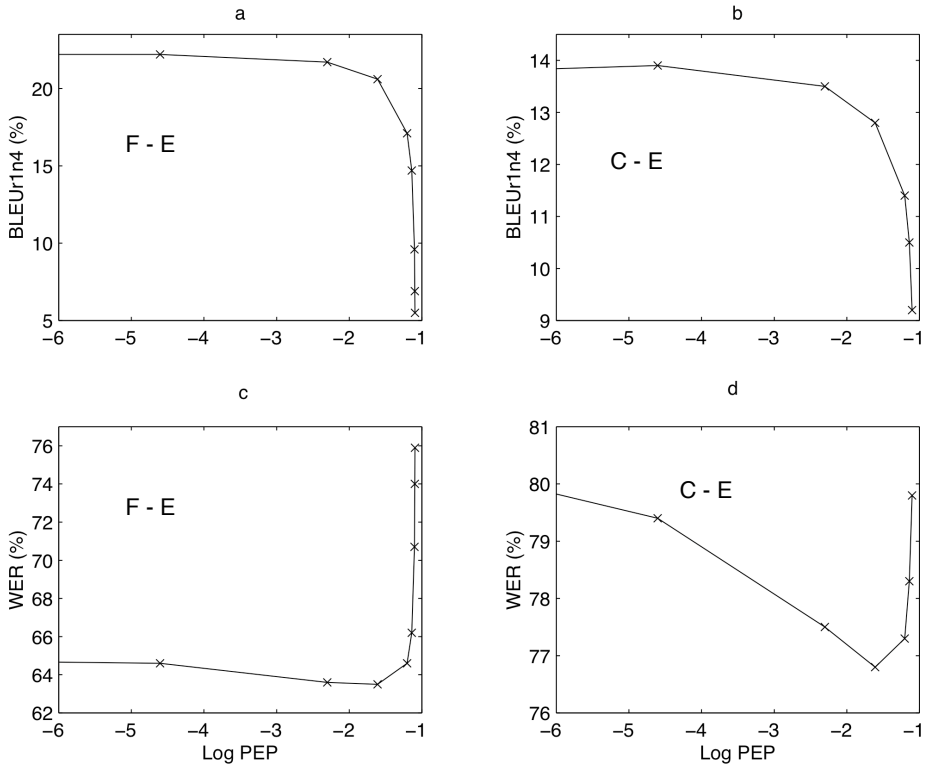

Figure 5.3: Translation Performance of the TTM as a function of the Phrase Exclusion Probability (PEP) when one reference translation is considered. We measure BLEU and WER on the French-English (Panel a,c) and the Chinese-English Tasks (Panel b,d).

We next discuss how PEP influences BLEU. Since BLEU is influenced by both BPrecision (Equation 5.3) and Brevity Penalty (Equation 5.4), we plot these components separately in Figure 5.4. We note first that as PEP increases, the translations grow shorter. This is measured by the Source-to-Target Length Ratio (STLRatio) (Figure 5.4d) which is the ratio of the number of words in the translation to number of words in the French sentence. This behavior is consistent with the role of PEP; it allows target phrases to delete in translation. The Brevity Penalty (Figure 5.4c) is governed by the number of words in the translation hypothesis, and therefore closely tracks the STLRatio. Somewhat surprisingly, BLEU score (Figure 5.4a) closely tracks the Brevity Penalty and does not improve despite improvements in BPrecision. Analogous to the case of bitext word alignment, increasing PEP allows the model to produce

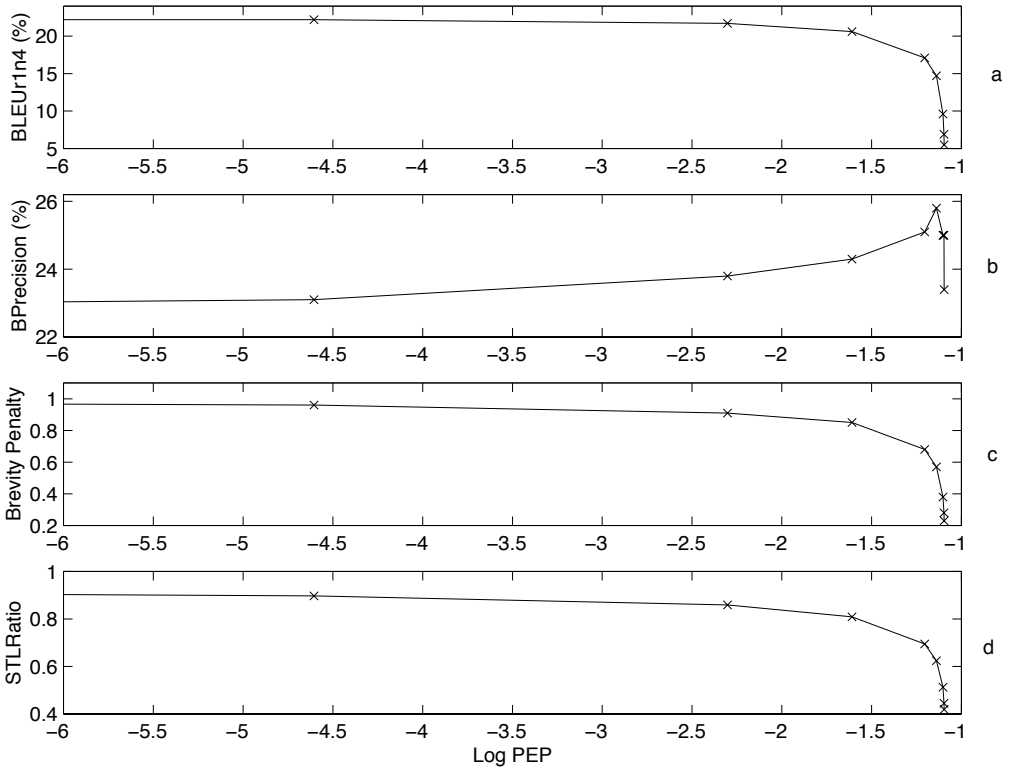

Figure 5.4: Translation Performance of the TTM as a function of the Phrase Exclusion Probability (PEP) on the French-English task. We measure BLEU (Panel a), BPrecision (Panel b), Brevity Penalty (Panel c), and Source-to-Target Length Ratio (Panel d) as functions of PEP.

higher quality translation when BPrecision (Figure 5.4b) is taken alone. However, the interaction between BPrecision and Brevity Penalty is such that the shorter sentences, although of higher precision, incur a very high Brevity Penalty so that the increase in precision does not improve BLEU overall.

The behavior of BPrecision is interesting in itself. Intuitively, it should be possible to increase the PEP so that only the most likely phrase translations are retained and thus improve the BPrecision. However we note in Figure 5.4b that BPrecision itself falls off above a critical value of PEP.

To explain this behavior of BPrecision, we study the contribution to the BLEU precision of the four n-gram precision measures (Equation 5.5) in the French-English task (Figure 5.5). In the TTM, the dominant mechanism by which shorter trans-

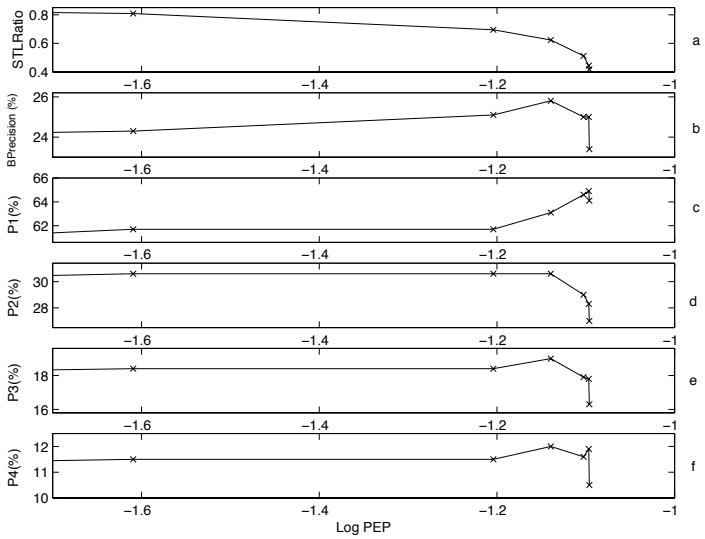

Figure 5.5: Analysis of BLEU Precision for values of Phrase Exclusion Probability (PEP) close to its maximum permissible value. We measure the following as functions of PEP : STLRatio (Panel a), BPrecision (Panel b) and each of the n-gram precisions, $n = 1, 2, 3, 4$ (Panels c-f). Results are shown on the French-English task.

lations are produced is to insert French phrases so that fewer French phrases are translated. As a result, English phrases in the translation arise from French phrases which are likely to be separated in the French sentence. It is correspondingly unlikely that English phrases in the translation (generated by separated French phrases) would follow each other in a fluent translation. Therefore the hypothesis translation contains phrases that are unlikely to be found next to each other in the reference translation. Consequently when precision statistics (Equation 5.5) are gathered over the translation, the hypothesized n-grams spanning these phrase boundaries are unlikely to be present in the reference translation, thus reducing precision. Figure 5.5 shows this behavior, the precision of higher n-grams ($n > 1$) falls off as the translations get shorter. Because of the need to account for n-grams spanning phrase boundaries, it is not possible to 'game' precision by merely producing shorter translations.

We now discuss the translation performance when multiple reference translations are considered for measuring translation performance (Figure 5.6). The most notable difference between the four-reference and one-reference scenarios is that BLEU score

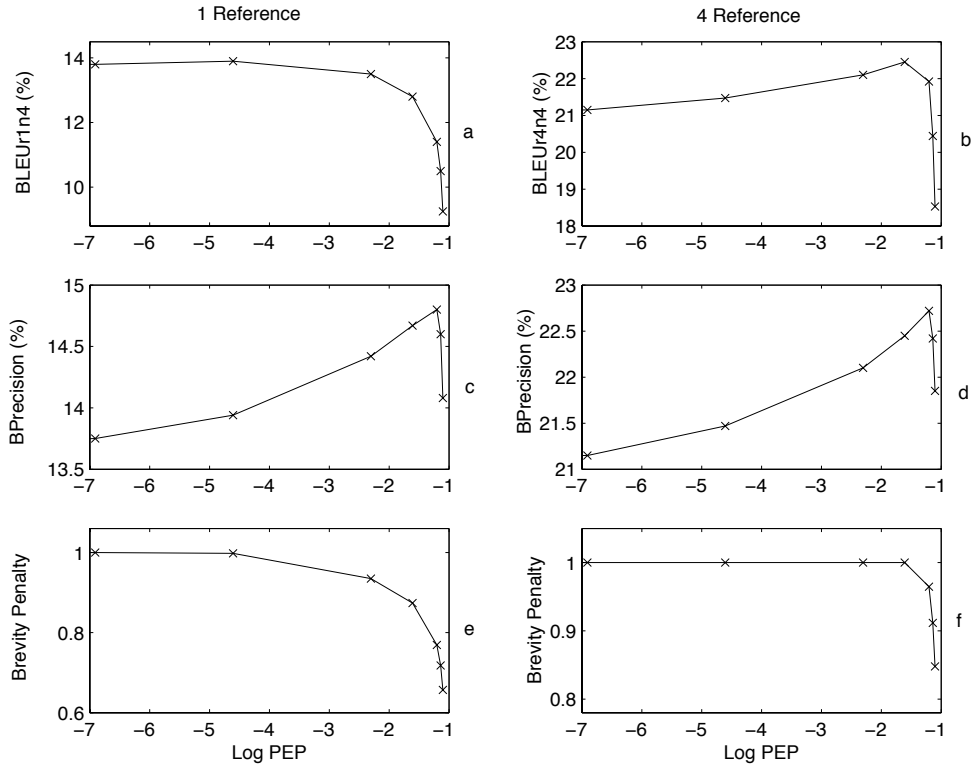

Figure 5.6: Translation Performance of TTM as a function of the Phrase Exclusion Probability when Multiple Reference Translations are considered for scoring. We obtain BLEU, BPrecision, and Brevity Penalty as functions of PEP in two situations: when 1 reference is considered (Panels a,c,e), and when 4 references are considered (Panels b,d,f). Results are shown on the Chinese-English task.

actually shows a substantial increase as PEP varies when four references are available. Relative to the single reference case, over the range of PEP, BPrecision also shows a greater increase and the Brevity Penalty is less severe.

We can explain this behavior by noting that Brevity Penalty is less severe when multiple reference translations are available in scoring (Figure 5.6e,f). In the single reference and multiple reference cases, the BPrecision increases with PEP, although the absolute values of BPrecision in the multiple reference case is higher due to the greater diversity of n-grams in the references. However in the multi-reference case, there is a greater range of PEP values over which the Brevity Penalty has little influence on the overall BLEU score. Within this range, the BPrecision can be

improved substantially by varying PEP so that BLEU shows a strong maximum.

Conclusion The variation of BLEU score with Phrase Exclusion Probability (PEP) is sensitive to the number of reference translations. When one reference is used in measuring performance, BLEU decreases with PEP (Figure 5.4a). Brevity Penalty increases with PEP due to a decrease in the length of the translations (Figure 5.4c,d). BLEU Precision increases at intermediate values of PEP because the model is able to produce higher quality translations (Figure 5.4b). However, beyond a critical value of PEP, BLEU Precision falls off due to a sharp decrease in the precision of the higher order n-grams (Figure 5.5d,e,f). The overall BLEU score is dominated by the Brevity Penalty and therefore decreases with PEP. When multiple reference translations are considered for scoring, BLEU shows a substantial increase as PEP is varied (Figure 5.6). This is due to diversity found in the multiple reference translations that, in turn, leads to a higher value of BLEU Precision and a less severe Brevity Penalty over the range of PEP values (Figure 5.6b,d,f).

5.2.2 Richness of the Phrase-Pair Inventory

We have described how the richness of the phrase-pair inventory can influence word alignment under the TTM (Section 4.3.2). We now investigate whether translation performance of the TTM might vary similarly. We use the four phrase-pair inventories described in Table 4.5 that are extracted from the same set of underlying word alignments but constructed from different amounts of bitext. Using each inventory, we construct a TTM system and measure its translation performance (under BLEU, NIST, and WER metrics) (Figure 5.7). In this experiment we measure performance at the optimal value of the PEP that was determined previously (Section 5.2.1). As the bitext size employed to construct the inventory increases, we observe an improvement in performance as measured with respect to all three translation metrics. This shows that if the underlying word alignment quality does not change, additional data helps to improve coverage of the test set by the inventory, and therefore improves the translation performance.

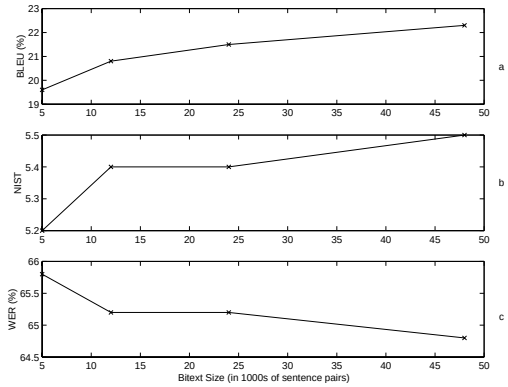

Figure 5.7: Effect of phrase-pair inventory size on translation performance of TTM. IBM-4 models are trained on 48K sentence-pairs from French-English Hansards, and word alignments are obtained over the collection. Four subsets are constructed from this set of word alignments and phrase-pair inventories were collected over each subset. For each inventory, translations under the TTM are obtained, and BLEU (Panel a), NIST (Panel b), and WER (Panel c) are plotted as functions of the bitext size employed to construct the inventory. Inventories are shown in Table 4.5.

Conclusion If we hold word alignment quality of IBM-4 translation models constant when constructing phrase-phrase inventories over different sizes of bitext (Table 4.5), increasing the size of the bitext improves coverage of target phrases on the test set (Table 4.5), and in turn improves translation performance of the TTM (Figures 5.7).

5.2.3 Word Alignment Quality of Underlying IBM-4 Models

We have studied how word alignment performance of the TTM varies with the quality of its underlying IBM-4 models (Section 4.3.3). We now study translation performance of the TTM in a similar way. We use the four phrase-pair inventories described in Table 4.6; these are constructed from the same bitext but their underlying word alignments arise from IBM-4 models trained on varying amounts of bitext. Using each inventory we construct a TTM system and measure its translation performance (under BLEU, NIST, and WER metrics) as a function of the bitext training set size in Figure 5.8, where we fix the PEP to its optimal value found in Section 5.2.1. We

observe that as the training set is increased, the AER of the IBM-4 models decreases, and the translation performance improves under all three translation metrics. We conclude that more bitext improves quality of the underlying word alignments and in turn improves translation quality.

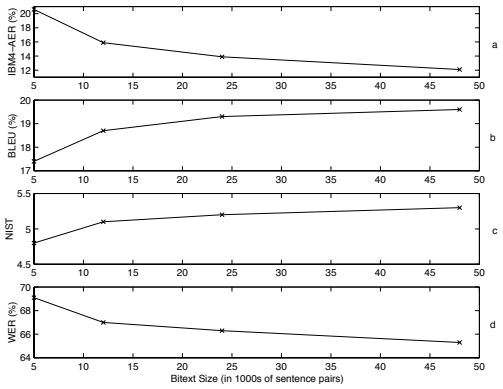

Figure 5.8: Translation performance of TTM as a function of the bitext size employed in training the underlying IBM-4 models. IBM-4 models are trained on four nested subsets of the French-English Hansards bitext, and word alignments are obtained over the smallest subset (5K sentence pairs). A phrase-pair inventory is constructed over each word alignment. For each inventory, translations under the TTM are obtained, and BLEU (Panel b), NIST (Panel c), and WER (Panel d) are plotted as functions of the bitext size employed in training the underlying IBM-4 models. We also measure AER of the underlying IBM-4 models (Panel a). Inventories are shown in Table 4.6.

Conclusion We investigate construction of phrase-pair inventories from IBM-4 word alignments of varying quality while keeping coverage of target phrases on a test set nearly constant (Table 4.6). In this case, increasing the size of the bitext for training IBM-4 translation models improves the word alignment quality of the models (Table 4.5), and consequently improves the translation performance of the TTM (Figure 5.8).

5.2.4 Lattice Quality

The goal of this experiment is to study the usefulness of translation lattices for rescoring purposes. For this purpose we generate N-best lists of translation hypotheses

from each translation lattice, and show the variation of their oracle-best BLEU scores with the size of the N-best list (Figure 5.9). The oracle-best BLEU score is obtained in the following way. For each sentence in the test set, we obtain the oracle-best hypothesis by selecting the translation from N-best list with the highest sentence-level BLEU score relative to the reference translation. We concatenate these oracle-best hypotheses over all sentences in the test set and then measure the test-set BLEU score of the resulting hypothesis.

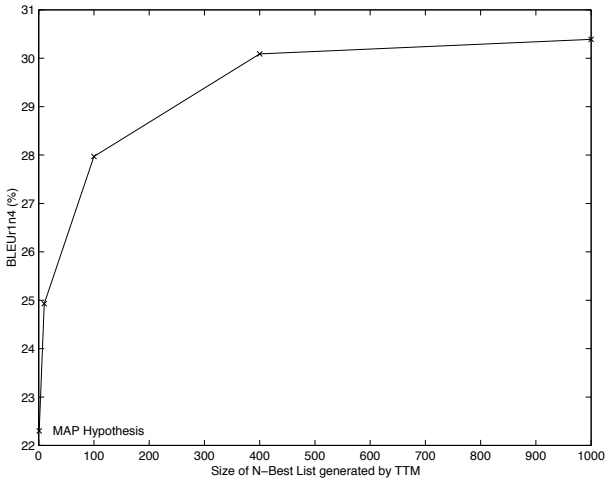

Figure 5.9: Variation of oracle-best BLEU scores with the size of the N-best list on the French-English Task. For each N-best list on the test set, the oracle BLEU hypothesis is computed under the sentence-level BLEU metric. The oracle hypotheses are concatenated over the test set, and the test-set BLEU score is measured.

We observe that the oracle-best BLEU score sharply increases with the size of the N-Best List.

Conclusion Oracle-best BLEU scores over an N-best list of translations can be obtained by measuring the BLEU score of the hypothesis in the list that is closest to the reference translation(s). Oracle-best BLEU scores improve substantially with the size of the N-best list generated by the TTM. We can therefore expect to rescore the lattices and N-best lists generated by TTM with more sophisticated models and achieve improvements in translation quality.

5.2.5 Translation Examples

We now present examples of translations generated by the TTM at different levels of translation performance as measured by the BLEU score. We provide these examples to illustrate good translations and poor translations under the BLEU metric in an attempt to provide the reader some intuition of this metric. The examples are selected from the NIST 2002 Chinese-English evaluation set. Table 5.2 shows five examples each of translations with sentence-level BLEU scores in the following intervals: $60-70\%$, $40-50\%$, $20-30\%$, and $0-10\%$.

We observe from this sample that translations with BLEU scores of 40% or higher are generally readable while those with BLEU scores lower than 10% appear poor and ill-formed.

5.3 Translation Performance with Large Bitext Training Sets

We report the performance of the Translation Template Model on the Chinese-to-English translation tasks in the NIST 2004 MT evaluation [78]. We will describe the training and test data, model training procedures, and the experiments performed in the development of this evaluation system.

5.3.1 Data

The goal of the NIST 2004 Chinese-English task [78] is the translation of news stories, editorials and speeches from Chinese to English. The large data track in this task restricts the allowable bitext to that provided by the LDC but places no restrictions on the monolingual English text used by the systems. We first present the training and test data for this task. We then describe the steps involved in preparing this text for translation model and language model training.

Translations	BLEU (%)
	$60-70\%$
Afghan Earthquake Victims begin to rebuild their homes .	66.1
Prior to this , the ANC has issued a statement calling for the international community to respect the choice of the people and help them survive .	66.0
Statistics show that since 1992 , a total of 204 UN personnel have been killed , but only 15 criminals have been arrested .	64.4
Chavez emphasized that Venezuela needs peace , stability and reason for all parties should make joint efforts to end the conflict .	62.2
London Financial Times Index Friday at closing newspaper 5,292.70 points , up 31.30 points .	61.0
	$40-50\%$
Indonesia Reiterates Opposition to the presence of foreign troops in	49.0
Witnesses said that Zambia Ji lying in a pool of blood , aside from his briefcase .	48.4
Hong Kong Police Narcotics Bureau pointed out that this is the first discovery , should attach great importance to this issue .	46.4
Three pieces of wall sports in October last year were shipped to New York to mark the 1990 reunification of Germany .	42.5
Xiao Yang , president of the work report yesterday by 2026 votes to 528 votes against 259 abstentions .	41.4
	$20-30\%$
Japan to temporarily freeze asked Russia to provide humanitarian assistance ,	30.0
Opposition Senator held that the president should focus more on domestic affairs and not eager to go abroad .	26.2
Taiwan DPP Legislator Chen Kim de fisheries groups to visit to Beijing .	23.9
Recently , the international community for the recent conflict , the fiercest Jenin camp conflict investigation of spreading .	20.8
opinion maintained : Gusmao victory is a strong possibility because he is considered the East Timor independence hero .	20.0
	$0-10\%$
Japan Telecom company in 2000 to spend 5.5 billion dollars buy back .	0.0
However , the voting result shows that Zhu because there is no reason to be losing power by NPC deputies desolate .	0.0
77 private manufacturing enterprises also reported a foreign trade management right .	0.0
Identification Department found that college students of the certificate , many of them were fake .	0.0
The European Union would be implemented in steel imports temporary protective measures to discuss with the Chinese side ,	0.0
Georgia from a section of the great mountains Canyon withdrawal ,	0.0

Table 5.2: Examples of translations under the TTM at various levels of translation performance as measured by the sentence-level BLEU score. These examples are selected from the NIST 2002 MT evaluation set.

Source Language Texts and Bitexts Our translation model training data consists of Chinese-English parallel texts derived from the following seven sources: Foreign Broadcast Information Service (FBIS) [57], Hong Kong News (HKNews) [59], Xinhua News (Xinhua) [54], Hong Kong Hansards (Hansards) [58], Translations from the Chinese Treebank (CTB) [55], Sinorama Magazine (Sinorama) [53] and the United Nations (UN) [61].

Our language model training data consists of English text derived from the following four sources: English side of FBIS bitext (10.5M words), Xinhua news agency (Xinhua) (155.7M words), Agency France Presse (AFP) (200.8M words), and online archives (Sept 1998 to Feb 2002) of The People's Daily (PD) [18] (16.2M words). The Xinhua and AFP texts are obtained from the LDC Gigaword English corpus [56].

Test Sets We report translation performance on four test sets; these include the NIST 2001 [52], 2002 [60], 2003 and 2004 evaluation sets sets [78]. The test sets consist of 993, 878, 919 and 1788 sentences respectively. The NIST 2001, 2002 and 2003 form our development sets while NIST 2004 is the blind test set. There are four reference translations for each Chinese sentence in all four sets.

Text Processing Our automatic Chinese text processing consists of word segmentation (using the LDC word segmenter [51]) followed by grouping of numbers. The English text is processed using a simple tokenizer based on the text processing utility available in the the NIST MT evaluation toolkit [78]. Following the above normalization, singletons (words with a frequency of one) on the English and the Chinese sides of the training bitext are replaced with the same token. This is done primarily to reduce vocabulary sizes and therefore the memory requirements of IBM-4 model training. The final vocabulary sizes are $169,561$ words in English, and $233,183$ words in Chinese.

Bitext Chunking The original bitext from all seven sources is aligned at the document level. For some of these sources (FBIS, Xinhua, HKNews and HKHansards), we have the document pairs available from LDC. These documents are aligned au-

Bitext Source	Chunk Pairs (K)	Words	
		English (M)	Chinese (M)
FBIS	368.2	10.5	7.8
HKNews	615.9	16.3	15.2
Xinhua	137.1	3.9	3.7
Hansards	1426.8	35.3	30.8
CTB	4.7	0.1	0.1
Sinorama	138.4	3.7	3.3
UN	4936.6	137.7	114.8
Total	7627. 7	207.4	175.7

Table 5.3: Statistics computed over chunk-pairs extracted from bitext sources in the NIST Chinese-English 2004 MT task.

tomatically into chunk-pairs under a statistical chunk alignment model [20]. For the other bitext sources (UN, Sinorama and CTB), the original document pairs are not available; we therefore use the sentence alignments provided by the LDC. From the LDC sentence alignments, we retain sentence pairs that are such that : 1) both English and Chinese sentences are shorter than 60 words and 2) ratio of the number of words in the English sentence to the number of words in the Chinese sentence is less than 6. The rest of the sentence-pairs are realigned at the sub-sentence level to obtain shorter chunk-pairs. Statistics computed over the chunk pairs from all bitext sources are shown in Table 5.3.

5.3.2 Model Training

We now describe the procedures involved in training the translation model and the language model.

IBM-4 Translation Model Training We partition the available bitext into three parts and train IBM-4 translation models (IBM4-F, IBM4-E) on each partition. We then use each IBM-4 model to obtain word alignments (IBM4-$E \cup F$) over the corresponding partition. The contribution by bitext source (in English words) in each of the three partitions is given in Table 5.4. For each partition we also report the Alignment Error Rate (AER) of the corresponding IBM-4 model. The AER is com-

Partition ID	Contribution by Source: En words (M)					AER (%)	
	FBIS	HKNews	H.C.	UN	Total	IBM4-E	IBM-F
1	10.5	16.3	0	26.7	53.5	36.5	34.0
2	10.5	0	0	85.0	95.5	36.9	32.9
3	10.5	16.3	43.0	25.8	95.6	36.5	32.4

Table 5.4: Composition by Bitext Source over 3 partitions of the Chinese-English bitext training set. For each partition we also report the Alignment Error Rate of the IBM-4 models on a 124-sentence subset of Eval01 test set. H.C. refers to a heterogeneous collection of bitext sources (Xinhua+Hansards+CTB+Sinorama).

puted over a 124-sentence subset of NIST eval01 test set [52] for which manual word alignments are available.

Phrase-Pair Extraction Following IBM-4 model training, the IBM-4 word alignments from the three training set partitions are merged and phrase-pairs are extracted from the resulting word alignments (using the procedure described in Section 3.3). For reducing storage requirements of the phrase-pair inventory, we extract only those phrase-pairs whose Chinese side is seen in the test set. Table 5.5 reports some statistics over the phrase-pair inventories restricted to the eval01, eval02, eval03 and eval04 test sets.

Statistics	Test Sets			
	eval01	eval02	eval03	eval04
# of Phrase-Pairs (K)	1438.6	1305.9	1318.3	2543.8
# of English Phrases (K)	142.6	120.9	118.0	242.4
# of Chinese Phrases (K)	34.5	26.0	25.4	54.6

Table 5.5: Statistics computed over TTM Phrase-Pair inventories restricted to the NIST 2001, 2002, 2003 and 2004 Chinese-English MT test sets.

.

English Language Models We build three language models from the source language (English) texts using modified Kneser-Ney smoothing as implemented in the SRILM tookit [90]. A small trigram model (Small 3g) is trained on People's Daily (16.2 M words) and 4.3 M words of Xinhua English news. A larger trigram model

(Large 3g) is constructed by first building a separate trigram model from each of the 4 text sources (Xin, AFP, FBIS and PD), and then interpolating the four language models using a weight of 0.25. A four-gram model (Large 4g) is similarly constructed by first building a 4-gram model from each of the 4 sources, and then interpolating the four LMs using a weight of 0.25.

5.3.3 Performance of Evaluation systems

We now report the performance of the TTM systems at each stage of the evaluation (Table 5.6). For comparison we will also report the performance of the four best competing MT systems that were fielded by other industrial and academic researchers in the 2004 evaluation (Table 5.7).

We first construct a TTM system using the phrase-inventory (described in Table 5.5), and measure its translation performance on the four test sets. Translation performance is measured using the case insensitive BLEU score on NIST 2001, 2002 and 2003 development sets, and using the case sensitive BLEU score on the NIST 2004 test set. For case-sensitive evaluation, we need to restore case information in our translations; this is done using a capitalizer (built in the JHU Summer Workshop WS'03 [81]) that uses a trigram language model trained on case-preserved English texts from FBIS, PD and Xinhua [81].

In Table 5.6 we report the performance of the TTM system under each of the three language models (Small 3g, Large 3g and Large 4g). For performing translation under either of the two trigram language models, we first generate a translation lattice using a pruned version of the language model and then rescore the lattice using the unpruned language model. For performing translation under the four gram language model, we first generate a translate lattice under the large trigram LM (Large 3g), and then rescore this lattice with the four gram language model. As a final stage of the evaluation system, we perform Minimum Bayes-Risk rescoring under the BLEU loss function (to be discussed in Chapter 7) on N-best lists generated with the four-gram language model.

We observe that the large trigram LM outperforms the smaller trigram LM by

Stage	BLEU (%)			
	eval01	eval02	eval03	eval04
Small 3g	29.5	27.0	25.7	-
Large 3g	30.2	28.0	27.2	26.5
Large 4g	31.2	28.8	27.5	27.6
MBR-BLEU	31.4	29.0	27.7	27.8

Table 5.6: Performance of the Chinese-to-English system at various evaluation stages on the NIST 2004 MT task. We report the performance of the TTM system under three different language models, and the performance of MBR decoders over N-best lists generated under the 4-gram LM.

System	BLEU% (eval04)
JHU-TTM	27.8
Competing MT systems	
1	32.1
2	27.2
3	22.7
4	21.9

Table 5.7: Performance of the JHU-TTM system relative to the 4 best competing MT systems that were fielded by other industrial and academic researchers in the NIST 2004 evaluation.

about 1.0% in BLEU. The four gram LM yields a further improvement of about 1.0% BLEU over the bigger trigram LM. Finally MBR-BLEU decoding gives an improvement of about 0.2% BLEU over the four gram LM.

5.3.4 Summary of Evaluation systems

We have described the use of TTM in building a Chinese-English MT system from large bitexts. The respectable performance of the TTM on the NIST 2004 task shows that our approach is competitive relative to contemporary research MT systems. Our MT system has benefitted considerably from the investigative experiments done to study the contribution of model components to the overall system performance (Sections 4.3 and 5.2). The WFST-TTM architecture supports the generation and rescoring of translation lattices and N-best lists. We have found this to be valuable in performing rescoring under various language models as well under Minimum Bayes-Risk decoding procedures.

5.4 Discussion

We have discussed translation under the TTM. We have shown how translation under the TTM can be performed using standard Weighted Finite State Transducer operations involving the TTM component transducers. The WFST operations also allow generation of translation lattices without any extra effort in implementation.

The translation performance of the TTM compares favorably to that of the ReWrite decoder that employs the same set of IBM-4 translation models. We find that Phrase Exclusion Probability (PEP) influences translation performance of TTM. As PEP is increased, we observe that BLEU degrades but WER improves slightly at first before degrading. Examining this behavior shows that as PEP is increased, the translations become shorter and the Brevity Penalty increases while the BPrecision increases. However, the interaction between BPrecision and Brevity Penalty is such that the shorter sentences, although of higher precision, incur a very high Brevity Penalty so that the increase in precision does not improve BLEU overall. Furthermore, BPreci-

sion itself falls off above a critical value of PEP; therefore it is not possible to 'game' BLEU Precision by merely producing shorter translations.

Interestingly, we find that the variation in BLEU score is sensitive to the number of reference translations used for scoring. The most notable difference between the four-reference and one-reference scenarios is that BLEU score actually shows a substantial increase as PEP varies when four references are available. We can attribute this difference to the greater diversity in the reference translations with respect to n-grams and length; therefore multiple reference translations are more permissive of variations in the hypothesized translation length.

The quality of underlying word alignments and richness of the phrase-pair inventory influence translation performance of the TTM. When the alignment quality of underlying IBM-4 models is fixed, additional data helps to improve coverage of the test set by the inventory, and therefore improves the translation performance. On the other hand, we can fix the bitext collection from which the phrase-pair inventory is gathered and vary the amount of bitext for training the underlying IBM-4 models; we find that this improves word alignment quality of the underlying models and in turn improves translation quality of the TTM.

Finally we study the variation of oracle-best BLEU scores on N-best lists generated by the TTM. We observe that the oracle-best BLEU score sharply increases with the size of the N-Best List; this shows that we can expect to rescore the translation lattices with more sophisticated models and achieve improvements in translation quality. This concludes the overview of the TTM investigative experiments.

We have described the construction of a large vocabulary Chinese-English TTM system for the NIST 2004 MT evaluations. Our results demonstrate that the TTM can be successfully scaled up to large training bitexts. The respectable performance of TTM on this task show that this framework is competitive relative to contemporary research MT systems.

The Translation Template Model is a very promising modeling framework for statistical machine translation. The model offers a simple and unified framework for bitext word alignment and translation. The simplicity of the model has allowed us to perform a detailed investigation of the several factors that influence the alignment

and translation performance of the model; we believe that this analysis has improved our understanding of the strengths and weaknesses of the model.

Chapter 6

Minimum Bayes-Risk Word Alignments of Bitexts

In this chapter we describe the application of Minimum Bayes-Risk (MBR) techniques to bitext word alignment [45]. We will show how specialized MBR decoders can be constructed to optimize performance under each alignment quality metric.

We first discuss loss functions for comparing word alignments (Section 6.1). We show that alignment loss functions can be derived from standard error metrics such as Alignment Precision, Alignment Recall and Alignment Error Rate. We also extend the Alignment Error loss function to incorporate linguistic features from parse-trees and part-of-speech tags. We next present the formulation of MBR decoders for bitext word alignment (Section 6.2). We demonstrate closed-form solutions to MBR decoders on alignment lattices under two classes of alignment loss functions.

In Section 6.3 we present the word alignment experiments. We first describe a procedure to generate alignment lattices under the IBM-3 translation model [10]. We then report the performance of MBR decoders on alignment lattices generated by the IBM-3 model and the Transducer Template Model respectively. We finally discuss experiments in Section 6.4.

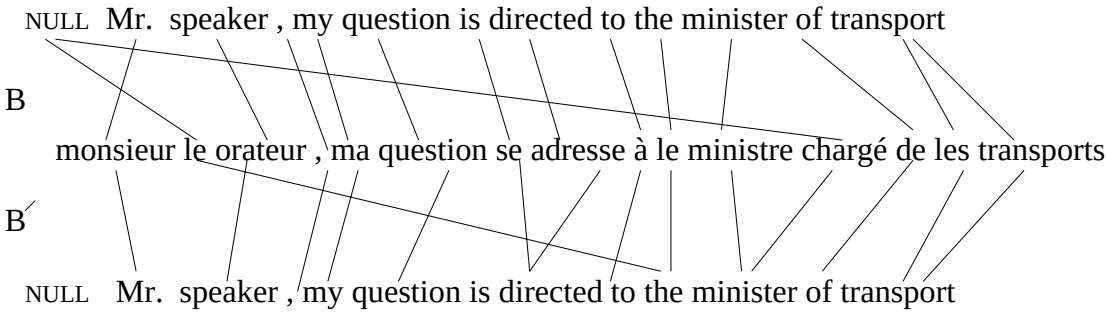

Figure 6.1: An example of two competing word alignments for an English-French sentence pair.

6.1 Alignment Loss Functions

We now introduce alignment loss functions to measure the quality of automatically produced alignments. In doing so, we will use the word alignment definitions introduced in Section 4.1.

We study the problem of aligning a pair of translated source and target sentences $E = e_0^I$ and $F = f_0^J$. For this sentence-pair, we wish to compare an automatically generated alignment B' to a reference alignment B, which we assume was produced by an automatic translator. Figure 6.1 shows an example of two such competing word alignments for an English-French sentence pair.

We will now define various loss functions $L(B, B')$ to measure the quality of B' with respect to B. These loss functions fall into two general classes. The first class of loss functions is derived from standard alignment metrics such as Precision, Recall, and Alignment Error. The second class of loss functions extends the Alignment Error to incorporate features from parse-trees, part-of-speech tags and word classes.

6.1.1 Precision, Recall and Alignment Error

Och and Ney [82] introduced three metrics to measure quality of automatically produced alignments. These are Alignment Precision, Alignment Recall, and Alignment Error Rate, and were defined in Section 4.3. For the reader's convenience, we repeat these definitions here.

We note that in these measurements, links to the NULL word are ignored. This is done by defining modified link sets for the reference alignment: $\bar{B} = B - \{(i,j) : i = 0 \vee j = 0\}$ and the automatic alignment: $\bar{B}' = B' - \{(i',j') : i' = 0 \vee j' = 0\}$. The reference annotation procedure allows the human annotators to identify which links in $\bar{B}$ they judge to be unambiguous. In addition to the reference alignment, this gives a set of *sure links* (S) which is a subset of $\bar{B}$. The ambiguous links ($\bar{B} \setminus S$) in the reference alignment are used especially to align words within idiomatic expressions, free translations, and missing function words [82]. The alignment metrics are defined as follows:

$$\begin{aligned}
\text{Precision } (S, B; B') &= \frac{|\bar{B}' \cap \bar{B}|}{|\bar{B}'|} \\
\text{Recall } (S, B; B') &= \frac{|\bar{B}' \cap S|}{|S|} \\
\text{AER } (S, B; B') &= 1 - \frac{|\bar{B}' \cap S| + |\bar{B}' \cap \bar{B}|}{|\bar{B}'| + |S|}.
\end{aligned}$$

Since our modeling techniques require loss functions rather than error rates or accuracies, we introduce the Precision Error, Recall Error and Alignment Error loss functions:

$$L_{PE}(B, B') = |\bar{B}'| - |\bar{B} \cap \bar{B}'| \tag{6.1}$$

$$L_{RE}(B, B') = |\bar{B}| - |\bar{B} \cap \bar{B}'| \tag{6.2}$$

$$\begin{aligned}
L_{AE}(B, B') &= |\bar{B}| + |\bar{B}'| - 2|\bar{B} \cap \bar{B}'| \\
&= |\bar{B}| + |\bar{B}'| - 2 \sum_{b \in \bar{B}} \sum_{b' \in \bar{B}'} \delta_b(b').
\end{aligned} \tag{6.3}$$

We consider error rates to be "normalized" loss functions. We note that, unlike AER, Precision, or Recall, the loss functions L_{AE}, L_{PE} and L_{RE} do not distinguish between ambiguous and unambiguous links. However, if a decoder generates an alignment B' for which $L_{AE}(B, B')$ is zero, the AER is also zero. Therefore if AER is the metric of interest, we will design alignment procedures to minimize L_{AE}. Similarly, if Alignment Precision (Alignment Recall) is the metric of interest, we develop alignment procedures to minimize L_{PE} (L_{RE}).

6.1.2 Generalized Alignment Error

We are interested in extending the Alignment Error loss function (Equation 6.3) to incorporate various linguistic features into the measurement of alignment quality. The *Generalized Alignment Error* is defined as

$$L_{GAE}(B, B') = 2 \sum_{b \in B} \sum_{b' \in B'} \delta_i(i') d_{ijj'}. \tag{6.4}$$

where $b = (i, j)$, $b' = (i', j')$ and

$$d_{ijj'} = D((j, e_j), (j', e_{j'}); f_i). \tag{6.5}$$

Here we have introduced the word-to-word distance measure $D((j, e_j), (j', e_{j'}); f_i)$ which compares the links (i, j) and (i, j') as a function of the words in the translation. L_{GAE} refers to all loss functions that have the form of Equation 6.4. Specific loss functions are determined through the choice of D. To see the value in this, suppose f_i is a verb in the French sentence and that it is aligned in the reference alignment to e_j, the verb in the English sentence. If our goal is to ensure verb alignment, then D can be constructed to penalize any link $(e_{j'}, f_i)$ in the automatic alignment in which $e_{j'}$ is not a verb.

We will now give examples of distances in which L_{GAE} is based on part-of-speech (POS) tags, parse tree distances, and automatically determined word clusters. We note that the L_{GAE} can almost be reduced to L_{AE}, except for the treatment of NULL in the source (English) sentence.

Syntactic Distances From Parse-Trees Suppose a parser is available that generates a parse-tree for the English sentence. Our goal is to construct an alignment loss function that incorporates features from the parse. One way to do this is to define a graph distance

$$d_{ijj'} = g(N_{e_j}, N_{e_{j'}}). \tag{6.6}$$

Here N_{e_j} and $N_{e_{j'}}$ are the parse-tree leaf nodes corresponding to the English words e_j and $e_{j'}$. This quantity is computed as the sum of the distances from each node to their closest common ancestor. It gives a syntactic distance between any pair of

English words based on the parse-tree. This distance has been used to measure word association for information retrieval [68]. It reflects how strongly the words e_j and $e_{j'}$ are bound together by the syntactic structure of the English sentence as determined by the parser. Figure 6.2 shows the parse tree for an English sentence in the test data with the pairwise syntactic distances between the English words corresponding to the leaf nodes.

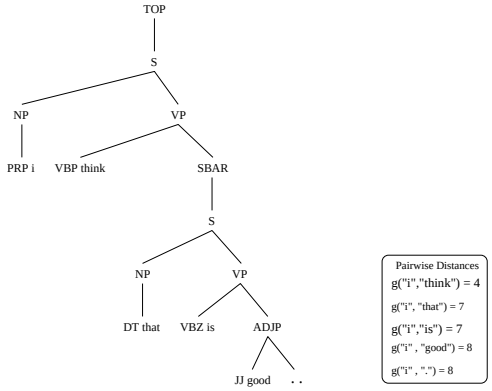

Figure 6.2: Parse tree for an English sentence with the pairwise syntactic distances between words.

With D defined as in Equation 6.6, the Generalized Alignment Error loss function (Equation 6.4) is called the Parse-Tree Syntactic Distance (L_{PTSD}).

Distances Derived From Part-of-Speech Labels Suppose a Part-of-Speech (POS) tagger is available to tag each word in the English sentence. If POS(e_j) denotes the POS of the English word e_j, we can define the word-to-word distance measure D (Equation 6.5) as

$$d_{ijj'} = \begin{cases} 0 & \text{POS}(e_j) = \text{POS}(e_{j'}) \\ 1 & otherwise. \end{cases} \tag{6.7}$$

With D specified by Equation 6.7, the Generalized Alignment Error loss function (Equation 6.4) is called the Part-of-Speech Distance (L_{POSD}).

Automatic Word Cluster Distances Suppose we are working in a language for which parsers and POS taggers are not available. In this situation we might wish to

construct the loss functions based on word classes determined by automatic clustering procedures. If $C(e_j)$ specifies the word cluster for the English word e_j, then we define the distance

$$d_{ijj'} = \begin{cases} 0 & C(e_j) = C(e_{j'}) \\ 1 & otherwise. \end{cases} \tag{6.8}$$

With D as defined in Equation 6.8, the Generalized Alignment Error loss function (Equation 6.4) is called the Automatic Word Class Distance (L_{AWCD}).

6.2 Minimum Bayes-Risk Decoding for Automatic Word Alignment

We here present the formulation of MBR decoders for word alignment, and derive MBR alignment procedures under the loss functions of Section 6.1.

Bitext word alignment can be described as a classification problem in which a pair of sentences (E, F) is mapped to a word alignment $B' = \delta(E, F)$. Given an alignment loss function L and the true distribution $P(E, F, B)$ over word alignments, we can measure the classifier performance under the risk $E_{P(B|F,E)}[L(B, \delta(E, F)]$. The classifier that minimizes this risk is the Minimum Bayes-Risk decision rule (Equation 1.1)

$$\hat{B} = \underset{B' \in \mathcal{B}}{\operatorname{argmin}} \sum_{B \in \mathcal{B}} L(B, B') P(B|F, E). \tag{6.9}$$

In the above formulation, we do not have access to the true distribution over translations. We therefore use statistical translation models to approximate $P(B|F, E)$.

$$P(B|F, E) = \frac{P(B, F|E)}{\sum_{B'} P(B', F|E)}, \tag{6.10}$$

where $P(B, F|E)$ is obtained using a statistical model. We also assume that the space of alignment alternatives can be restricted to an *alignment lattice* $\mathcal{B}$ (Section 4.1.3), which is a compact representation of the most likely word alignments of the sentence pair (E, F) under the statistical model.

We now describe the alignment lattice (Section 6.2.1) and introduce the lattice based probabilities required for the MBR alignment (Section 6.2.2). The derivation

of the MBR alignment under AE, PE, and RE loss functions is presented in Sections 6.2.3, and the derivation under GAE loss function is presented in Section 6.2.4.

6.2.1 Alignment Lattice

Section 4.1.3 introduced the alignment lattice and described a procedure to generate alignment lattices under the Translation Template Model. We will now describe the alignment lattice in greater detail.

The alignment lattice $\mathcal{B}$ is represented as a Weighted Finite State Transducer (WFST) [74] $\mathcal{B} = (Q, \Lambda, \kappa, F, \mathcal{T})$ with a finite set of states Q, a set of transition labels Λ, an initial state κ, the set of final states F, and a finite set of transitions $\mathcal{T}$. A transition in this WFST is given by $t = (p, q, b, s)$ where p is the starting state, q is the ending state, $b \in \Lambda$ is the alignment link and s is the weight. For an English sentence of length I and a French sentence of length J, we define Λ as $\Lambda = \{(i, j) : i \in \{0, 1, ..., J\}, j \in \{0, 1, ..., I\}\}$. We note that the TTM alignment lattice shown in Figure 4.4 can be converted to this format by replacing each phrase-pair in the lattice with its internal word alignments (from Table 4.4).

A *complete path* through the WFST is a sequence of transitions given by $T = \{(p_k, q_k, b_k, s_k)\}_{k=1}^{n}$ such that $p_1 = \kappa$ and $q_n \in F$. Each complete path defines an alignment link set $B = \{b_k\}_{k=1}^{n}$. When we write $B \in \mathcal{B}$, we mean that B is derived from a complete path through $\mathcal{B}$. This allows us to use alignment models in which the probability of an alignment can be written as a sum over alignment link weights, i.e. $logP(B, F|E) = \sum_{k=1}^{n} s_k$.

6.2.2 Alignment Link Posterior Probability

We first introduce the *lattice transition posterior probability* of each transition $t = (p, q, b, s)$ in the lattice

$$P(t|F, E) = \sum_{B \in \mathcal{B}} \psi_B(t) P(B|F, E) \tag{6.11}$$

where $\psi_B(t)$ is 1 if $b \in B$ and 0 otherwise. The lattice transition posterior probability is the sum of the posterior probabilities of all lattice paths passing through the tran-

sition t. This can be computed very efficiently with a forward-backward algorithm on the alignment lattice [104]. $P(B|F,E)$ is the posterior probability of an alignment link set which can be obtained as

$$P(B|F,E) = \frac{P(B,F|E)}{\sum_{B' \in \mathcal{B}} P(B',F|E)}. \tag{6.12}$$

We now define the *alignment link posterior probability* for a link $b = (i,j)$

$$P(b|F,E) = \sum_{t' \in \mathcal{T}} \delta_b(b') P(t'|F,E) \tag{6.13}$$

where $t' = (p', q', b', s')$. This is the probability that any two words (f_i, e_j) are aligned given all the alignments in the lattice $\mathcal{B}$.

6.2.3 MBR Alignment Under L_{AE}, L_{PE} and L_{RE}

In this section we derive MBR alignment under the Alignment Error, Precision Error and the Recall Error loss functions. We start with the Alignment Error (Equation 6.3). In this case the MBR decoder has the form (Equation 6.9)

$$\hat{B} = \underset{B' \in \mathcal{B}}{\operatorname{argmin}} \sum_{B \in \mathcal{B}} L_{AE}(B,B') P(B|F,E). \tag{6.14}$$

The summation is equal to

$$\begin{aligned} & |\bar{B}'| + \sum_{B \in \mathcal{B}} |\bar{B}| P(B|F,E) \\ & -2 \sum_{b' \in \bar{B}'} \Big\{ \sum_{B \in \mathcal{B}} \sum_{b \in \bar{B}} \delta_b(b') P(B|F,E) \Big\}. \end{aligned}$$

If $\bar{\mathcal{T}} \subseteq \mathcal{T}$ is the subset of transitions ($t = (p,q,b,s)$) that do not contain links with the NULL word, we can simplify the bracketed term as

$$\begin{aligned} & \sum_{B \in \mathcal{B}} \sum_{b \in \bar{B}} \delta_b(b') P(B|F,E) \\ = & \sum_{B \in \mathcal{B}} \sum_{t \in \bar{\mathcal{T}}} \psi_B(t) \delta_b(b') P(B|F,E) \\ = & \sum_{t \in \bar{\mathcal{T}}} \delta_b(b') \sum_{B \in \mathcal{B}} \psi_B(t) P(B|F,E) \\ = & \sum_{t \in \bar{\mathcal{T}}} \delta_b(b') P(t|F,E) \end{aligned}$$

For an alignment link $b' \in \bar{B}'$ we note that $\sum_{t \in \bar{\mathcal{T}}} \delta_b(b') P(t|F,E) = P(b'|F,E)$. Therefore, the MBR alignment (Equation 6.14) can be found in terms of the modified link weight for each alignment link $b' = (i', j')$

$$\hat{B} = \underset{B' \in \mathcal{B}}{\operatorname{argmin}} \sum_{b' \in \bar{B}'} \big(1 - 2P(b'|F,E)\big).$$

We can rewrite the above equation as

$$\begin{aligned} \hat{B} &= \operatorname{argmin}_{B' \in \mathcal{B}} \textstyle\sum_{b' \in B'} y_{b'} \\ y_{b'} &= \begin{cases} 1 - 2P(b'|F,E) & i' \neq 0 \wedge j' \neq 0 \\ 0 & i' = 0 \vee j' = 0. \end{cases} \end{aligned} \tag{6.15}$$

The MBR alignments under the Precision Error (Equation 6.1) and the Recall Error (Equation 6.2) loss functions can similarly be obtained as:

$$\begin{aligned} \hat{B}_{PE} &= \operatorname{argmin}_{B' \in \mathcal{B}} \textstyle\sum_{b' \in B'} y_{b'} \\ y_{b'} &= \begin{cases} 1 - P(b'|F,E) & i' \neq 0 \wedge j' \neq 0 \\ 0 & i' = 0 \vee j' = 0, \end{cases} \end{aligned} \tag{6.16}$$

and

$$\begin{aligned} \hat{B}_{RE} &= \operatorname{argmin}_{B' \in \mathcal{B}} \textstyle\sum_{b' \in B'} y_{b'} \\ y_{b'} &= \begin{cases} -P(b'|F,E) & i' \neq 0 \wedge j' \neq 0 \\ 0 & i' = 0 \vee j' = 0. \end{cases} \end{aligned} \tag{6.17}$$

6.2.4 MBR Alignment Under L_{GAE}

We here derive MBR alignment under the Generalized Alignment Error loss function (Equation 6.4). The optimal decoder has the form (Equation 6.9)

$$\hat{B} = \underset{B' \in \mathcal{B}}{\operatorname{argmin}} \sum_{B \in \mathcal{B}} L_{GAE}(B, B') P(B|F,E). \tag{6.18}$$

The summation can be rewritten as

$$\begin{aligned}
&\sum_{B\in\mathcal{B}} L_{GAE}(B,B')P(B|F,E) \\
&= \sum_{B\in\mathcal{B}} 2\sum_{b\in B}\sum_{b'\in B'} \delta_i(i')d_{ijj'}P(B|F,E) \\
&= 2\sum_{b'\in B}\Big\{\sum_{B\in\mathcal{B}}\sum_{b\in B} \delta_i(i')d_{ijj'}P(B|F,E)\Big\}
\end{aligned}$$

where $b=(i,j)$ and $b'=(i',j')$.
We can simplify the bracketed term as

$$\begin{aligned}
&\sum_{B\in\mathcal{B}}\sum_{b\in B} \delta_i(i')d_{ijj'}P(B|F,E) \\
&= \sum_{B\in\mathcal{B}}\sum_{t\in\mathcal{T}} \delta_i(i')d_{ijj'}\psi_B(t)P(B|F,E) \\
&= \sum_{t\in\mathcal{T}} \delta_i(i')d_{ijj'}\sum_{B\in\mathcal{B}}\psi_B(t)P(B|F,E) \\
&= \sum_{t\in\mathcal{T}} \delta_i(i')d_{ijj'}P(t|F,E)
\end{aligned}$$

where $t=(p,q,b,s)$ and $b=(i,j)$.

The MBR alignment (Equation 6.18) can be found in terms of the modified link weight for each alignment link b'

$$\begin{aligned}
\hat{B} &= \text{argmin}_{B'\in\mathcal{B}} \textstyle\sum_{b'\in B'} z_{b'} \\
z_{b'} &= \textstyle\sum_{t\in\mathcal{T}} \delta_i(i')d_{ijj'}P(t|F,E).
\end{aligned} \tag{6.19}$$

6.2.5 MBR Alignment Using WFST Techniques

The MBR alignment procedures under the L_{PE}, L_{RE}, L_{AE} and L_{GAE} loss functions begin with a WFST that contains the alignment probabilities $P(B,F|E)$ as described in Section 6.2.1. To build the MBR decoder for each loss function the weights on the transitions ($t'=(p',q',b',s')$) of the WFST are modified according to either Equations 6.15-6.17 ($s'=y_{b'}$) or Equation 6.19 ($s'=z_{b'}$). Once the weights are modified, the search procedure for the MBR alignment is the same in each case. The search is carried out using a $O(n^3)$ shortest-path algorithm [74, 71].

6.2.6 Computation of Oracle-best Alignments

The Oracle-best Alignment under an error metric is defined as the hypothesis in the alignment lattice with the lowest error relative to the reference word alignment B_r. We now describe procedures to compute the oracle-best alignment under the four loss functions presented in Section 6.1.

We first describe the computation of the oracle-best alignment under the L_{AE} loss function (Equation 6.3).

$$\begin{aligned}
\hat{B}_o^{AE} &= \operatorname*{argmin}_{B' \in \mathcal{B}} L_{AE}(B_r, B) \\
&= \operatorname*{argmin}_{B' \in \mathcal{B}} |\bar{B}'| + |\bar{B}_r| - 2|\bar{B}_r \cap \bar{B}'| \\
&= \operatorname*{argmin}_{B' \in \mathcal{B}} \sum_{b' \in \bar{B}'} \left\{ 1 - 2 \sum_{b \in \bar{B}_r} \delta_b(b') \right\} \\
&= \operatorname*{argmin}_{B' \in \mathcal{B}} \sum_{b' \in B'} y_{b'},
\end{aligned} \tag{6.20}$$

where the modified weight of an alignment link $b' = (i', j')$, for $b' \in B'$, is given by

$$y_{b'} = \begin{cases} 0 & i' = 0 \vee j' = 0 \\ -1 & i' \neq 0 \wedge j' \neq 0 \wedge b' \in B_r \\ 1 & i' \neq 0 \wedge j' \neq 0 \wedge b' \notin B_r. \end{cases} \tag{6.21}$$

The oracle-best alignment hypotheses under the L_{PE} (Equation 6.1), L_{RE} (Equation 6.2), and the L_{GAE} (Equation 6.4) loss functions can similarly be obtained as:

$$\hat{B}_o^{PE} = \operatorname*{argmin}_{B' \in \mathcal{B}} \sum_{b' \in B'} y_{b'} \tag{6.22}$$

$$y_{b'} = \begin{cases} 0 & i' = 0 \vee j' = 0 \\ 0 & i' \neq 0 \wedge j' \neq 0 \wedge b' \in B_r \\ 1 & i' \neq 0 \wedge j' \neq 0 \wedge b' \notin B_r, \end{cases}$$

$$\hat{B}_o^{RE} = \operatorname*{argmin}_{B' \in \mathcal{B}} \sum_{b' \in B'} y_{b'} \tag{6.23}$$

$$y_{b'} = \begin{cases} 0 & i = 0 \vee j' = 0 \\ -1 & i' \neq 0 \wedge j' \neq 0 \wedge b' \in B_r \\ 0 & i' \neq 0 \wedge j' \neq 0 \wedge b' \notin B_r, \end{cases}$$

and

$$\hat{B}_o^{GAE} = \operatorname*{argmin}_{B' \in \mathcal{B}} \sum_{b' \in B'} y_{b'} \tag{6.24}$$

$$y_{b'} = \sum_{b \in B_r} \delta_i(i') d_{ijj'}.$$

For computing the oracle-best hypothesis under Alignment Error, Precision Error, Recall Error, and the Generalized Alignment Error, we first modify the weights on the transitions ($t' = (p', q', b', s')$) of the WFST according to Equations 6.21-6.24 ($s' = y_{b'}$). Once the weights are modified, the search procedure for the oracle-best alignment is the same in each case. The search is performed using a shortest-path algorithm [74, 71].

6.3 Performance of MBR Word Alignments

We now report the performance of MBR word alignments. Since the true distribution over alignments is not known, we use statistical MT models to approximate $P(B|F, E)$. We also approximate the space of possible alignments with an alignment lattice (Section 6.2.1) generated by the statistical model.

In our MBR decoding experiments we consider two different statistical models for generating the alignment lattices. These include the IBM-3 translation model [10] and the Translation Template model described in Chapter 3. In each case, the alignment lattices generated by the model will be rescored by the MBR decoding procedures described in Sections 6.2.3 - 6.2.5. The generation of alignment lattices under the TTM was discussed in Section 4.1.3. We here describe how alignment lattices can be generated under the IBM-3 translation model.

6.3.1 Word Alignments under the IBM-3 translation model

As discussed in Section 3.1, IBM researchers proposed a series of five translation models based on word-for-word substitution. Each of these models defines a different method to compute the probability $P(B, f|e)$ for a word alignment B of the sentence pair (e, f). Our focus here is on generating word alignments under the IBM Model-3 [10]. The IBM-3 model (see Appendix A) is specified through four component models: Fertility probabilities for words, Fertility probabilities for the NULL word, Word Translation probabilities, and Distortion probabilities. We use a modified version of the IBM-3 distortion model [43] in which each of the permutations of the target language sentence is equally likely.

We obtain word alignments under the modified IBM-3 models using the WFST translation framework introduced by Knight and Al-Onaizan [43]. The finite state operations are carried out using the AT&T Finite State Machine Toolkit [72, 74].

The WFST framework involves building a transducer for each constituent of the IBM-3 translation model: the word fertility model M; the NULL fertility model N; and the word translation model T. For each sentence pair we also build a finite state acceptor E that accepts the English sentence and another acceptor F which accepts all legal permutations of the French sentence. The alignment lattice $\mathcal{B}$ for the sentence pair is then obtained by the following weighted finite state composition

$$\mathcal{B} = E \circ M \circ N \circ T \circ F.$$

In practice, the WFST obtained by the composition is pruned to a maximum of 10,000 states using a likelihood based pruning operation.

A heavily pruned IBM-3 alignment lattice $\mathcal{B}$ for a sentence-pair is shown in Figure 6.3. For clarity of presentation, each alignment link $b = (i, j)$ in the lattice is shown as an ordered pair $x\_j : y\_i$ where $x = e_j$ and $y = f_i$ are the English and French words on the link. For each sentence, we also computed the lattice path with the highest probability $P(B|f, e)$. This gives the Maximum Likelihood (ML) alignment under the IBM-3 model.

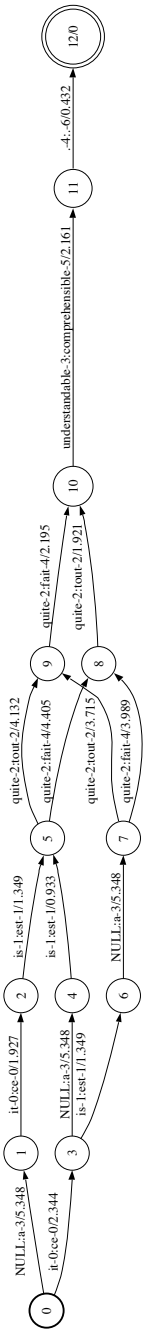

Figure 6.3: A heavily pruned IBM-3 alignment lattice for the English-French sentence pair: E=*it is quite understandable .* F=*ce est tout a fait comprehensible .* Each transition in this lattice has the format $x\_j : y\_i$ where $x = e_j$ and $y = f_i$; the link (i, j) indicates that English word e_j is aligned to the French word f_i.

6.3.2 MBR Alignments

In the previous sections we introduced a total of six loss functions: L_{PE}, L_{RE}, L_{AE}, L_{PTSD}, L_{POSD} and L_{AWCD}. Using either Equation 6.16, 6.17, 6.15, or 6.19, an MBR decoder can be constructed for each. These decoders are called MBR-PE, MBR-RE, MBR-AE, MBR-PTSD, MBR-POSD, and MBR-AWCD, respectively.

In our experiments, we use Collins parser [16] to obtain parse trees for the English side of the test corpus. Ratnaparkhi's POS tagger [88] is used to obtain POS tags for each word in the English sentence. We obtain word clusters for English words using a statistical learning procedure [42] where the total number of word classes is restricted to be 100.

6.3.3 Evaluation Metrics

The performance of the MBR decoders is measured with respect to the alignments provided by human experts [82]. The first set of evaluation metrics used are Alignment Precision, Alignment Recall and Alignment Error Rate. We also evaluate each decoder under the *Generalized Alignment Error Rates* (GAER).

$$GAER(B, B') = \frac{L_{GAE}(B, B')}{|B| + |B'|}. \tag{6.25}$$

There are six variants of *GAER*. These arise when L_{GAE} is specified by L_{PTSD}, L_{POSD} or L_{AWCD}. There are two versions of each of these: one version is sensitive only to sure (S) links. The other version considers all (A) links in the reference alignment. We therefore have the following six Generalized Alignment Error Rates: PTSD-S, POSD-S, AWCD-S, and PTSD-A, POSD-A, AWCD-A. We say we have a matched condition when the same loss function is used in both the error rate and the decoder design.

6.3.4 MBR decoders over IBM-3 lattices

We first report the performance of MBR decoders over alignment lattices generated by IBM-3 models. We train the IBM-3 models on 48,739 sentence-pairs from

the French-English Hansards (Section 4.2) using the GIZA++ toolkit. Our test data consists of 207 unseen French-English sentence pairs from the Hansards corpus. These form a subset of the 500 sentences for which reference word alignments are available (Section 4.2). These sentence pairs have at most 16 words in the French sentence; this restriction on the sentence length is necessary to control the memory requirements of the IBM-3 composition.

We report two sets of experiments. In the first set, MBR alignments are obtained under Precision Error, Recall Error, and Alignment Error Loss Functions (Table 6.1).

Decoder	Alignment Performance (%)		
	Precision	Recall	AER
ML (IBM3)	78.9	86.6	18.1
MBR			
PE	*90.7*	66.9	21.3
RE	74.4	*87.9*	20.9
AE	87.6	81.9	*14.9*

Table 6.1: Performance (%) of MBR decoders over IBM-3 alignment lattices. We measure the quality of the ML alignments and the MBR alignments under Precision Error (PE), Recall Error (RE), and Alignment Error (AE). Results are shown under Alignment Precision, Alignment Recall, and Alignment Error Rate metrics. For each metric the error rate of the matched decoder is in italics.

We next report the performance of the MBR decoders under the Alignment Error and the Generalized Alignment Error Loss Functions (Table 6.2).

We observe that in none of these experiments (Tables 6.1 and 6.2) is the ML decoder found to be optimal. In all instances, the MBR decoder tuned for each loss function is the best performing decoder under the corresponding error rate. In particular, we note that alignment performance as measured under the AER metric can be improved by using MBR instead of ML alignment. This demonstrates the value of finding decoding procedures matched to the performance criterion of interest.

We observe some affinity among the loss functions. In particular, the ML decoder performs better under the AER than any of the MBR-GAE decoders. This is because the $L_{0/1}$ loss, for which the ML decoder is optimal, is closer to the L_{AE} loss than any of the L_{GAE} loss functions. The NULL symbol is treated quite differently under

		Generalized Alignment Error Rates (%)					
Decoder	AER	PTSD-S	POSD-S	AWCD-S	PTSD-A	POSD-A	AWCD-A
ML (IBM3)	18.1	3.1	4.4	4.7	29.4	51.4	54.6
MBR							
AE	*14.9*	1.3	1.9	1.9	19.8	36.4	38.6
PTSD	23.3	0.6	0.7	0.8	*14.5*	26.8	28.4
POSD	28.6	2.4	0.7	3.2	15.7	*26.3*	29.5
AWCD	24.7	1.0	1.0	0.9	14.9	26.8	*28.4*

Table 6.2: Performance (%) of MBR decoders over IBM-3 alignment lattices. We measure the alignment quality of the ML alignments and the MBR alignments under Alignment Error (AE) and Generalized Alignment Error using Parse-Trees (PTSD), Part-of-Speech Tags (POSD), and Automatic Word Classes (AWCD). Results are shown under Alignment Error Rate (AER) and the Generalized Alignment Error Rates. For each metric the error rate of the matched decoder is in italics.

L_{AE} and L_{GAE}, and this leads to a large mismatch between the MBR-GAE decoders and the AER metric. Similarly, the performance of the MBR-POS decoder degrades significantly under the AWCD-S and AWCD-A metrics. Since there are more word clusters (100) than POS tags (55), the MBR-POS decoder is incapable of producing hypotheses that can match the word clusters used in the AWCD metrics.

6.3.5 MBR decoders over TTM lattices

We next present the performance of MBR decoders over word alignment lattices generated by the Translation Template Model (TTM). We will report performance on the same test set used in Section 6.3.4. In this case we give results only under the Precision Error, Recall Error, and the Alignment Error loss functions (Table 6.3).

Under all the three loss functions, MBR alignments performs better (or no worse) than the ML alignments generated by the TTM. However, the gains from MBR decoding over TTM lattices are smaller relative to the corresponding gains over IBM-3 lattices (Table 6.1). To understand this behavior, we measure oracle-best values of alignment error rate, precision and recall on alignment lattices generated by both IBM-3 and TTM (Table 6.4). Under each metric, we define the Delta Score as the difference in alignment performance between the ML alignment and the oracle-best

Decoder	Alignment Performance (%)		
	Precision	Recall	AER
ML-TTM	95.1	86.3	8.9
MBR			
PE	*100.0*	1.5	96.9
RE	94.4	*86.6*	9.0
AE	95.1	86.3	*8.8*

Table 6.3: Performance (%) of MBR decoders over TTM alignment lattices. We measure the quality of the ML alignments and the MBR alignments under Precision Error (PE), Recall Error (RE), and Alignment Error (AE). Results are shown under Alignment Precision, Alignment Recall, and Alignment Error Rates. For each metric the error rate of the matched decoder is in bold.

alignment. This gives us the following three Delta Scores: Delta Precision, Delta Recall, and Delta AER. We observe that the Delta Precision, Delta Recall and Delta AER values for the IBM-3 lattices are much higher than the corresponding values for the TTM lattices. This suggests that IBM-3 lattices allow a greater room for improvement under all alignment metrics. This would explain why we obtain higher performance improvements from MBR decoding over IBM-3 lattices.

	IBM-3	TTM
Oracle-best Precision (%)	98.1	100
Delta Precision (%)	11.8	4.9
Oracle-best Recall (%)	97.3	88.9
Delta Recall (%)	10.7	2.6
Oracle-best AER (%)	2.1	5.9
Delta AER (%)	16.0	3.0

Table 6.4: Oracle-best Alignment Error Rate, Alignment Precision, and Alignment Recall on Alignment Lattices generated by the IBM-3 translation model and the TTM. The difference in alignment performance (Delta scores) between the Oracle-best hypothesis and the maximum likelihood hypothesis is also shown in each case.

6.4 Discussion

In this chapter we have described how Minimum Bayes-Risk decoding framework can be used to optimize alignment performance under various loss functions. Loss

functions can be derived from standard alignment evaluation metrics. We have also shown the construction of loss functions that incorporate information from varied analyses such as parse trees, POS tags, and automatically derived word clusters. We have derived and implemented lattice based MBR decoders under these loss functions. These decoders rescore the lattices produced by maximum likelihood decoding to produce the optimal MBR alignments.

We have performed MBR decoding over alignment lattices generated from two types of statistical translation models. These include a word-for-word translation model (IBM-3) and a phrase based translation model (TTM). However MBR decoding is not restricted to these frameworks. It can be applied more broadly using other MT model architectures that might be selected for reasons of modeling fidelity or computational efficiency.

We have presented these alignment loss functions to explore how linguistic knowledge might be incorporated into machine translation systems without building detailed statistical models of these linguistic features. However we stress that the MBR decoding procedures described here do not preclude the construction of complex MT models that incorporate linguistic features. The application of such models, which could be trained using conventional maximum likelihood estimation techniques, should still benefit by the application of MBR decoding techniques.

Finally we have presented procedures to compute oracle-best alignments under the various loss functions. Our experiments show that the gains from MBR decoding are linked to the difference in performance between the oracle-best alignment and the Maximum Likelihood alignment. A higher difference is seen to correspond with a bigger performance improvement via MBR decoding.

Chapter 7

Minimum Bayes-Risk Decoders for Translation

In this chapter we present the application of Minimum Bayes-Risk (MBR) decoding techniques to the problem of translating texts from one language to another. Our goal here is to describe how the MBR decoding framework can be used to build specialized Machine Translation (MT) decoders for each individual translation metric [48].

We will show that MBR decoding can be applied to machine translation in two scenarios. Given an automatic MT metric, we can design a loss function based on the metric and use MBR decoding to tune MT performance under the metric. We also show how MBR decoding can be used to incorporate syntactic structure into a statistical MT system by building specialized loss functions. These loss functions can use information from word strings, word-to-word alignments and parse-trees of the target sentence and its translation. In particular we describe the design of a *Bilingual Tree Loss Function* that can explicitly use syntactic structure for measuring translation quality. MBR decoding under this loss function allows us to integrate syntactic knowledge into a statistical MT system without building detailed models of linguistic features, and retraining the system from scratch.

We first present a hierarchy of loss functions for translation based on different levels of lexical and syntactic information from target and source language sentences

(Section 7.1). This hierarchy includes the loss functions useful in both situations where we intend to apply MBR decoding. We then present the formulation of MBR decoders for statistical machine translation under the various translation loss functions (Section 7.2). We report the performance of MBR decoders optimized for each loss function in Section 7.3. We end with a discussion in Section 7.4.

7.1 Translation Loss Functions

We here introduce translation loss functions to measure the quality of automatically generated translations. Suppose we have a sentence F in a target language for which we have generated an automatic translation E' with word-to-word alignment B' relative to F. The word-to-word alignment B' specifies the words in the target sentence F that are aligned to each word in the translation E'. We wish to compare this automatic translation with a reference translation E that has word-to-word alignment B relative to F.

We will now present a three-tier hierarchy of translation loss functions of the form $L((E,B),(E',B');F)$ that measure (E',B') against (E,B). These loss functions will make use of different levels of information from word strings, MT alignments and syntactic structure from parse-trees of both the source and target strings as illustrated in the following table.

Loss Function	Functional Form
Lexical	$L(E,E')$
Source Language Parse-Tree	$L(T_E,T_{E'})$
Bilingual Parse-Tree	$L((T_E,B),(T_{E'},B');T_F)$

We start with an example of two competing English translations for a Chinese sentence (shown in Pinyin without tones), with their word-to-word alignments in Figure 7.1. The reference translation for the Chinese sentence with its word-to-word alignment is shown in Figure 7.2. In this section, we will show the computation of different loss functions for this example.

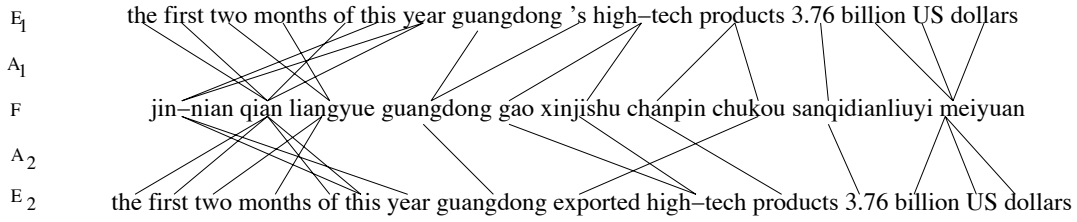

Figure 7.1: Two competing English translations for a Chinese sentence with their word-to-word alignments.

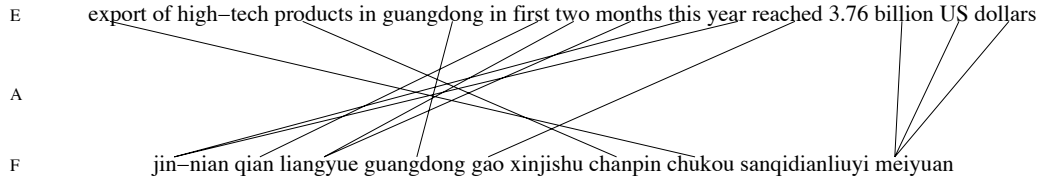

Figure 7.2: The reference translation for the Chinese sentence from Figure 7.1 with its word-to-word alignments. Words in the Chinese (English) sentence shown as unaligned are aligned to the NULL word in the English (Chinese) sentence.

7.1.1 Lexical Loss Functions

The first class of loss functions uses no information about word alignments or parse-trees, so that $L((E,B),(E',B');F)$ can be reduced to $L(E,E')$. We consider three loss functions in this category: BLEU score [86], word-error rate, and position-independent word-error rate [79]. Other examples of loss functions in this class are NIST score [21], and the F-score introduced in Melamed et.al. [67]. A loss function of this type depends only on information from word strings.

BLEU score [86] (defined in Section 5.2) measures the agreement between a hypothesis translations E' and its reference translation E by computing the geometric mean of the precision of their common n-grams ($n \in \{1..N\}$). The score also includes a 'Brevity Penalty' that is applied if the hypothesis is shorter than the reference. The BLEU score is non-zero if and only if all the N n-gram precisions are non-zero for that sentence pair. We note that $0 \leq BLEU(E,E') \leq 1$. We derive a loss function from BLEU score as

$$L_{\text{BLEU}}(E,E') = 1 - BLEU(E,E').$$

Word Error Rate (WER) is the ratio of the string-edit distance between the reference and the hypothesis word strings to the number of words in the reference. String-edit distance is measured as the minimum number of edit operations needed to transform a word string to the other word string.
Position-independent Word Error Rate (PER) measures the minimum number of edit operations needed to transform a word string to any permutation of the other word string. The PER score [79] is then computed as a ratio of this distance to the number of words in the reference word string.

7.1.2 Source Language Parse-Tree Loss Functions

The second class of translation loss functions uses information only from the parse-trees of the two translations (in the source language), so that $L((E, B), (E', B'); F) = L(T_E, T_{E'})$. This loss function has no access to any information from the target sentence or the word alignments.

Examples of such loss functions are tree-edit distances between parse-trees, string-edit distances between event representation of parse-trees [93], and tree-kernels [15]. The computation of tree-edit distance involves an unconstrained alignment of the two English parse-trees. We can simplify this problem once we have a third parse tree (for the Chinese sentence) with node-to-node alignment relative to the two English trees. We will introduce such a loss function in the next section. We do not perform experiments involving this class of loss functions, but mention them for completeness in the hierarchy of loss functions.

7.1.3 Bilingual Parse-Tree Loss Functions

The third class of loss functions uses information from word strings, alignments and parse-trees in both languages, and can be described by

$$L((E, B), (E', B'); F) = L((T_E, B), (T_{E'}, B'); T_F).$$

We will now describe one such loss function using the example in Figures 7.1 and 7.2. Figure 7.3 shows a tree-to-tree mapping between the target (Chinese) parse-tree

and parse-trees of its reference translation and two competing hypothesis (English) translations.

We assume that a node n in the target tree T_F can be mapped to a node m in T (and a node m' in T') using word alignment B (and B' respectively). We denote the subtree of T rooted at node m by t_m and the subtree of T' rooted at node m' by $t'_{m'}$. We will now describe a simple procedure that makes use of the word alignment B to construct node-to-node alignment between nodes in the target tree T_F and the source tree T.

Alignment of Parse-Trees

For each node n in the target tree T_F we consider the subtree t_n rooted at n. We first read off the target word sequence corresponding to the leaves of t_n. We next consider the subset of words in the source sentence that are aligned to any word in this target word sequence, and select the leftmost and rightmost words from this subset. We locate the leaf nodes corresponding to these two words in the source parse tree T, and obtain their closest common ancestor node $m \in T$. This procedure gives us a mapping from a node $n \in T_F$ to a node $m \in T$ and this mapping associates one subtree $t_n \in T_F$ to one subtree $t_m \in T$.

We note that the above procedure was used in JHU 2003 summer workshop group on *Syntax for Statistical Machine Translation* [81] to obtain alignments between parse-trees of Chinese sentences and parse trees of their English translations.

Loss Computation between Aligned Parse-Trees

Given the subtree alignment between T_F and T, and T_F and T', we first identify the subset of nodes in T_F for which we can identify corresponding nodes in both T and T'.

$$\bar{N}_F = \{n \in T_F : m \neq \epsilon \cap m' \neq \epsilon\}.$$

The *Bilingual Parse-Tree (BiTree) Loss Function* is then computed as

$$\text{BiTreeLoss}((T_E, B), (T_{E'}, B'); T_F) = \sum_{n \in \bar{N}_F} d(t_m, t'_{m'}), \tag{7.1}$$

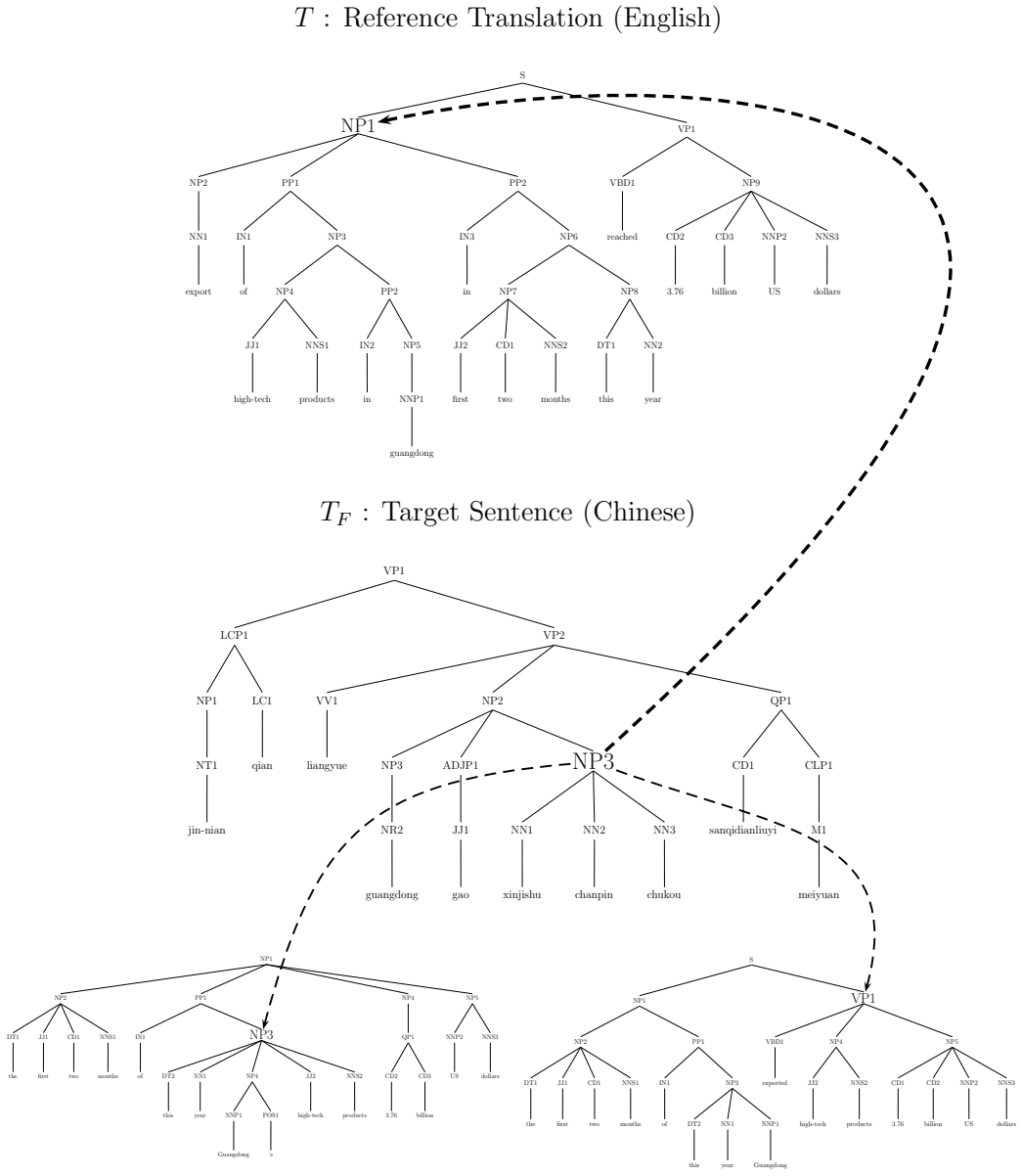

Figure 7.3: An example showing a parse-tree for a Chinese sentence and parse-trees for its reference translation and two competing hypothesis translations. We also show the alignment between one of the nodes in the Chinese tree with corresponding nodes in the three English trees. The complete node-to-node alignment between the Chinese parse tree and the three English trees is given in Table 7.1.

Target Node $n \in T_F$	Reference Node $m \in T$	Hypothesis 1 Node $m_1 \in T_1$	$L(t_m, t_{m_1})$ -	Hypothesis 2 Node $m_2 \in T_2$	$L(t_m, t'_{m_2})$
VP1	S	NP1	1	S	1
LCP	NP6	NP1	1	NP1	1
NP1	NP8	NP3	1	NP3	1
NT1	NP8	NP3	1	NP3	1
jin-nian	NP8	NP3	1	NP3	1
LC1	first	NP2	1	NP1	1
qian	first	NP2	1	NP1	1
VP2	S	NP1	1	S	1
VV	NP7	NP2	1	NP2	1
liangyue	NP7	NP2	1	NP2	1
NP2	S	NP1	1	S	1
NP3	Guangdong	NP4	1	Guangdong	0
NR2	Guangdong	NP4	1	Guangdong	0
guangdong	Guangdong	NP4	1	Guangdong	0
ADJP1	reached	high-tech	1	high-tech	1
JJ1	reached	high-tech	1	high-tech	1
gao	reached	high-tech	1	high-tech	1
NP3	NP1	NP3	1	VP1	1
NN2	products	products	0	products	0
chanpin	products	products	0	products	0
NN3	export	products	1	exported	1
chukou	export	products	1	exported	1
QP1	NP9	NP1	1	NP5	0
CLP1	NP9	NP1	1	NP5	0
M1	NP9	NP1	1	NP5	0
meiyuan	NP9	NP1	1	NP5	0
BiTree Loss		Loss(E, E_1)	24	Loss(E, E_2)	17
BiTree Error Rate			24/26 = 92.3		17/26 = 65.4

Table 7.1: Bi-Tree Loss Computation for the parse-trees shown in Figure 7.3. Each row shows a mapping between a node in the parse-tree of the Chinese sentence and the nodes in parse-trees of its reference translation and two hypothesis translations.

where $d(t, t')$ is a distance measure between sub-trees t and t'. Specific Bi-tree loss functions are determined through particular choices of d. In our experiments, we used a 0/1 loss function between sub-trees t and t'.

$$d(t, t') = \begin{cases} 1 & t \neq t' \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

We note that other tree-to-tree distance measures can also be used to compute d, e.g. the distance function could compare if the subtrees t and t' have the same headword/non-terminal tag.

The Bitree loss function measures the distance between two trees in terms of distances between their corresponding subtrees. In this way, we replace the string-to-string (Levenshtein) alignments (for WER) or n-gram matches (for BLEU/PER) with subtree-to-subtree alignments.

The *Bitree Error Rate* (in %) is computed as a ratio of the Bi-tree Loss function to the number of nodes in the set $\bar{N}_F$.

The complete node-to-node alignment between the parse-tree of the target (Chinese) sentence and the parse trees of its reference translation and the two hypothesis translations (English) is given in Table 7.1. Each row in this table shows the alignment between a node in the Chinese parse-tree and nodes in the reference and the two hypothesis parse-trees. The computation of the Bitree Loss function and the Bitree Error Rate is presented in the last two rows of the table.

7.1.4 Comparison of Loss Functions

In Table 7.2 we compare various translation loss functions for the example from Figure 7.1. The two hypothesis translations are very similar at the word level and therefore the BLEU score, PER and the WER are identical. However we observe that the sentences differ substantially in their syntactic structure (as seen from Parse-Trees in Figure 7.3), and to a lesser extent in their word-to-word alignments (Figure 7.1) to the target sentence. The first hypothesis translation does not have a main verb and is parsed as a noun phrase while the second translation has a main verb (*exported*) and is parsed as a sentence $S \rightarrow NP\ VP$. The Bitree loss function which depends

Metric	Loss Functions	$L(E, E_1)$	$L(E, E_2)$
BLEU	BLEU Loss (%)	73.6	73.6
WER	String edit distance	12	12
PER	Position Independent string edit distance	4	4
BiTree Error Rate	BiTree Loss	24	17

Table 7.2: Comparison of the different loss functions for hypothesis and reference translations from Figures 7.1, 7.2.

both on the parse-trees and the word-to-word alignments, is therefore very different for the two translations (Table 7.2). While string based metrics such as BLEU, WER and PER are insensitive to the syntactic structure of the translations, BiTree Loss is able to measure this aspect of translation quality, and assigns different scores to the two translations.

We provide this example to show how a loss function which makes use of syntactic structure from target and source parse trees, can capture properties of translations that string based loss functions are unable to measure.

7.2 Minimum Bayes-Risk Decoding

We here describe the formulation of Minimum Bayes-Risk (MBR) decoders for statistical machine translation and derive MBR rescoring procedures under the loss functions of Section 7.1.

Statistical Machine Translation can be described as a classification problem in which a sentence F in a target language is mapped to a sentence E' in the source language with word alignment B' relative to F. Given a translation loss function L and the true distribution $P(E, B, F)$ over alignments and translations, the classifier performance can be measured under the posterior risk $E_{P(E,B|F)}[L((E, B), \delta(F))]$. The classifier that minimizes the posterior risk is the Minimum Bayes-Risk decision rule (Equation 1.1)

$$\delta(F) = \underset{E',B'}{\operatorname{argmin}} \sum_{E,B} L((E, B), (E', B'); F)P(E, B|F). \tag{7.3}$$

In the above decoder, we do not have access to the true distribution over translations. We therefore use a statistical translation model to approximate the distribution $P(E, B|F)$. In addition, we consider the space of translations to be an N-best list of translation alternatives generated under the baseline translation model. We now describe the implementation of the MBR decoder as a rescoring procedure over an N-best list of translations.

Decoder Implementation The MBR decoder (Equation 7.3) on the N-best List is implemented as

$$\hat{i} = \underset{i \in \{1,2,\ldots,N\}}{\operatorname{argmin}} \sum_{j=1}^{N} L((E_j, B_j), (E_i, B_i)) P(E_j, B_j|F),$$

and $\delta(F) = (E_{\hat{i}}, B_{\hat{i}})$. This is a rescoring procedure that searches for consensus under a given loss function. The posterior probability of each hypothesis in the N-best list is derived from the joint probability assigned by the baseline translation model.

$$P((E_j, B_j)|F) = \frac{P(E_j, B_j, F)}{\sum_{i=1}^{N} P(E_i, B_i, F)}. \tag{7.4}$$

7.3 Performance of MBR Translations

We now report the performance of MBR decoders. Our experiments are performed on the Large Data Track of the 2001 and 2002 NIST Chinese-to-English MT task [78]. In our MBR decoding experiments, we consider two different translation models for generating N-best lists of translation hypotheses. The first of these models is the Alignment Template Translation Model developed by Och and Ney [81, 79] that served as the baseline model in the JHU 2003 Summer Workshop (WS '03) Research Group on *Syntax for Statistical Machine Translation*. The second model is the Translation Template Model described in Chapter 3.

7.3.1 Evaluation Metrics

The performance of the baseline and the MBR decoders under the different loss functions is measured with respect to the four reference translations provided for the

test set. Four evaluation metrics are used. These are multi-reference Word Error Rate (mWER) [79], multi-reference Position-independent word Error Rate (mPER) [79] , BLEU and multi-reference BiTree Error Rate.

Among these evaluation metrics, the BLEU score directly takes into account multiple reference translations [86]. In case of the other metrics, we consider multiple references in the following way. For each sentence, we compute the error rate of the hypothesis translation with respect to the most similar reference translation under the corresponding loss function.

7.3.2 MBR decoders over WS'03 N-best lists

We first report the performance of MBR decoders over N-best lists generated for the JHU 2003 Summer Workshop [81]. These N-best lists are generated by the Alignment Template Translation model [81] trained on a Chinese-English parallel corpus [78] ($170M$ English words and $157M$ Chinese words). 1000-best translation hypotheses are obtained for each Chinese sentence in the test set, and then rescored using the different translation loss functions described in Section 7.1.

We report results on a test set that has a total of 1791 sentences, consisting of 993 sentences from the NIST 2001 MT-eval set and 878 sentences from the NIST 2002 MT-eval set.

The English sentences in the N-best lists are parsed using the Collins parser [16], and the Chinese sentences are parsed using a Chinese parser provided to us by D. Bikel [8]. The English parser is trained on the Penn Treebank and the Chinese parser on the Penn Chinese treebank.

Under each loss function, the MBR decoding is performed using Equation 7.3. We say we have a matched condition when the same loss function is used in both the error rate and the decoder design. The performance of the MBR decoders on the NIST 2001+2002 test set is reported in Table 7.3.

	Performance Metrics			
Decoder	BLEU (%)	mWER(%)	mPER (%)	mBiTree Error Rate(%)
MAP(baseline)	31.2	64.9	41.3	69.0
MBR				
BLEU	*31.5*	65.1	41.1	68.9
WER	31.3	*64.3*	40.8	68.5
PER	31.3	64.6	*40.4*	68.6
BiTree Loss	30.7	64.1	41.1	*68.0*

Table 7.3: Translation performance of the MBR decoders on WS'03 N-best lists. We measure performance of the MAP decoder and the MBR decoders under BLEU, WER, PER, and BiTree loss functions. Results are reported on the NIST 2001+2002 Test set. For each metric, the performance under a matched condition is shown in italics. Note that better results correspond to higher BLEU scores and to lower error rates.

	Performance Metrics		
Decoder	BLEU (%)	mWER(%)	mPER (%)
MAP(baseline)	28.5	63.9	41.8
MBR			
BLEU	*28.8*	63.9	41.6
WER	28.8	*63.3*	41.3
PER	29.0	63.5	*41.0*

Table 7.4: Translation performance of the MBR decoders on TTM N-best lists. We measure the performance of the MAP decoder and the MBR decoders under BLEU, WER, and PER loss functions. Results are reported on the NIST 2002 Test set. For each metric, the performance under a matched condition is shown in italics. Note that better results correspond to higher BLEU scores and to lower error rates.

7.3.3 MBR decoders over TTM N-best lists

We next report the performance of MBR decoders on 1000-best lists generated by the Translation Template Model (Chapter 3). The TTM is trained on Chinese-English bitext (207.4M English words and 175.7M Chinese words) as part of the JHU 2004 system for the NIST 2004 Large Data Chinese-English MT task (See Section 5.3).

We report results on the NIST 2002 Chinese-English test sets consisting of 878 sentences. In this set of experiments, we report performance only under the BLEU, WER and the PER loss functions (Table 7.4).

7.3.4 Discussion

From Tables 7.3 and 7.4, we observe in most cases that the MBR decoder under a loss function performs the best under the corresponding error metric i.e. matched conditions perform the best. We note that the MAP decoder is not optimal in any of the cases. In particular, the translation performance under the BLEU metric can be improved by using MBR relative to MAP decoding. This shows the value of finding decoding procedures matched to the performance criterion of interest.

We also notice some affinity among the loss functions. The MBR decoding under the Bitree Loss function performs better under the WER relative to the MAP decoder, but perform poorly under the BLEU metric. The MBR decoder under WER and PER perform better than the MAP decoder under all error metrics.

7.4 Discussion

We have described the formulation of Minimum Bayes-Risk decoders for machine translation. We have focused on two situations where this framework could be applied.

Given an MT evaluation metric of interest such as BLEU, PER or WER, we can use this metric as a loss function within the MBR framework to design decoders optimized for the evaluation criterion. In particular, the MBR decoding under the BLEU loss function can yield further improvements on top of MAP decoding.

Suppose we are interested in improving syntactic structure of automatic translations and would like to use an existing statistical MT system that is trained without any linguistic features. We have shown in such a situation how MBR decoding can be applied to the MT system. This can be done by the design of translation loss functions from varied linguistic analyzes. We have shown the construction of a Bitree loss function to compare parse-trees of any two translations using alignments with respect to a parse-tree for the source sentence. The loss function therefore avoids the problem of unconstrained tree-to-tree alignment. Using an example, we have shown that this loss function can measure qualities of translation that string (and ngram) based metrics cannot capture. The MBR decoder under this loss function gives im-

provements in performance relative to this criterion but degrades performance under the BLEU metric. However it might be possible to construct other syntax based loss functions that yield performance improvements under the BLEU metric within the MBR framework.

We present results under the Bitree loss function as an example of incorporating linguistic information into a loss function; we have not yet measured its correlation with human assessments of translation quality. This loss function allows us to integrate syntactic structure into the statistical MT framework without building detailed models of syntactic features and retraining models from scratch. However we emphasize that the MBR techniques do not preclude the construction of complex models of syntactic structure. Translation models that have been trained with linguistic features could still benefit by the application of MBR decoding procedures.

We now contrast the MBR decoding procedure to the Minimum Error Training technique developed by Och [80]. These techniques are similar in that they incorporate the evaluation criterion (BLEU), but the Minimum Error Training technique of Och focuses on discriminative parameter estimation while MBR is a decoding procedure. Och's training technique assumes that the posterior probability over translations can be modeled using a log-linear model. This procedure estimates the parameters of the log-linear model so as to optimize a desired evaluation criterion on a development set. Translation of unseen test data is then performed under the MAP decoding criterion using the models estimated in training. By contrast, MBR decoding does not assume any underlying form of the statistical model. The MBR approach attempts to improve translation performance by matching the decoding criterion to the error criterion. Therefore the two approaches are complementary, and MBR decoding can be performed using models trained under the Minimum Error Training criterion (See [81]).

That machine translation evaluation continues to be an active area of research is evident from recent workshops [5]. We expect new automatic MT evaluation metrics to emerge frequently in the future. Given any translation metric, the MBR decoding framework will allow us to optimize existing MT systems for the new criterion. This is intended to compensate for any mismatch between decoding strategy of MT systems

and their evaluation criteria. While we have focused on developing MBR procedures for loss functions that measure various aspects of translation quality, this framework can also be used with loss functions which measure MT application-specific error criteria.

Part IV

Future Work and Conclusions

Chapter 8

Future Work

In this chapter we present some potential continuations of the work described in this dissertation.

8.1 Minimum Bayes-Risk Speech Recognition

In Chapter 2 we described a lattice-to-string alignment procedure that identifies node sets that can be used to segment the lattice. We then introduced Periodic Lattice Cutting as a cut set selection procedure that finds a balance between segmental MBR search errors and and errors due to poor approximation of the Levenshtein loss function. In this case it might be useful to investigate alternate approaches for choosing cut sets. Rather than selecting cut sets at equal intervals (as is done in Periodic Lattice Cutting), we might wish to employ other selection criteria that attempt to improve segmental MBR decoding. A possible criterion would be to minimize the overall increase in the expected risk via lattice segmentation. Other criteria could be based on measurements over segmented lattices such as lattice density, confidence measures, etc.

A second research direction is the extension of lattice-based MBR recognizers [33] to allow two separate lattices to form the hypothesis and evidence spaces. This extension would allow us to perform MBR search over a new lattice that contains more hypotheses relative to the original lattice, but is unweighted. An example

of such a lattice is the Word Transition Network (Section 2.3) that contains many more hypotheses relative to the N-best list from which it is created. By considering more hypotheses during search, we might obtain greater performance improvements through MBR decoding.

8.2 Translation Template Model

We next describe various possible extensions to the Translation Template Model (TTM).

Parameterized Phrase Transduction Models Inside the TTM generative process (Figure 3.2), source language phrases are mapped to target language phrases under the phrase transduction model (Section 3.4.5). In our experiments, the phrase transduction probabilities are estimated by the relative frequencies of phrase-pairs in the bitext word alignments. However, this estimate might be unreliable when the bitext is small or if the word alignments are noisy. In these cases a parameterized model, trained on phrase-pairs, could provide a better estimate of the phrase transduction probabilities. Parameterized phrase transduction models can be derived from bitext alignment models in the MT literature such as the IBM-1 [10], IBM-2 [10], or the HMM [102]. However, these models were originally proposed to describe the sentence-level translation process; therefore they might not be accurate enough to model phrase transductions. An interesting direction is the development of novel phrase transduction models that can result in better translation under the TTM.

Induction of Phrase-Pairs The TTM relies on an inventory of target language phrases and their translations in the source language. In our experiments (Chapters 4 and 5), we construct the phrase-pair inventory using the extraction procedure described in Section 3.3 [79]. This procedure obtains word alignments of the bitext training set under the IBM-4 translation models, and then extracts phrase-pairs from these alignments using a set of heuristics. The performance of the TTM is therefore very sensitive to the word alignment quality of IBM-4 models (Sections 4.3.3 and

5.2.3).

An alternate approach would be to induce phrase-pairs directly under the TTM without relying on word alignments from any underlying translation model. This approach would exploit the ability of TTM to perform bitext word alignment (Section 4.1). We now sketch the steps involved in this approach.

For each sentence pair, we first enumerate all source-target phrase pairs such that both the source phrase and the target phrase satisfy a maximum length requirement. This defines our initial phrase-pair inventory (PPI). Each iteration of the procedure can be described as follows.

1. Estimate a parameterized phrase transduction model over the current PPI.
2. Under the current PPI and phrase transduction model, use the TTM to obtain MAP word alignments over the training bitext.
3. Collect phrase-pairs from the MAP word alignments to obtain a new phrase-pair inventory.

The approach attempts to simultaneously improve word alignment quality and induce phrase-pair inventories. However, a possible drawback of this approach is the lower word alignment performance of TTM relative to the IBM-4 translation model (Section 4.3).

Context Modeling Under the Phrase Transduction Model (Section 3.4.5), source phrases are mapped to target phrases independent of their phrase context. A direction for future research is to introduce phrase-level context dependency in the phrase transduction model. Moving from context independent to context dependent models will lead to an explosion in the number of model parameters; we would therefore need to tie these parameters. A possible approach is to start with a parameterized phrase transduction model, and then make the parameters of this model depend on the left and right contexts of the phrase in question. The phrase contexts can then be clustered under a likelihood based criterion analogous to approaches used for triphone clustering in automatic speech recognition [108].

8.3 Minimum Bayes-Risk Decoders for Machine Translation

Chapter 7 discussed the implementation of MBR decoders for machine translation via rescoring of N-best translation hypotheses. A direction for future research here is the extension of the search space of MBR decoders to translation lattices produced by MT systems. While an N-best list contains only a limited re-ordering of hypotheses, a translation lattice contains more hypotheses with a vastly greater number of re-orderings. MBR decoding over lattices would require efficient lattice search procedures. By extending the search space of the decoder to a much larger space than the N-best list, we can expect further performance improvements in the MBR translation framework.

Chapter 9

Conclusions

This thesis has investigated the application of Minimum Bayes-Risk (MBR) classification techniques to three different problems in Automatic Speech Recognition and Statistical Machine Translation.

Chapter 1 reviewed the formulation of MBR decoders in automatic speech recognition. These decoders can become computationally intractable especially in large vocabulary speech recognition tasks. Chapter 2 presented the segmental MBR framework which simplifies the implementation of MBR decoders. The framework transforms the utterance level MBR recognition problem into a sequence of smaller, independent MBR recognizers that are easier to solve than the original problem. This is achieved by a lattice cutting strategy that segments a recognition word lattice into a sequence of smaller sub-lattices. Lattice cutting, in conjunction with MBR decoding, gives us consistent gains as a final stage of a large vocabulary speech recognition evaluation system. We also showed how segmental MBR framework can be used to describe and enhance ASR system combination procedures. Segmental MBR decoding provides a powerful framework for the development and description of novel ASR decoding strategies.

We next discussed the application of MBR procedures in statistical machine translation. A prerequisite for MBR decoding is the availability of statistical translation models for generating word alignment and translation hypotheses. Chapter 3 presented a weighted finite state transducer Translation Template Model (TTM) that

can generate N-best lists and lattices of word alignment and translation alternatives. Chapters 4 and 5 then discussed the word alignment and translation performance of this model. The TTM offers a simple and unified framework for bitext word alignment and translation. The simplicity of the model has allowed us to perform a detailed investigation of the several factors that influence the alignment and translation performance of the model. In the NIST 2004 international MT evaluation, the TTM Chinese-to-English system obtained a very competitive performance rivaling contemporary MT systems (Section 5.3).

Chapters 6 and 7 finally presented the formulation of MBR decoders for bitext word alignment and translation. In each case we showed the construction of loss functions either from standard evaluation metrics or from linguistic analyzes such as parse-trees and part-of-speech tags. In both word alignment and translation, MBR decoding yields consistent gains relative to Maximum A Posteriori (MAP) decoding.

We conclude that Minimum Bayes-Risk decoding is a promising framework for statistical modeling of speech and language. It embodies the philosophy that automatic systems should be constructed to incorporate the error criterion in decoding. This can allow us to compensate the mismatch between decoding criteria of systems and their error criteria. These criteria could come from evaluation metrics or from other desiderata (such as syntactic well-formedness) that we wish to see in outputs of automatic systems.

9.1 Highlights of Thesis Contributions

We here outline the three main research contributions of this dissertation.

Risk Based Lattice Segmentation Our first contribution is the development of a risk-based lattice segmentation procedure for Segmental Minimum Bayes-Risk speech recognition. The procedure allows us to segment a large word lattice into a sequence of smaller sub-lattices. The core of this procedure is a *lattice-to-string alignment* technique that produces a simultaneous Levenshtein alignment of all word strings in a lattice against any given word string. We utilize this technique to ob-

tain an alignment between the recognition lattice and the MAP hypothesis, and then use this alignment to identify candidate node sets for lattice cutting. Segmenting a lattice along fewer node sets results in better approximation to the Levenshtein loss function; we therefore develop a *Periodic Lattice Cutting* scheme to select a subset of the candidate node sets. By selecting node sets at equal intervals along the MAP word string, this procedure attempts to balance the errors in loss approximation with the search errors in MBR decoding. We have shown that SMBR decoding over lattice segments yields performance improvements over MBR decoding on unsegmented lattices. Lattice segmentation also allows us to perform SMBR decoding over lattices produced by multiple ASR systems. While our lattice segmentation procedure was originally motivated by the need to simplify MBR decoding, it has also formed the basis for novel estimation and classification procedures in automatic speech recognition [24, 25, 23, 22, 100, 99].

Translation Template Model Our second contribution is the formulation and implementation of a Weighted Finite State Transducer (WFST) *Translation Template Model* (TTM) for statistical machine translation. The TTM is a generative, source-channel model of phrase-based translation and it defines a series of stochastic transformations by which a French sentence is generated from an English sentence. Each of these stochastic transformations is formulated so that it can be implemented as a WFST. This approach allows the stochastic transformations to come together to define a complete probability distribution over French-English sentence pairs. Furthermore, translation and bitext word alignment can be realized almost immediately by standard algorithms to merge the component transducers into an overall processing system. This avoids the necessity to develop specialized search procedures (such as A^* decoding or beam search strategies) for performing word alignment and translation under the model. The framework also facilitates generation of alignment and translation lattices without any extra effort in implementation.

The TTM is the first phrase-based translation model to be used for bitext word alignment. The ability to do this is crucial to implement iterative parameter estimation procedures such as Expectation Maximization (EM) for this model.

We have used the TTM in constructing a Chinese-to-English translation system from large training bitexts for the NIST 2004 evaluation; this system ranked among the very top performing MT systems in this blind international evaluation.

Minimum Bayes-Risk Procedures for Word Alignment and Translation Our final contribution is the development of Minimum Bayes-Risk decoding procedures for bitext word alignment and translation. In both cases we have presented loss functions for comparing automatic word alignments and translations relative to references created by human translators. We have described the construction of loss functions either from standard evaluation criteria or from linguistic analyzes of sentences via parse-trees and part-of-speech tags. For bitext word alignment we have derived closed form solutions to MBR decoders on alignment lattices; this is done for two classes of loss functions. For translation we have implemented MBR decoders via rescoring of N-best lists under various loss functions. We have shown that the MBR framework can be used to build specialized MT decoders under each individual alignment and translation loss function.

Part V

Appendices and Bibliography

Appendix A

IBM-3 and IBM-4 translation models

In this appendix we briefly review the IBM translation models 3 and 4 [10, 43]. Each of these models is a generative model of the translation process that specifies a method to compute the conditional probability $P(F = f|E = e)$ for pairs of translations (e, f) in the source and target languages. In both models a source sentence e can produce the same target sentence f in several ways; each way is covered by a word alignment $A = a$ and assigned the probability $P(F = f, A = a|E = e)$. In the IBM models, the alignment A is restricted such that each target word is connected to at most one source word. The set of source words that generates a target word is referred to as a *cept*; therefore each cept in the IBM-style alignment consists of either a single source word or is empty.

If the source string $e = e_1^l = e_1 e_2 ... e_l$ has l words and the target string $f = f_1^m = f_1 f_2 ... f_m$ has m words, then the alignment a is represented by the m length sequence $a = a_1 a_2 ... a_m$, where a_i indicates the position of the source word to which i^{th} target word is aligned. Therefore if $a_i = j$ the target word f_i is aligned to source word e_j. If a target word f_i is not aligned to any source word, it is aligned to the NULL word e_0 so that $a_i = 0$.

The probability $P(f|e)$ can be obtained in terms of the distribution $P(f, a|e)$:

$$P(f|e) = \sum_{a} P(f, a|e), \tag{A.1}$$

where a ranges over all possible alignments of (e, f).

The generative process underlying the IBM-3 and IBM-4 translation models transforms the source sentence $e = e_1^l$ into the target sentence $f = f_1^m$, and can be described as follows. For each source word e_i, we first choose a *fertility* Φ_{e_i} that decides how many target words are connected to it. We next decide the list of target words T_i connected to e_i; the k^{th} word in T_i is denoted T_{ik}. We refer to T_i as a *tablet* and the collection of tablets $T_1 T_2 ... T_l$ is called a *tableau*. We finally permute the target words in the tableau to produce $f = f_1^m$. The permutation is a random variable Π; the position in f of the k^{th} word in T_i is given by Π_{ik}.

The joint likelihood of a tableau τ and a permutation π is given by

$$\begin{aligned} P(\tau, \pi|e) \;=\; & \prod_{i=1}^{l} P(\phi_i|\phi_1^{i-1}, e) P(\phi_0|\phi_1^l, e) \\ & \times \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} P(\tau_{ik}|\tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^l, e) \\ & \times \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} P(\pi_{ik}|\pi_{i1}^{k-1}, \pi_0^{i-1}, \tau_0^l, \phi_0^l, e) \\ & \times \prod_{k=1}^{\phi_0} P(\pi_{0k}|\pi_{01}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e), \end{aligned} \tag{A.2}$$

where τ_{i1}^{k-1} represents the series of values $\tau_{i1}, ..., \tau_{ik-1}$; π_{i1}^{k-1} represents the series of values $\pi_{i1}, ..., \pi_{ik-1}$; and ϕ_i is a shortcut for ϕ_{e_i}. We note that in general several τ, π pairs may lead to the same pair f, a. We denote the set of such pairs $< f, a >$. Then

$$P(f, a|e) = \sum_{(\tau,\pi) \in <f,a>} P(\tau, \pi|e). \tag{A.3}$$

In the next two sections we outline the modeling assumptions underlying IBM Models 3 and 4. Parameter estimation for these models is discussed in detail in the original IBM publication [10]; we will not review these procedures here.

A.1 Model 3

Assumptions

- **Fertility Probabilities**
  - For $i \in \{1, 2..., l\}$, $P(\phi_i|\phi_1^{i-1}, e) = n(\phi_i|e_i)$
  - $P(\phi_0|\phi_1^l, e) = \binom{\phi_1 + ... + \phi_l}{\phi_0} {p_0}^{\phi_1+...+\phi_l-\phi_0} {p_1}^{\phi_0}$.
- **Translation Probabilities**
  - For $i \in \{1, 2, ..., l\}$, $P(\tau_{ik}|\tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^l, e) = t(\tau_{ik}|e_i)$.
- **Distortion Probabilities**
  - For $i \in \{1, 2, ..., l\}$, $P(\pi_{ik}|\pi_{i1}^{k-1}, \pi_0^{i-1}, \tau_0^l, \phi_0^l, e) = d(\pi_{ik}|i, m, l)$.
  - For $i = 0$ and $k \in \{1, 2, .., \phi_0\}$, $P(\pi_{0k}|\pi_{01}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e) = \frac{1}{\phi_0!}$.

Parameters

- $t(f|e)$ is *translation probability* of target word f given source word e.
- $n(\phi|e)$ is the *fertility probability* of source word e.
- p_0, p_1 govern the *fertility probability for the NULL word* e_0.
- $d(j|i, l, m)$ is the *distortion probability* of the j^{th} target word given that it is connected to the i^{th} source word; l and m are the lengths of the source and the target sentence.

Joint Probability The joint probability of a target string f and an alignment a given the source string e is given by:

$$P(f, a|e) = \binom{m - \phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \prod_{j=1}^{m} t(f_j|e_{a_j}) \prod_{j=1, a_j \neq 0}^{m} d(j|a_j, m, l). \quad \text{(A.4)}$$

A.2 Model 4

Assumptions IBM Model 4 differs from Model 3 in its distortion probabilities.

As in Model 3, target words connected to the NULL word are assumed to be spread uniformly throughout the target string, i.e. for $i = 0$ and $k \in \{1, 2, .., \phi_0\}$, $P(\pi_{0k}|\pi_{01}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e) = \frac{1}{\phi_0!}$.

Distortion probabilities for the target words connected to source words are specified in the following way. We first recall that an alignment resolves the source string into cepts. Each cept consists of one or zero source words and accounts for one or more target words. Among the one-word cepts, there is a natural order corresponding to the order in which they appear in the source string. Let $[i]$ denote the position in the source string of the i^{th} one-word cept. We define the center of the cept $\odot_i$ to be the ceiling of the average value of positions in the target string from its tablet. We define its head to be that word in its tablet for which the position in the target string is smallest.

In Model 4 the distortion probabilities $d(j|i, m, l)$ are replaced by two sets of parameters: one for placing the head of each cept, and one for placing any remaining words. For $[i] > 0$ we require that head for cept i be $\tau_{[i]1}$ and we assume that

$$P(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, e) = d_1(j - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(f_j)). \tag{A.5}$$

Here $\mathcal{A}$ and $\mathcal{B}$ are classes of the source and the target word respectively. $j - \odot_{i-1}$ is called the displacement for the head of cept i.

For placing the k^{th} word of cept i for $[i] > 0$, $k > 1$, we assume that

$$P(\Pi_{[i]k} = j|\pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, e) = d_{>1}(j - \pi_{[i]k-1}|\mathcal{B}(f_j)). \tag{A.6}$$

We assume that subsequent words from $\tau_{[i]}$ are placed in order, i.e. the second word from $\tau_{[i]}$ must lie to the right of the first word, and so on.

Parameters

- $t(f|e)$ is *translation probability* of target word f given source word e.

- $n(\phi|e)$ is the *fertility probability* of source word e.
- p_0, p_1 govern the *fertility probability for the NULL word* e_0.
- $d_1(\Delta j|\mathcal{A}, \mathcal{B})$ is the distortion probability for the first word of a tablet.
- $d_{>1}(\Delta j|\mathcal{B})$ is the distortion probability for other words of the tablet.

Here Δj is an integer; $\mathcal{A}$ is a source word class and $\mathcal{B}$ is a target word class.

Joint Probability The joint probability of a target string f and an alignment a given the source string e is given by:

$$\begin{aligned} P(f, a|e) &= \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \\ &\times \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{ik}|e_i) \prod_{i=1,\phi_i>0}^{l} d_1(\pi_{i1} - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(\tau_{i1})) \\ &\times \prod_{i=1}^{l} \prod_{k=2}^{\phi} d_{>1}(\pi_{ik} - \pi_{ik-1}|\mathcal{B}(\tau_{ik})). \end{aligned} \tag{A.7}$$

Bibliography

[1] C. Allauzen, M. Mohri, and B. Roark. Generalized algorithms for constructing statistical language models. In *Proceedings of the 41th annual meeting of the Association for Computational Linguistics (ACL)*, pages 40–47, Sapporo, Japan, 2003.

[2] J.C. Amengual, J.M. Bened, F. Casacuberta, A. Castao, A. Castellanos, V. Jimnez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[3] S. Bangalore, V. Murdock, and G. Riccardi. Bootstrapping bilingual data using consensus translation for a multilingual instant messaging system. In *International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, 2002.

[4] S. Bangalore and G. Ricardi. A finite-state approach to machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Pittsburgh, PA, USA, 2001.

[5] L. Barrett, M. King, K. Miller, and A. Popescu-Belis. Workshop on Machine Translation Evaluation, MT Summit IX, 2003. www.issco.unige.ch/projects/isle/MTE-at-MTS9.html.

[6] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.

[7] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected topics*. Holden-Day Inc., Oakland, CA, USA, 1977.

[8] D. Bikel and D. Chiang. Two statistical parsing models applied to the chinese treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 1–6, Hong Kong, 2000.

[9] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J . Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[11] W. Byrne, V. Doumpiotis, S. Kumar, S. Tsakalidis, and V. Venkataramani. The JHU 2002 large vocabulary speech recognition system. In *Proceedings of the NIST RT-02 Workshop*. NIST, 2002. Available at http://www.clsp.jhu.edu/research/rteval/.

[12] W. Byrne, A. Gunawardana, S. Kumar, and V. Venkataramani. The JHU March 2001 Hub-5 conversational speech transcription system. In *Proceedings of the NIST LVCSR Workshop*. NIST, 2001. Available at http://www.clsp.jhu.edu/research/rteval/.

[13] W. Byrne, S. Khudanpur, W. Kim, S. Kumar, P. Pecina, P. Virga, P. Xu, and D. Yarowsky. The Johns Hopkins University 2003 Chinese-English machine translation system. In *Proceedings of the MT Summit IX*, pages 447–450, New Orleans, LA, USA, 2003.

[14] M. Carl and S. Fissaha. Phrase-based evaluation of word-to-word alignments. In *NAACL Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton Canada, 2003.

[15] M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the weighted perceptron. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, USA, 2002.

[16] M. J. Collins. *Head-driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1999.

[17] World News Connection, 2004. http://wnc.fedworld.gov/.

[18] The People's Daily, 2002. http://www.english.people.com.cn.

[19] L. Damianos, S. Bayer, M. A. Chisholm, J. Henderson, L. Hirschman, W. Morgan, M. Ubaldino, and J. Zarrella. MiTAP for SARS detection. In *Proceedings of the Conference on Human Language Technology (Companion Volume)*, pages 241–244, Boston, MA, USA, 2004.

[20] Y. Deng, S. Kumar, and W. Byrne. Bitext chunk alignment for statistical machine translation. In *Research Note, Center for Language and Speech Processing*, 2004.

[21] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Conference on Human Language Technology*, San Diego, CA. USA, 2002.

[22] V. Doumpiotis and W. Byrne. Lattice segmentation and minimum Bayes risk discriminative training for large vocabulary continuous speech recognition. *Speech Communication.* Submitted.

[23] V. Doumpiotis and W. Byrne. Pinched lattice minimum Bayes-risk discriminative training for large vocabulary continuous speech recognition. In *International Conference on Spoken Language Processing*, Jeju Island, Korea, 2004.

[24] V. Doumpiotis, S. Tsakalidis, and W. Byrne. Discriminative training for segmental minimum Bayes-risk decoding. In *International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 2003.

[25] V. Doumpiotis, S. Tsakalidis, and W. Byrne. Lattice segmentation and minimum Bayes-risk discriminative training. In *European Conference on Speech Communication and Technology*, Geneva, Switzerland, 2003.

[26] D. K. Evans, J. L. Klavans, and K. R. McKeown. Columbia newsblaster: Multilingual news summarization on the web. In *Proceedings of the Conference on Human Language Technology (Companion Volume)*, pages 229–232, Boston, MA, USA, 2004.

[27] G. Evermann and P. Woodland. Posterior probability decoding, confidence estimation and system combination. In *Proceedings of the NIST Speech Transcription Workshop*, College Park, MD, USA, 2000.

[28] J. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, 1997.

[29] G. Foster, P. Langlais, and G. Lapalme. User-friendly text prediction for translators. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, USA, 2002.

[30] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *International Conference on Computational Linguistics (COLING)*, pages 228–235, New Brunswick, NJ, USA, 2001.

[31] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone Speech Corpus for Research and Development. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, SanFrancisco, CA, USA, 1992.

[32] V. Goel. *Minimum Bayes-Risk Automatic Speech Recognition*. PhD thesis, Johns Hopkins University, Baltimore, MD, USA, 2001.

[33] V. Goel and W. Byrne. Minimum Bayes-risk automatic speech recognition. *Computer Speech and Language*, 14(2):115–135, 2000.

[34] V. Goel and W. Byrne. Recognizer output voting and DMC in minimum Bayes-risk framework. In *Research Notes No. 40, Center for Language and Speech Processing*, 2000.

[35] V. Goel, S. Kumar, and W. Byrne. Segmental minimum Bayes-risk ASR voting strategies. In *International Conference on Spoken Language Processing*, volume 3, pages 139–142, Beijing, China, 2000.

[36] V. Goel, S. Kumar, and W. Byrne. Confidence based lattice segmentation and minimum Bayes-risk decoding. In *European Conference on Speech Communication and Technology*, volume 4, pages 2569–2572, Aalborg, Denmark, 2001.

[37] V. Goel, S. Kumar, and W. Byrne. Segmental Minimum Bayes-risk Decoding for Automatic Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 12(3):287–333, 2004.

[38] J. Goodman. Parsing algorithms and metrics. In *Proceedings of the 34th annual meeting of the Association for Computational Linguistics (ACL)*, pages 177–183, Santa Cruz, CA, USA, 1996.

[39] E. Hovy. Towards finely differentiated evaluation metrics for machine translation. In *Proceedings of the Eagles Workshop on Standards and Evaluation*, Pisa, Italy, 1999.

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

[41] J. Kaiser, B. Horvat, and Z. Kacic. A novel loss function for the overall risk criterion based discriminative training of HMM models. In *International Conference on Spoken Language Processing*, volume 2, pages 887–890, Beijing, China, 2000.

[42] R. Kneser and H. Ney. Forming word classes by statistical clustering for statistical language modelling. In *The 1st Quantititative Linguistics Conference*, Trier, Germany, 1991.

[43] K. Knight and Y. Al-Onaizan. Translation with finite-state devices. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, pages 421–437, Langhorne, PA, USA, 1998.

[44] P. Koehn, F. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Conference on Human Language Technology*, pages 127–133, Edmonton, Canada, 2003.

[45] S. Kumar and W. Byrne. Minimum Bayes-risk alignment of bilingual texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 140–147, Philadelphia, PA, USA, 2002.

[46] S. Kumar and W. Byrne. Risk based lattice cutting for segmental minimum Bayes-risk decoding. In *International Conference on Spoken Language Processing*, pages 373–376, Denver, CO, USA, 2002.

[47] S. Kumar and W. Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the Conference on Human Language Technology*, pages 142–149, Edmonton, Canada, 2003.

[48] S. Kumar and W. Byrne. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Human Language Technology*, pages 169–176, Boston, MA, USA, 2004.

[49] S. Kumar and W. Byrne. A weighted finite state transducer translation template model for statistical machine translation. In *Research Note No. 48, Center for Language and Speech Processing*, 2004.

[50] S. Kumar, Y. Deng, and W. Byrne. A weighted finite state transducer translation template model for statistical machine translation. *Journal of Natural Language Engineering*. To appear.

[51] LDC. *Chinese Segmenter*, 2002. http://www.ldc.upenn.edu/Projects/Chinese.

[52] LDC. *Multiple-Translation Chinese Corpus*, 2002. LDC Catalog Number LDC2002T01.

[53] LDC. *Sinorama Chinese-English Parallel Text*, 2002. LDC Catalog Number LDC2002E58.

[54] LDC. *Xinhua Chinese-English Parallel News Text Version 1.0 beta 2*, 2002. LDC Catalog Number LDC2002E18.

[55] LDC. *Chinese Treebank English Parallel Corpus*, 2003. LDC Catalog Number LDC2003E07.

[56] LDC. *English Gigaword Corpus*, 2003. LDC Catalog Number LDC2003T05.

[57] LDC. *FBIS Chinese-English Parallel Corpus*, 2003. LDC Catalog Number LDC2003E14.

[58] LDC. *Hong Kong Hansard Parallel Text*, 2003. LDC Catalog Number LDC2004E09.

[59] LDC. *Hong Kong News Parallel Text*, 2003. LDC Catalog Number LDC2003E25.

[60] LDC. *Multiple-Translation Chinese Corpus Part 2*, 2003. LDC Catalog Number LDC2003T17.

[61] LDC. *UN Chinese-English Parallel Text Version 2*, 2004. LDC Catalog Number LDC2004E12.

[62] V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information transmission*, 1(1):8–17, 1965.

[63] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000.

[64] D. Marcu and U. Germann. *The ISI ReWrite Decoder Release 0.7.0b*, 2002. http://www.isi.edu/licensed-sw/rewrite-decoder/.

[65] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Philadelphia, PA, USA, 2002.

[66] I. D. Melamed. Empirical methods for MT lexicon development. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, pages 18–30, 1998.

[67] I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In *Proceedings of the Conference on Human Language Technology*, Edmonton, Canada, 2003.

[68] M. Mittendorfer and W. Winiwarter. Experiments with the use of syntactic analysis in information retrieval. In *Proceedings of the 6th International Workshop on Applications of Natural Language and Information Systems*, Bonn, Germany, 2001.

[69] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(3), 1997.

[70] M. Mohri. Edit-distance of weighted automata. In Jean-Marc Champarnaud and Denis Maurel, editors, *Seventh International Conference on Implementation and Application of Automata*, 2002.

[71] M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

[72] M. Mohri, F. Pereira, and M. Riley. *AT&T General-purpose finite-state machine software tools*, 1997. http://www.research.att.com/sw/tools/fsm/.

[73] M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

[74] M. Mohri, F. C. N. Pereira, and M. Riley. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231(1):17–32, 2000.

[75] K. Na, B. Jeon, D. Chang, S. Chae, and S. Ann. Discriminative training of hidden markov models using overall risk criterion and reduced gradient method. In *European Conference on Speech Communication and Technology*, pages 97–100, Madrid, Spain, 1995.

[76] A. Nadas. A Decision Theoretic Formulation of the Training Problem in Speech Recognition and a Comparison of Training by Unconditional Versus Conditional Maximum Likelihood. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-31(4):814–817, 1983.

[77] A. Nadas. Optimal Solution of a Training Problem in Speech Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33(1):326–329, 1985.

[78] NIST. The NIST Machine Translation Evaluations, 2002–2004. http://www.nist.gov/speech/tests/mt/.

[79] F. Och. *Statistical Machine Translation: From Single Word Models to Alignment Templates*. PhD thesis, RWTH Aachen, Germany, 2002.

[80] F. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41th annual meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003.

[81] F. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. Syntax for statistical machine translation. Technical report, JHU, 2003. http://www.clsp.jhu.edu/ws2003/groups/translate/.

[82] F. Och and H. Ney. Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the Association for Computational Linguistics (ACL)*, pages 440–447, Hong Kong, China, 2000.

[83] F. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, College Park, MD, USA, 1999.

[84] F. Och, N. Ueffing, and H. Ney. An efficient A* search algorithm for statistical machine translation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics (ACL)*, pages 55–62, Toulouse, France, July 2001.

[85] D. D. Palmer, P. Bray, M. Reichman, K. Rhodes, N. White, A. Merlino, and F. Kubala. Multilingual video and audio news alerting. In *Proceedings of the Conference on Human Language Technology (Companion Volume)*, pages 245–246, Boston, MA, USA, 2004.

[86] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, 2001.

[87] Canadian Parliament. Canadian hansards, 2003. http://www.parl.gc.ca/.

[88] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Philadelphia, PA, USA, 1996.

[89] D. Sankhoff and J. B. Kruskal. *Time Warps, String Edits and Macromolecules: The Theory and Practice of String Comparison.* Addison-Wesley Publishing Company, Inc., Reading, MA, USA, 1983.

[90] A. Stolcke. SRILM – an extensible language modeling toolkit. In *International*

Conference on Spoken Language Processing, pages 901–904, Denver, CO, USA, 2002. http://www.speech.sri.com/projects/srilm/.

[91] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng. The SRI March 2000 Hub-5 conversational speech transcription system. In *Proceedings of the NIST Speech Transcription Workshop*, College Park, MD, USA, 2000.

[92] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *European Conference on Speech Communication and Technology*, volume 1, pages 163–165, Rhodes, Greece, 1997.

[93] M. Tang, X. Luo, and S. Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, 2002.

[94] C. Tillmann. *Word Reordering and Dynamic Programming based Search Algorithm for Statistical Machine Translation*. PhD thesis, RWTH Aachen, Germany, 2001.

[95] C. Tillmann. A projection extension algorithm for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, 2003.

[96] S. Tsakalidis, V. Doumpiotis, and W. Byrne. Discriminative linear transforms for feature normalization and speaker adaptation in HMM estimation. In *International Conference on Spoken Language Processing*, pages 2585–2588, Denver, CO, USA, 2002.

[97] G. Tur, J. Wright, A. Gorin, G. Riccardi, and D. Hakkani-Tur. Improving spoken language understanding using confusion networks. In *International Conference on Spoken Language Processing*, Denver, CO, USA, 2002.

[98] N. Ueffing, F. Och, and H. Ney. Generation of word graphs in statistical machine

translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 156–163, Philadelphia, PA, USA, 2002.

[99] V. Venkataramani, S. Chakrabartty, and W. Byrne. *Gini*support vector machines for segmental minimum Bayes-risk decoding of continuous speech. *Computer Speech and Language*. Submitted.

[100] V. Venkataramani, S. Chakrabartty, and W. Byrne. Support vector machines for segmental minimum Bayes-risk decoding of continuous speech. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003.

[101] E. Vidal. Finite-state speech-to-speech translation. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 111–114, Munich, Germany, 1997.

[102] S. Vogel, H. Ney, and C. Tillmann. HMM based word alignment in statistical translation. In *International Conference on Computational Linguistics (COLING)*, pages 836–841, Copenhagen, Denmark, 1996.

[103] Y. Wang and A. Weibel. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th annual meeting of the Association for Computational Linguistics (ACL)*, pages 366–372, Madrid, Spain, 1997.

[104] F. Wessel, K. Macherey, and R. Schlueter. Using word probabilities as confidence measures. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 225–228, Seattle, WA, USA, 1998.

[105] J. S. White and T. O'Connell. The ARPA MT evaluation methodologies. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, pages 193–205, Columbia, MD, USA, 1994.

[106] D. Yarowsky and G. Ngai. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 200–207, 2001.

[107] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book, Version 3.1*, December 2001.

[108] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of ARPA Workshop on Human Language Technology*, pages 307–312, 1994.

[109] R. Zens and H. Ney. Improvements in phrase-based statistical machine translation. In *Proceedings of the Conference on Human Language Technology*, pages 257–264, Boston, MA, USA, 2004.

[110] Y. Zhang, S. Vogel, and A. Weibel. Integrated phrase segmentation and alignment model for statistical machine translation. In *Proceedings of the Workshop on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

Vita

Shankar Kumar was born in Chennai, India on August 3, 1977. In 1998, he graduated from the Birla Institute of Technology and Science, Pilani, India with a B.E. (Honors) degree in Electrical and Electronics Engineering. He received a M.S.E. degree in Electrical and Computer Engineering from the Johns Hopkins University in 2000. Since then, he has been pursuing his Ph.D. in Electrical and Computer Engineering at the Center for Language and Speech Processing at the Johns Hopkins University.

During the summer of 2000, he interned at Microsoft Research. He also participated in the 1999 and 2003 summer research workshops in language engineering at the Center for Language and Speech Processing.

His research interests include statistical modeling and classification techniques, and their applications to machine translation, speech recognition, and other problems in speech and language processing.