Bitext Alignment for Statistical Machine Translation

Yonggang Deng

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

Baltimore, Maryland
2005

Abstract

Bitext alignment is the task of finding translation equivalence between documents in two languages, collections of which are commonly known as bitext. This dissertation addresses the problems of statistical alignment at various granularities from sentence to word with the goal of creating Statistical Machine Translation (SMT) systems.

SMT systems are statistical pattern processors based on parameterized models estimated from aligned bitext training collections. The collections are large enough that alignments must be created using automatic methods. The bitext collections are often available as aligned documents, such as news stories, which usually need to be further aligned at the sentence level and the word level before statistics can be extracted from the bitext. We develop statistical models that are learned from data in an unsupervised way. Language independent alignment algorithms are derived for efficiency and effectiveness. We first address the problem of extracting bitext chunk pairs, which are translation segments at the sentence or sub-sentence level.

To extract these bitext chunk pairs, we formulate a model of translation as a stochastic generative model over parallel documents, and derive several different alignment procedures through various formulations of the component distributions. Based on these models we propose a hierarchical chunking procedure that produces chunk pairs by a series of alignment operations in which coarse alignment of large sections of text is followed by a more detailed alignment of their subsections. We show practical benefits with this chunking scheme, observing in particular that it makes efficient use of bitext by aligning sections of text that simpler procedures would discard as spurious.

For the problem of word alignment in bitext, we propose a novel Hidden Markov Model based Word-to-Phrase (WtoP) alignment model, which is formulated so that

alignment and parameter estimation can be performed efficiently using standard HMM algorithms. We find that the word alignment performance of the WtoP model is comparable to that of IBM Model-4, currently considered the state of the art, even in processing large bitext collections. We use this Word-to-Phrase model to define a posterior distribution over translation phrase pairs in the bitext, and develop a phrase-pair extraction procedure based on this posterior distribution. We show that this use of the phrase translation posterior distribution allows us to extract a richer inventory of phrases than results from with current techniques. In the evaluation of large Chinese-English SMT systems, we find that systems derived from word-aligned bitext created using the WtoP model perform comparably to systems derived from Model-4 word alignments, and in Arabic-English we find significant gains from using WtoP alignments.

Advisor: Prof. William Byrne.

Readers: Prof. William Byrne and Prof. Trac D. Tran.

Thesis Committee: Prof. William Byrne, Prof. Trac D. Tran, Prof. Jerry Prince and Prof. Gerard G. L. Meyer.

Acknowledgements

My first and foremost thank goes to my advisor, Prof. Bill Byrne for his support, guidelines and encouragements, without which this dissertation would not have been possible. I thank him for his countless insightful discussions, allowing me freedom in exploring new ideas while giving me directions whenever needed. His research style and scientific persistence have influenced me during my study and will continue to shape my skills in the future.

I am very grateful to Prof. Frederick Jelinek for giving me the opportunity and privilege to work at the Center for Speech and Language Processing (CLSP). I am thankful to Prof. Sanjeev Khudanpur for his guidelines on language model. The work on LSA-based language models in the second part of this thesis was carried out under his supervision. Special thanks to CLSP faculty members, Frederick Jelinek, Bill Byrne, David Yarowsky, Sanjeev Khudanpur, Jason Eisner and Izhak Shafran, for their consistent hark work of making CLSP a dynamic and enjoyable place to work in.

I thank my thesis committee members, Prof. Bill Byrne, Prof. Trac Tran, Prof. Jerry Prince and Prof. Gerard Meyer, for their valuable feedback which helped me to improve the dissertation in many ways, and for their timely flexibility.

I am thankful to CLSP senior students, Ciprian Chelba, Asela Gunarwardana, Jun Wu, Vlasios Doumpiotis, Shankar Kumar, Woosung Kim, Veera Venkataramani, Peng Xu, Ahmad Emami, Stavros Tsakalidis, Gideon Mann, Charles Schafer, for sharing their experiences in CLSP and many technical assistances. Special thanks to Shankar Kumar for providing the TTM decoder and insightful discussions, Peng Xu for many technical and non-technical discussions, Veera Venkataramani for sharing a lot of fun in Barton 322.

My 5.5 years' study in Hopkins was made enjoyable by my colleagues and friends at CLSP, notably Jia Cui, Erin Fitzgerald, Arnab Ghoshal, Lambert Mathias, Srividya Mohan, Srihari Reddy, Yi Su, Paola Virga, Christopher White, Ali Yazgan, Lisa Yung and Haolang Zhou. I enjoyed all the vivid discussions we had on various topics and had lots of fun being a member of this fantastic ECE side of CLSP.

I am grateful to Laura Graham and Sue Porterfield of the CLSP administrative staff for their professional help in making all administrative matters run smoothly and perfectly. Special thanks to Eiwe Lingefors and Jacob Laderman for their hard work keeping the CLSP grid power up and running, without which this thesis couldn't have been possible with many float-point numbers.

I'm deeply indebted to my parents, my parents-in-law, my sisters and my sister-in-law for their continuous support, endless patience and encouragements through all these years, especially the hard time during my study at Hopkins.

Finally, and most importantly, this thesis is for my dear wife, Jia Cui, and our son Colin. They make my life really unique, meaningful, enjoyable and complete.

To my family

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**
1.1 Bitext and Bitext Alignment 1
1.1.1 Automatic Evaluation of Bitext Alignment 4
1.2 Statistical Machine Translation 5
1.2.1 Translation Template Model 6
1.2.2 Automatic Translation Evaluation 7
1.3 Language Model 9
1.3.1 Evaluation of Language Models 10
1.4 Organization 11

I Models of Bitext Alignment **14**

2 Statistical Bitext Chunk Alignment Models **15**
2.1 Introduction 15
2.2 A Generative Model of Bitext Segmentation and Alignment 17
2.3 Chunk Alignment Search Algorithms 22
2.3.1 Monotonic Alignment of Bitext Segments 22
2.3.2 Divisive Clustering of Bitext Segments 24
2.3.3 A Hybrid Alignment Procedure 26
2.4 Summary 27

3 Statistical Word Alignment Models **29**
3.1 Introduction 29
3.2 Statistical Generative Word Alignment Models 31
3.2.1 HMM-based Statistical Word Alignment Models 32
3.2.2 IBM Models 33
3.2.3 Discussion 36

3.3 HMM-Based Word-to-Phrase Alignment Models . . . 37
3.3.1 Component Variables and Distributions . . . 37
3.3.2 Comparisons with other Models . . . 41
3.3.3 Embedded Model Parameter Estimation . . . 43
3.3.4 Deriving Word Alignments . . . 49
3.3.5 Discussion . . . 50
3.4 Summary . . . 51

4 Statistical Phrase Alignment Models 52
4.1 Introduction . . . 52
4.2 Word Alignment Induced Phrase Translation Models . . . 53
4.2.1 Word Alignments . . . 54
4.2.2 Phrase Pair Extraction . . . 54
4.2.3 Phrase Translation Table . . . 55
4.2.4 Discussion . . . 56
4.3 Model-based Phrase Pair Posterior . . . 56
4.3.1 IBM Model 1 & 2 . . . 58
4.3.2 IBM Fertility-based Models . . . 58
4.3.3 HMM-based Alignment Models . . . 58
4.4 PPI Induction Strategy . . . 59
4.5 Summary . . . 60

5 Experimental Results of Bitext Chunk Alignment 61
5.1 Corpora . . . 61
5.1.1 Chinese-English . . . 61
5.1.2 Arabic-English . . . 62
5.2 Unsupervised Bitext Sentence Alignment . . . 63
5.2.1 Monotonic Sentence Alignment Using Sentence Length Statistics 64
5.2.2 Iterative Alignment and Translation Table Refinement . . . 64
5.2.3 Length Distributions, Divisive Clustering, and Alignment Initialization . . . 66
5.2.4 Comparable Sentence Alignment Procedures and Performance Upper Bounds . . . 67
5.3 Evaluation via Statistical Machine Translation . . . 69
5.3.1 Bitext Chunking and Alignment . . . 70
5.3.2 Bitext Word Alignment and Translation Performance . . . 71
5.4 Maximizing the Aligned Bitext Available for Training . . . 73
5.5 Improved Subsentence Alignment Can Improve Word Alignment . . . 75
5.6 Translation Lexicon Induction . . . 78
5.7 Summary . . . 80

6 Experimental Results of Word and Phrase Alignment **81**
6.1 Data . . . 81
6.1.1 Chinese-English . . . 81
6.1.2 Arabic-English . . . 82
6.2 Chinese-English Bitext Word Alignment . . . 83
6.2.1 Alignment Evaluation . . . 83
6.2.2 Initial Experiments on FBIS Corpus . . . 84
6.2.3 Aligning Large Bitexts . . . 85
6.3 Translation Evaluation . . . 88
6.3.1 Chinese-English Translations . . . 88
6.3.2 Arabic-English Translations . . . 88
6.3.3 WtoP Model and Model-4 Comparison . . . 89
6.4 Effect of Language Model . . . 92
6.5 Summary . . . 94

II Language Modeling **95**

7 Latent Semantic Analysis **96**
7.1 Introduction . . . 96
7.2 Word-Document Frequency Matrix W . . . 98
7.3 Singular Value Decomposition of W . . . 99
7.4 Similarity Measurements . . . 101
7.5 Representing Pseudo-Documents . . . 102
7.6 Word Clustering Examples . . . 103
7.7 Summary . . . 104

8 Latent Semantic Analysis in Language Models **105**
8.1 Introduction . . . 105
8.2 Calculating Word-Probabilities Using LSA . . . 107
8.3 Combining LSA probabilities with N-grams . . . 108
8.4 Exponential Models with LSA Features . . . 109
8.4.1 A Similar ME Model from the Past . . . 112

9 Experimental Results of LSA-Based Language Models **113**
9.1 Corpus . . . 113
9.2 Perplexity: LSA + N-gram Models . . . 114
9.3 Effect of Replacing $\tilde{d}_{t-1}$ with $\hat{d}_{t-1}$ . . . 115
9.4 Perplexity: ME Model with LSA Features . . . 116
9.5 Word Error Rates for the ME Model . . . 117
9.5.1 Benefits of Dimensionality Reduction . . . 118
9.6 Summary . . . 119

III Conclusions and Future Work **120**

10 Conclusions and Future Work **121**
10.1 Thesis Summary 121
10.2 Suggestions for Future Work 123
10.2.1 Word-to-Phrase HMM 123
10.2.2 Phrase-to-Phrase HMM 124
10.2.3 Beyond String-to-String Alignments 124
10.2.4 Language Modeling 125

A HMM-based Phrase-to-Phrase Alignment Models **126**

Bibliography **129**

List of Figures

1.1 Chinese/English bitext aligned at sentence and word level: horizontal lines denote the segmentations of a sentence alignment and arrows denote a word-level mapping. 2
1.2 Bitext alignments for statistical machine translation 4
1.3 The Translation Template Model with Monotone Phrase Order. . . . 6

2.1 Bitext Chunking via. Dynamic Programming 24
2.2 An example of the Divisive Clustering procedure applied to Chinese-English Bitext. Each line with a number on it splits a chunk pair into two smaller chunk pairs. The numbers represent the sequence of divisive clustering: i.e., "1" means the first split, so on and so forth. . 26

3.1 An example of HMM-based word-to-word Alignment Models. The source string is a Chinese word sequence and the target string is an English word sequence. A Markov network is established by treating source words as Markov states and target words as output sequence. A target word is emitted one time after a state transition. 33
3.2 Simplified Example of Word-to-Phrase Alignments under IBM Model-4. Source word *fertility* determines how many target words are generated by each source word; word-to-word translation tables create the specified number of target words as translations of each source word; and the *distortion* model then specifies the order of the target words in the target sentence. This example presents the target sentence as a sequence of phrases, but there is no requirement that the target words generated by a source word should appear as neighbors in the target sentence. 35
3.3 Simplified Example of HMM-Based Word-to-Phrase Alignment Model. The source string is a Chinese word sequence and the target string is an English word sequence. A Markov network is established by treating source words as Markov states, with the state dependent observation distributions defined over phrases of target words. 42

3.4 Word-to-Word and Word-to-Phrase Links 43

4.1 Phrase pair extracting based on word alignments 55
4.2 An example of Model-4 word alignments showing that incorrect word alignments prevent perfect phrase pairs from being extracted. 57

5.1 Precision and Recall of Automatically Sentence Alignment Procedures Over the FBIS Sentence-Alignment Corpus with Different Initialization and Search Strategies. Alignment procedures 'F' and 'G' were initialized from iteration 0 of the DP+DC($\lambda = 3.0$, $\alpha = 0.9$) alignment procedure. 68
5.2 Precision of induced IBM Model 1 lexicons measured against the LDC Chinese-to-English bilingual dictionary. Each curve is associated with a single alignment of the bitext. DP+DC algorithm is applied to 100, 300, 500 and all document pairs from FBIS Chinese/English parallel corpus. From each set of alignments eight iterations of EM are used to induce an IBM Model 1 lexicon. Each curve is obtained by pruning the lexicon by a sequence of thresholds on the translation probability. Each point on each curve represents a pruned lexicon. The precision of each of these is plotted versus its number of entries. 79

6.1 Balancing Word and Phrase Alignments. 85

7.1 Singular Value Decomposition of the Sparse Matrix W. 100

9.1 $K(w_t, \hat{d}_{t-1})$ and $K(w_t, \tilde{d}_{t-1})$ through a conversation (.9513.6$_{\text{TOP}}$), $K(w_t, \hat{d}_{t-1}) - K(w_t, \tilde{d}_{t-1})$ (.9513.6$_{\text{MIDDLE}}$), and $K(w_t, \hat{d}_T) - K(w_t, \hat{d}_{t-1})$ (.9513.6$_{\text{BOTTOM}}$). 116

List of Tables

3.1 An example showing that bigram translation probabilities may assign higher likelihood to correct phrase translations than unigram probabilities by utilizing word context information within a phrase. 50

5.1 Statistics of Chinese-English Parallel (document pairs) Corpora . . . 62
5.2 Statistics of NIST Chinese-English MT Evaluation Sets 62
5.3 Statistics of Arabic-English Parallel (document pairs) Corpora 63
5.4 Statistics of NIST Arabic-English MT Evaluation Sets 63
5.5 Bitext used at Each Iteration of Unsupervised Sentence Alignment. At Iteration 0, the entire 122 document bitext is used. At iterations 1 through 4 the chunk pairs found at the previous iteration are sorted by likelihood and only those with likelihoods above the specified threshold are retained for the estimation of the Model-1 translation table. . . . 65
5.6 Performance of Sentence Alignment Procedures Over the FBIS Sentence-Alignment Corpus. Procedures *a, b, c* are unsupervised; Champollion is provided with a Chinese-English translation lexicon; the 'Oracle' version of DP+DC uses Model-1 translation tables trained over the human-aligned sentences. 67
5.7 Aligned Chunk Pair Statistics Over Contrastive Alignment Configurations. Step 1: initial chunk alignments obtained by DP monotone alignment using sentence length statistics. Step 2: divisive clustering of aligned chunks from Step 1 under sentence-length statistics. The aligned chunks at Step 2 are used in training a Model-1 translation table; this table is held fixed for Steps 3 and 4. Step 3: chunk alignments obtained by DP monotone alignment using Model-1 translation table. Step 4: divisive clustering of aligned chunks from Step 1 under Model-1 translation table. 71

5.8 Word Alignment and Translation Performance Corresponding to IBM-4 Models Estimated over Bitext Collections Produced by Contrastive Alignment Configurations. Alignment Error Rates are provided in both translation directions. Translation performance is given as BLEU(%) scores of phrase-based SMT systems based on phrases extracted from the word aligned bitext. 72
5.9 Percentage of Usable Arabic-English Bitext. English tokens for Arabic-English news and UN parallel corpora under different alignment procedures. 74
5.10 Translation Performance of TTM Arabic-English Systems Based on Bitext Collections Extracted by the Alignment Procedures. 74
5.11 Influence of Subsentence Alignment on Alignment Error Rate 76
5.12 English-to-Chinese Word Alignment Links Accuracy Relative to Chunk Alignment Boundaries Found by Divisive Clustering 77

6.1 Statistics of Chinese-English Parallel (chunk pairs) Corpora 82
6.2 Statistics of Arabic-English Parallel (chunk pairs) Corpora 82
6.3 FBIS Bitext Alignment Error Rate. 84
6.4 AER Over Large C-E Bitexts. 86
6.5 Chinese→English Translation Analysis and Performance of Viterbi PPI Extraction (V-PE) and WtoP Posterior Induction Procedures 89
6.6 Arabic→English Translation Analysis and Performance of Viterbi PPI Extraction (V-PE) and WtoP Posterior Induction Procedures 90
6.7 Translation results on the merged test sets 91
6.8 The number of English text (in millions) used to train language models. 92
6.9 Language model effects on the large Arabic-English translation system measured by BLEU score. 93

9.1 Perplexities: N-gram + LSA Combination 115
9.2 Perplexities: Maximum Entropy Models 117
9.3 Error Rates: Maximum Entropy Models 118
9.4 A comparison between the model (A) of Khudanpur and Wu [35] and our model (B). 118

Chapter 1

Introduction

1.1 Bitext and Bitext Alignment

Bilingual text, or bitext, is a collection of text in two different languages. Bitext alignment is the task of finding translation equivalences within bitext. Depending on the granularity of parts to be aligned, bitext alignment can be performed at different levels. The coarsest implementation is at the document level [5], where document pairs that are translations of one another need to be identified from a collection of bilingual documents. A slightly finer problem is to align bitext at the sentence level [9, 26, 20], while the most complex, finest problems take place at the word level [7, 70, 68], where matching words between sentence pairs must be identified. Phrase alignment [57] [94] falls between word and sentence alignments, but it is usually resolved subsequent to word alignment. Figure 1.1 shows parallel Chinese/English bitext that is aligned at sentence and word levels: horizontal lines denote the segmentations of sentence alignment, and arrows denote word-level mapping. These are string-to-string alignments, though there are still other types of bitext alignment realization, for instance, string-to-tree alignments [88] that can be exploited in bilingual parsing and syntax-based machine translation.

Generally speaking, the finer the basic units to be aligned within bitext, the more sophisticated linguistic resources or statistical models required to obtain reasonable alignments. While statistics of the number of words [9] or even the number of charac-

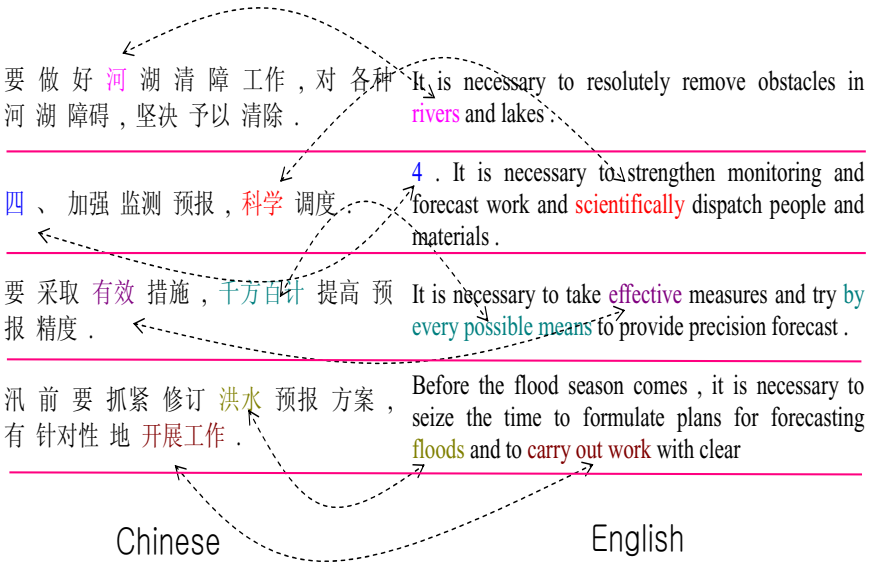


Figure 1.1: Chinese/English bitext aligned at sentence and word level: horizontal lines denote the segmentations of a sentence alignment and arrows denote a word-level mapping.

ters in sentences [26] seem good enough to achieve a decent performance in recognizing sentence-to-sentence correspondences within parallel documents, much more complicated statistical models [81] [7] are necessary to produce word alignments with good quality. This can be understood by the fact that higher complexity of unit order change during translation can usually be observed for units at finer granularity. For instance, when translating the English Bible into French, chapters or sections are usually kept in the same order, while for any given English sentence, more flexibility is allowed to choose French words and their order.

The use of bitext alignment is critical for many multilingual corpus-based Natural Language Processing (NLP) applications. To give some examples: sentence pairs with words aligned offer precious resources for work in bilingual lexicography[36]; in Statistical Machine Translation (SMT) [7], bitexts aligned at sentence level are the basic ingredients in building an Machine Translation (MT) system. There, better alignment in the first stage can lead to better performance of the learning components in extracting useful structural information and more reliable statistical parameters. With the increasing availability of parallel corpora, human alignments are expensive and

often unaffordable for practical systems, even on a small scale. In many applications, not only the alignments but also alignment models are needed.

Given these conditions, and the increased availability of parallel text, high performance automatic bitext alignment has become indispensable. In addition to its quality, several other properties make automatic alignment desirable. Insofar as they are more general and language independent, the procedure and model of automatic alignment render it better able to process widely discrepant languages, such as French, Chinese and Arabic, to give a few examples. The model parameters are better estimated from scratch, statistically, in an unsupervised manner from bitext. To process huge amounts of bitext, say millions of sentence pairs, models need to be effective and algorithms should be efficient. Finally, since noise and mismatch are often presented in real data, for example, parallel corpora mined from web pages, automatic bitext alignment needs to be robust.

This thesis shall address string-to-string bitext alignment at different granularities from coarse to finer: from sentence level to word level, including sub-sentence and phrases between them. We build statistical models and establish links between units at different levels with a special focus on application of statistical machine translation. Models and alignments are directly applicable for other multi-lingual tasks, for instance, statistical bilingual parsing, translation lexicon induction, cross lingual information retrieval [64] and language modeling [36].

As Figure 1.2 shows, we start with parallel documents. In the preprocessing stage, documents are tokenized or segmented into space-delimited token sequences. The bitext chunking module derives sentence pairs as training material for MT training components to build statistical word and phrase alignment models. The Translation Template Model (TTM) decoder (see section 1.2.1) takes phrase translation models and language models as input and translates English sentences into foreign hypotheses, and finally the MT evaluation component gives out translation performances by comparing hypotheses against human translation references.

The goal of this thesis is to improve bitext alignment for better translation performance. We propose statistical bitext alignment and translation models and investigate their usefulness and contributions in building a SMT system.

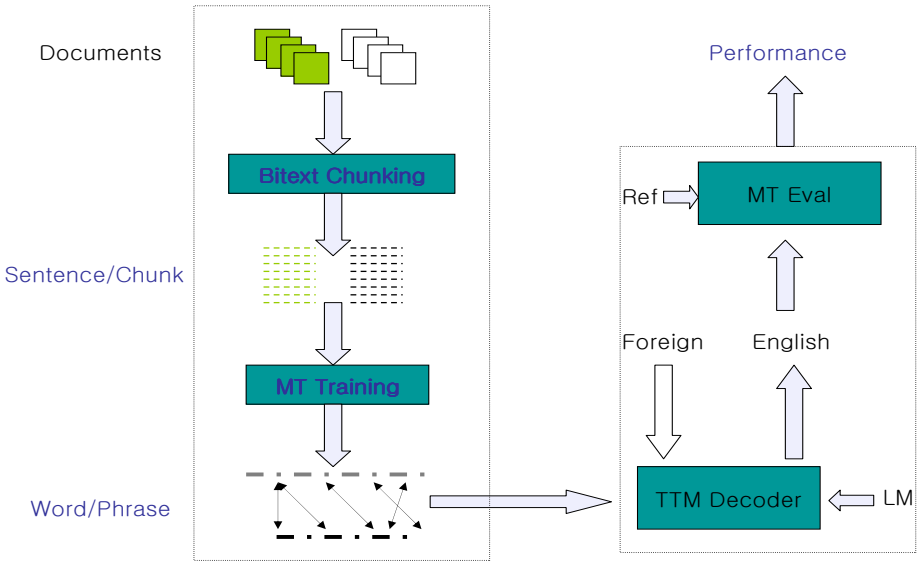


Figure 1.2: Bitext alignments for statistical machine translation

1.1.1 Automatic Evaluation of Bitext Alignment

Although our primary goal is to improve machine translation system performance, automatic evaluation of bitext alignment at earlier stages can be quite useful as well as telling. The automatic metric is defined from the point of view of information retrieval. A machine produces B', a bag of links between units, and a human provides B, a bag of reference links. The unit could be a sentence in the sentence alignment task or a word in the bitext word alignment task. We then define the indices of precision and recall, and the Alignment Error Rate (AER) [70] is defined as the complement of the so-called F-measure where precision and recall are weighted equally.

$$\text{Precision}(B, B') = \frac{|B \cap B'|}{|B'|} \tag{1.1}$$

$$\text{Recall}(B, B') = \frac{|B \cap B'|}{|B|} \tag{1.2}$$

$$\text{AER}(B, B') = 1 - \frac{2 \times |B \cap B'|}{|B'| + |B|} \tag{1.3}$$

1.2 Statistical Machine Translation

Statistical machine translation [7] [39] has achieved significant advancement in recent years. This is attributed to increased availability of parallel corpora and the progress of statistical modeling and automatic evaluation [39]. The most widely used model in statistical MT systems is the source-channel model [7]. The source string, say an English sentence $\mathbf{s}$, goes through a stochastic noisy channel and generates the target string, say a foreign sentence $\mathbf{t}$. It typically includes two components: a monolingual language model $P(\mathbf{s})$, which assigns probabilities to source language strings, and a translation model $P(\mathbf{t}|\mathbf{s})$ that assigns probabilities to target language strings given a source string. Bilingual sentence pairs are required to learn the statistical parameters of the translation model, and the translation process is usually implemented by source decoding algorithms, for instance, Maximum A Posterior (MAP) $\hat{\mathbf{s}} = \text{argmax}_{\mathbf{s}} P(\mathbf{s})P(\mathbf{t}|\mathbf{s})$.

Translation can be carried out based on word identity [7] [28]. Foreign words are translated into English words, and English words are reordered to produce sound sentences. Translation performance can be improved, though, when based on phrases [68] [40]. For instance, the foreign sentence is segmented into foreign phrases, and each foreign phrase is translated into English phrases, and finally English phrases are moved around with reorder models to produce the output hypothesis.

Publically available decoders include the ReWrite decoder [1], which is a word-based translation system, and the Pharaoh [2], which is a beam search decoder based on phrase-based statistical machine translation models, and the Translation Template Model (TTM) decoder [42], which is a phrase-based, weighted finite state implementation of the source-channel translation template model.

We next briefly introduce the Translation Template Model (TTM) [42]. In this thesis, all translation experiments are conducted via the TTM decoder as shown in Figure 1.2.

[1] http://www.isi.edu/licensed-sw/rewrite-decoder/
[2] http://www.isi.edu/licensed-sw/pharaoh/

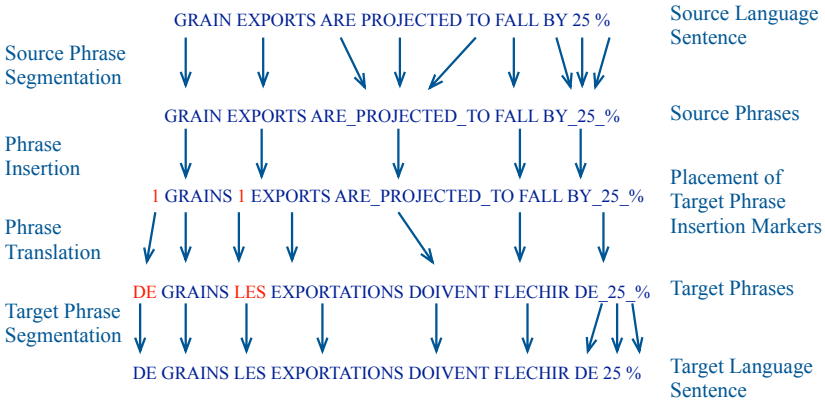


Figure 1.3: The Translation Template Model with Monotone Phrase Order.

1.2.1 Translation Template Model

The Translation Template Model (TTM) [42] is a source-channel model of translation with joint probability distribution over all possible segmentations and alignments of target language sentences and their translations in the source language. Translation is modeled as a mapping of source language phrase sequences to target language sentences. The model considers whole phrases rather than words as the basis for translation.

The translation process underlying the model is presented in Figure 1.3. The source sentence is segmented into source phrases; source phrases are mapped onto target phrases, which form the target sentence naturally. Target phrases are allowed to be inserted in the generative process. This corresponds to the deletion of target phrases during translation. Translation is in monotone phrase order.

Each of the conditional distributions that make up the model is realized independently and implemented as a weighted finite state acceptor or transducer. Translation of sentences under the TTM can be performed using standard Weighted Finite State Transduce (WFST) operations involving these transducers.

1.2.2 Automatic Translation Evaluation

Just as machine translation is a hard problem [38], so too is the evaluation of automatic translation. One of the reasons for this complexity is that there are multiple ways of translating a given sentence. This diversity and the word order issue sometimes pose challenges even for human judgement.

Human evaluations [82] [30] are expressed by at least two categories: adequacy, which captures how meanings in original sentences are preserved in translations, and fluency, which evaluates the correctness of grammar in translations. While adequacy requires knowledge of the original language, fluency expects proficiency at the level of a native speaker in the language into which it is being translated. A complete human evaluation of machine translation requires bilingual expertise. Consequently, the procedure can be very expensive and time consuming and makes it extremely inconvenient for development and diagnostic of machine translation systems.

Although automatic translation evaluation is a debatable problem, it is valuable in significantly accelerating MT system development and enabling experiments with many models and algorithms that might otherwise not be tested. The goal of automatic translation evaluation is to define metrics that correlate well with human judgement. It has become an active topic recently, and many evaluation criteria have been developed, for instance, BLEU-score [72], NIST-score[21], F-measure[60], multi-reference Position-independent Word Error Rate (mPER) [65], multi-reference Word Error Rate (mWER) [65] and Translation Error Rate (TER) [25]. Each criterion assumes that human references exist for machine-generated translations against which to be compared. There are also automatic metrics that do not require human references [27]. It is unlikely that any one of these metrics would perform better than the others for all translation tasks. This thesis has no intention of proposing an alternative evaluation function. Next, we will briefly introduce BLEU and TER.

BLEU metric

BLEU [72] is an automatic machine translation evaluation metric that has been widely recognized in the research community. It was adopted in NIST MT evaluation

[63] from 2002 to 2005 and has been found to correlate highly with human judgements in terms of fluency and adequacy. In this thesis, we mostly report translation system performance by BLEU score.

BLEU is defined as the geometric mean of n-gram precision weighted by an exponential brevity penalty factor. Given a translation candidate and the reference translation(s), let p_n be the n-gram precision, $n = 1, 2, \cdots, N$, and w_n be their positive weights summing to one. The geometric mean (GM) of n-gram precision is calculated as

$$GM = e^{\sum_{n=1}^{N} w_n \log p_n}.$$

Brevity penalty (BP) is calculated on the whole corpus rather than sentence by sentence. Let c be the length of translation hypothesis. For each test sentence, the best match length in the reference translation is found; Let their sum be the test corpus' effective reference length r. The brevity penalty is computed in this way:

$$BP = \begin{cases} 1 & c > r \\ e^{1-\frac{r}{c}} & c \leq r \end{cases}$$

Then BLEU score is defined as:

$$BLEU = GW \cdot BP = exp\Big(\sum_{n=1}^{N} w_n \log p_n + \min\{0, 1 - \frac{r}{c}\}\Big)$$

Usually, n-grams up to $N = 4$ are considered and weighted equally, namely $w_n = \frac{1}{N}$. Note that the more reference translations there are, the higher BLEU is. The metric is normalized between 0 and 1, and a higher score implies a better translation.

To give an example, let the reference translation be "Mr. Chairman , in absolutely no way ." and the hypothesis be "in absolutely no way , Mr. Speaker". We find the n-gram precisions are $\frac{7}{8}$, $\frac{3}{7}$, $\frac{2}{6}$, $\frac{1}{5}$ for $n = 1, 2, 3, 4$ respectively. Since the brevity penalty is 1, the BLEU score is just the geometric mean of n-gram precision $(\frac{7}{8} \times \frac{3}{7} \times \frac{2}{6} \times \frac{1}{5})^{\frac{1}{4}} = 0.3976$.

Translation Error Rate

Translation Error Rate (TER) [25] is a post-editing scoring metric for automatic machine translation evaluation. TER measures the amount of editing required to

transform a hypothesis into a human reference. The machine output is first operated by one or more shift operations, which moves a sequence of words within the hypothesis, and then it is compared against the human reference to calculate the number of insertions, deletions and substitutions using dynamic programming. All edits including the shift operation count as 1 error. When there are multiple references, TER is calculated against the closest one. It is defined as the number of edits over the average number of reference words.

TER is proposed for a better intuitive understanding of translation evaluation. Unlike the BLEU score, a lower TER score suggests a better translation performance. It has been found that TER has a higher correlation when used with BLEU.

1.3 Language Model

In the source-channel model, a monolingual language model is used to describe the source distribution with probability. The language model is one of the most crucial components in Statistical Machine Translation [7] and Automatic Speech Recognition [33]. It has also been applied in Information Retrieval [93], Language Identification [90], Spelling Correction [34], Optical Character Recognition (OCR) [41] and other applications.

From a statistics point of view, the language model assigns probability to a monolingual word sequence. Let $\mathbf{W} = w_1^I$ be a word sequence, and applying the chain rule:

$$P(\mathbf{W}) = P(w_1)P(w_2|w_1)\prod_{i=3}^{I} P(w_i|w_1, w_2, \cdots, w_{i-1}) \tag{1.4}$$

The problem is then transformed into a predictive task: given all the history from w_1 to w_{i-1}, how are we to make a prediction on the following word w_i probabilistically; i.e., what is the probability distribution of the next word? A natural choice would be the relative frequency in the training data, which is the *Maximum Likelihood Estimation* (MLE) without smoothing. When the history gets longer, though, the

prediction becomes harder to manage: most of the events are never seen in the training data even with a history of two words. This is the data sparseness problem. Clustering the history reduces the number of parameters need to estimate. Histories which have been grouped together predict the following word with the same distribution.

The simplest clustering of histories keeps the most recent $N-1$ words and ignores the others, and the corresponding language model is termed the *N-gram* model.

$$P(w_i|w_1, w_2, \cdots, w_{i-1}) = P(w_i|w_{i-1}, w_{i-2}, \cdots, w_{i-N+1}) \tag{1.5}$$

When $N = 1, 2, 3$, this is called the *unigram*, *bigram* and *trigram* language model, respectively.

Smoothing techniques have been found both extremely necessary and effective in general language modeling. The goal is to avoid zero probability for unseen events by redistributing probability mass. For the n-gram language model, modified Kneser-Ney smoothing [37] [11] generally performs well in speech recognition tasks. More smoothing techniques, i.e., Witten-Bell smoothing [83] and their comparisons can be found in [11].

1.3.1 Evaluation of Language Models

Statistical language models are usually estimated from a collection of monolingual texts. One means of automatic evaluation of language models is to measure the *Perplexity* (PPL) on a test set. Let L be a language model to be evaluated. Let $\mathbf{W} = w_1 w_2 \cdots w_K$ be the test word sequence. Perplexity is defined as the exponential of the negative average likelihood of predictions on the test set, which is also the exponential of the *cross entropy* of the language model on the test set.

$$PPL = \exp\Big\{ -\frac{1}{K} \log P_L(w_1 w_2 \cdots w_K) \Big\} \tag{1.6}$$

$$= \exp\Big\{ -\frac{1}{K} \sum_{k=1}^{K} \log P_L(w_k|w_1 w_2 \cdots w_{k-1}) \Big\} \tag{1.7}$$

Perplexity characterizes how closely the language model matches the test set. A lower perplexity indicates a closer match. Although the perplexity value is indicative

of the langauge model's performance in actual applications, it is necessary to evaluate the language model's usefulness in an ultimate system. For statistical machine translation, the language model can be judged by the performance of translation systems, for instance, BLEU score. In automatic speech recognition tasks, the language model is measured by the Word Error Rate (WER) of the speech decoder.

A speech decoder transcribes an acoustic observation sequence into a hypothesis, a word sequence. The number of errors is defined as the edit distance (found via a dynamic programming procedure) required to transform the hypothesis into the reference, the human transcription. The WER is defined as the number of errors (including deletion, substitution and insertion) divided by the number of reference words. For example, let "HELLO WHAT'S THE DATE ?" be the reference, and "HELLO WHAT'S DAY ?" be the ASR output; then the WER is 40% (one deletion and one substitution error over five reference words).

1.4 Organization

This thesis consists of three parts. In Part I, we discuss statistical bitext alignment models at different granularity for statistical machine translation. In Part II, we review language model techniques with a focus on incorporating latent semantic information into statistical language models. We conclude in the last part.

In Chapter 2, we present a statistical bitext chunk alignment model to perform alignments at sentence or sub-sentence level. Basic model components are presented, and two very different chunking procedures are derived naturally within the same statistical framework. One is the widely used dynamic programming algorithm; the other is the divisive clustering algorithm based on the divide and conquer technique, which derives short chunk pairs in hierarchial way. A hybrid approach is presented after the comparison of the two algorithms.

In Chapter 3, we discuss statistical word alignment models. We review the HMM-based word-to-word alignment model and the series of IBM models, comparing them and identifying their strengths and weaknesses. With the goal of building on the strengths of both models, we present the HMM-based word-to-phrase alignment mod-

els. We formally present basic model components, discuss model parameter training and address smoothing issues in robust estimation. We show how Viterbi word alignments can be derived and discuss several model refinements.

We address phrase alignment in Chapter 4. Word alignment induced statistical phrase translation models are presented with a focus on extraction of Phrase Pair Inventory (PPI). We present a model-based phrase pair posterior distribution which replies on multiple word alignments, not the one-best word alignments, in aligning phrases to phrases. We point out that the definition of posterior distribution can be applied to any statistical word alignment model, showing its implementation under HMM-based and IBM serial models. We propose a simple PPI augmenting scheme using the posterior distribution with the goal of improving phrase coverage on test sets.

Experimental results of bitext chunk alignment are presented in Chapter 5. We show the unsupervised sentence alignment performance of different chunking alignment algorithms. Next, we evaluate the effects of chunking procedures in the tasks of bitext word alignment and machine translation evaluation, demonstrating the practical advantages of divisive clustering in maximizing the aligned bitext available for deriving MT training material. We also show how chunking at the sub-sentence level can improve word alignment quality. Finally, we offer a simple translation lexicon induction procedure using bitext chunking approaches.

The evaluation of statistical word and phrase alignment models is performed, and the results of the experiment presented, in Chapter 6. There, we compare the performance of HMM-based models and IBM Models in word alignment tasks and translation evaluation. We illustrate the translation results of different PPI induction schemes in Chinese-English and Arabic-English translation tasks.

In the second part of the thesis, we discusses language modeling techniques. We begin with the introduction of Latent Semantic Analysis in Chapter 7, discussing procedures of extracting meaningful and compact representations of words and documents by collecting word-document co-occurrence statistics and applying Singular Value Decomposition (SVD) techniques.

Chapter 8 discusses applications of LSA in statistical language modeling tech-

niques. We induce LSA probabilities from similarity distributions and their combination with the n-gram language model. We then present a novel integration of LSA features and local n-gram features under the Maximum-Entropy framework. In Chapter 9, we present experimental results of how LSA-based language models perform the task of perplexity evaluation and conversational speech recognition.

We conclude in Chapter 10 by highlighting the contributions of the thesis and suggesting possible extensions of its research.

Part I

Models of Bitext Alignment

Chapter 2

Statistical Bitext Chunk Alignment Models

2.1 Introduction

Bitext corpora play an important role in the development of statistical Machine Translation (MT) systems [7]. A typical training scenario for a translation system starts with a collection of paired sentence translations in the languages of interest. Model-based estimation techniques extract from these bitext translation lexicons, word-to-word alignments, phrase translation pairs, and other information that is then incorporated into the translation system. Although it is a crucial first step in such training procedures, bitext sentence alignment is often considered as a separate modeling problem along with other practical concerns such as text normalization.

We discusses a modeling approach that is a first step towards a complete statistical translation model that incorporates bitext alignment. Our goal is to present a general statistical model of a large scale bitext chunk alignment within parallel documents and to develop an iterative language independent chunking procedure when no linguistic knowledge is available. We evaluate our model in the context of statistical machine translation.

Extracting chunk pairs is an alignment problem that falls somewhere between word alignment and sentence alignment. It incorporates and extends well-established

techniques for bitext sentence alignment, with the aim of aligning text at the subsentence level. There are two practical benefits to be had from doing this. Shorter bitext segments can lead to quicker training of MT systems since MT training tends to run faster on shorter sentences. Also, the ability to break down very long sentences into smaller segments will make it possible to train with text that would otherwise have to be discarded prior to training. While these may seem like mainly practical concerns, fast training and thorough exploitation of all available bitext are crucial for effective system development and overall performance. Beyond these practical concerns, we also provide evidence that word alignments achieved over chunks pairs aligned at the subsentence level produce better results than word alignment over sentence pairs.

Many approaches have been proposed to align sentence pairs in bitext. One widely used method is a dynamic programming procedure based on sentence length statistics [9, 26]. By bootstrapping from sentences aligned by hand and incorporating word translation probabilities, Chen [10] developed a method that improved alignment performance. Wu [84] extended the length-based method proposed by Gale and Church [26] to non-Indo-European languages by taking advantage of pre-defined domain specific word correspondences. To reduce reliability on prior knowledge about languages and improve robustness to different domains, Haruno and Yamazaki [29] iteratively acquired word correspondences during the alignment process with the help of a general bilingual dictionary. Melamed [58] developed a geometric approach to alignment based on word correspondences. Typically, there are two desirable properties of sentence alignment: the alignment procedure should be robust to variable quality bitext and the resulting alignment should be accurate. While accuracy is usually achieved by incorporating lexical cues, robustness can be addressed by bootstrapping with multi-pass search [76, 62], where those sentence pairs with "high quality" are identified initially as seeds and successive refined models are built and applied to discover more pairs in the whole corpora. There are, of course, many applications in NLP that rely on aligned bitext, including statistical bilingual parsing, translation lexicon induction, cross lingual information retrieval [64], and language modeling [36].

In this work, we develop a generative model of bitext chunk alignment that can

be used to extract and align chunks in bitext. Within this framework, two alignment algorithms are derived in a straightforward manner. One is a dynamic programming based procedure similar to those mentioned above. The second algorithm is a divisive clustering approach to bitext alignment that begins by finding coarse alignments that are then iteratively refined by successive binary splitting. Both of these algorithms are derived as maximum likelihood search procedures that arise due to variations in the formulation of the underlying model. This is certainly not the first application of binary search in translation; binary descriptions of sentence structure for translation were explored by Wu[85, 86]. However, our approach is intended to be much simpler (and less descriptive) and is developed for very different purposes, namely for preparing bitext for model-based SMT parameter estimation. Our interest here is not in the simultaneous parsing of the two languages. We are simply interested in a general procedure that relies on raw lexical statistics rather than a complex grammar. We sacrifice descriptive power to gain simplicity and the ability to align large amounts of bitext.

In the following sections we will introduce the model and derive the two alignment procedures. In Chapter 5, we will discuss their application to bitext alignment and measure their performance by their direct influence on MT evaluation performance as well as through indirect studies of the quality of induced translation lexicons and alignment error rates of the trained MT systems.

2.2 A Generative Model of Bitext Segmentation and Alignment

We begin with some definitions. A document is divided into a sequence of *segments* which are delimited by *boundary markers* identified within the text. The definition of the boundary markers will vary depending on the alignment task. Coarse segmentation results when boundary markers are defined at sentence or possibly even paragraph boundaries. Finer segmentation results from taking all punctuation marks as boundary points, in which case it is possible for a document to be divided into sub-

sentential segments. However, the process is deterministic. Once boundary markers are identified within a document, the segments are specified and are indivisible.

A *chunk* consists of one or more successive segments. Depending on how the segments are defined by the boundary marks, chunks can be formed of multiple sentences, single sentences, or even phrases or single words; the concept is generic. It is these chunks that are to be aligned as possible translations. If two chunks are hypothesized in the bitext as possible translations, they are considered to be a *chunk pair*. In this way, document alignment is performed by a deterministic segmentation of the documents, followed by joint chunking and alignment of the segments. We now describe the random variables and the underlying distributions involved in this alignment model.

Alignment Variables The parallel text to be chunk aligned has n segments in the target language (say, Chinese) $\mathbf{t} = \mathbf{t}_1^n$ and m segments in the source language (say, English) $\mathbf{s} = \mathbf{s}_1^m$. Note that each $\mathbf{t}_j$ and $\mathbf{s}_i$ is a segment, which is to say a string that cannot be further broken down by the aligner.

To describe an alignment between the documents $\mathbf{t}$ and $\mathbf{s}$, we introduce a (hidden) chunk alignment variable $a_1^K(m, n)$ which specifies the alignment of the chunks within the documents. The alignment process is defined and constrained as follows:

A1 The parallel text has m segments in source string $\mathbf{s}$ and n segments in target string $\mathbf{t}$.

A2 $(\mathbf{s}, \mathbf{t})$ is divided into K chunk pairs; K is a random variable.

A3 For each $k = 1, 2, \cdots, K$, a_k is a 4-tuple $a_k = (a[k].ss, a[k].se, a[k].ts, a[k].te)$.

$a[k].ss$ identifies the starting index on the source side, and $a[k].se$ identifies the final index on the the source side; $a[k].ts$ and $a[k].te$ play the same role on the target side. For convenience, we introduce $a[k].slen = a[k].se - a[k].ss + 1$ and $a[k].tlen = a[k].te - a[k].ts + 1$ which define the number of segments on each side of the chunk pair.

A4 There are boundary constraints on the chunk alignments:
$a[1].ss = a[1].ts = 1, a[K].se = m, a[K].te = n$

A5 There are continuity constraints on the chunk alignments:
$a[k].ss = a[k-1].se + 1, a[k].ts = a[k-1].te + 1, \quad k = 2, \cdots, K.$

We note that the above conditions require that chunks be paired sequentially; this will be relaxed later. Consequently, once $\mathbf{s}_1^m$ and $\mathbf{t}_1^n$ are each divided into K chunks, the alignment between them is fixed. We use a_1^K as a shorthand for $a_1^K(m,n)$ since we are considering known documents with m and n segments on source and target sides, respectively. Under a given alignment and segmentation, $\mathbf{s}_{a_k}$ and $\mathbf{t}_{a_k}$ denote the k^{th} chunks in the source and target documents respectively, i.e. $\mathbf{s}_1^n = \mathbf{s}_{a_1} \ldots \mathbf{s}_{a_K}$ and $\mathbf{t}_1^n = \mathbf{t}_{a_1} \ldots \mathbf{t}_{a_K}$.

Generative Chunk Alignment Model The conditional probability of generating $\mathbf{t}$ given $\mathbf{s}$ is

$$P(\mathbf{t}_1^n|\mathbf{s}_1^m) = \sum_{K,a_1^K} P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) \tag{2.1}$$

and by Bayes Rule, we have

$$P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) = P(n|\mathbf{s}_1^m)P(K|\mathbf{s}_1^m, n)P(a_1^K|\mathbf{s}_1^m, n, K)P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K)\,. \tag{2.2}$$

This defines the component distributions of the alignment model, as well as their underlying dependencies. We explain these component models in detail, pointing out the simplifying assumptions involved in each.

Source Segmentation Model $P(n|\mathbf{s}_1^m)$ is the probability that the source string generates a target language document with n segments. This is a component distribution, but it is not needed for alignment since the bitext segments are determined by the boundary markers within the text to be aligned.

Chunk Count Model $P(K|\mathbf{s}_1^m, n)$ is the probability that there are K chunks when $\mathbf{s}_1^m$ is paired with n segments of the target string. We ignore the words of the string $\mathbf{s}_1^m$ and assume K depends only on m and n: $P(K|\mathbf{s}_1^m, n) \equiv \beta(K|m,n)$.

Chunk Alignment Sequence Model We make two assumptions in the alignment process distribution $P(a_1^K|\mathbf{s}_1^m, n, K)$:

(a) the chunk alignment a_1^K is independent of the source words, and

(b) the chunk pairs are independent of each other, i.e., each target segment depends only on the source segment to which it is aligned.

With these assumptions, we have $P(a_1^K|\mathbf{s}_1^m, n, K) = \frac{1}{Z_{m,n,K}} \prod_{k=1}^K p(a_k)$ with the normalization constant $Z_{m,n,K} = \sum_{a_1^K} \prod_{k=1}^K p(a_k)$.

There are many possibilities in defining the alignment distribution $p(a_k)$. One form that we study specifies the range of segment lengths that will be allowed in chunk alignment. If $x = a[k].slen$ and $y = a[k].tlen$, then

$$p(a_k) = p(x,y) = \begin{cases} \frac{1}{g_{\lambda,\alpha}} e^{-\lambda(\alpha(x+y)+(1-\alpha)|x-y|)} & 1 \le x, y \le R \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

where $\lambda \ge 0$ and $1 \le \alpha \le 1$; R specifies the maximum number of segments that can be incorporated into a chunk. In previous work, this distribution over lengths has been tabulated explicitly (e.g. Table 1, [84]); we use a parameterized form mainly for convenience. Setting $\alpha = 0$ favors chunk alignments of equal segment lengths, while $\alpha = 1$ prefers shorter length segments. Setting $\lambda = 0$ specifies a uniform distribution over the allowed lengths.

Target Sequence Model $P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K)$ is the probability of generating the target string given the source string and the chunk alignment. We derive this probability from a word translation model with an independence assumption similar to (**b**) of the alignment model:

$$P(\mathbf{t}_1^n|\mathbf{s}_1^m, n, K, a_1^K) = \prod_{k=1}^K P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}). \tag{2.4}$$

The chunk-to-chunk translation probability is derived from a simple word translation model. With s_1^v and t_1^u denoting the word sequences for the chunk $\mathbf{s}_{a_k}$ and $\mathbf{t}_{a_k}$, we use

Model-1 [7] translation probabilities to assign likelihood to the translated segments:

$$P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}) = P(t_1^u|s_1^v, u)\, P(u|s_1^v) = \frac{P(u|v)}{(v+1)^u} \prod_{j=1}^{u} \sum_{i=0}^{v} t(t_j|s_i)\ . \tag{2.5}$$

$t(t_j|s_i)$ is the probability of source word s_i being translated into target word t_j; s_0 is a NULL word. Other formulations are, of course, possible. However, Model-1 treats translations as unordered documents within which any word in the target string can be generated as a translation of any source word, and this is consistent with the lack of structure within our model below the segment level. Model-1 likelihoods are easily computed and the Expectation-Maximization (EM) algorithm can be used to estimate those translation probabilities from a collection of sentence pairs [7].

The remaining component distribution $P(u|v)$ is the probability that the v words in a source string generate a target string of u words. We follow Gale and Church's [26] model and assume $u - cv$ is normally distributed.

$$\frac{u - cv}{\sqrt{v\sigma^2}} \sim \mathcal{N}(0, 1) \tag{2.6}$$

where the scalar c is the global length ratio between target language and source language. $P(u|v)$ can be calculated by integrating a standard normal distribution accordingly.

Summary We have presented a statistical generative chunk alignment models based on word-to-word translation model. The process to generate a target document $\mathbf{t}_1^n$ from $\mathbf{s}_1^m$ proceeds along the following steps:

1. Choose the number of source language segments n according to probability distribution $P(n|\mathbf{s}_1^m)$.

2. Choose the number of chunk pairs K according to probability distribution $\beta(K|m, n)$.

3. Choose chunk alignment a_1^K according to probability distribution $P(a_1^K|m, n, K)$

4. For each $k = 1, 2, \cdots, K$, produce $\mathbf{t}_{a_k}$ from $\mathbf{s}_{a_k}$via the word-to-word translation probability $P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k})$.

The above steps are, of course, only conceptual. In alignment, we have both source and target documents, and we search for the best hidden alignment sequence under the model.

2.3 Chunk Alignment Search Algorithms

We now address the problem of bitext alignment under the model presented in the previous section. We assume we have a bitext aligned at the document level and define a set of boundary markers which uniquely segment the documents into segment sequences $(\mathbf{s}_1^m, \mathbf{t}_1^n)$. The goal is to find the optimal alignment of these segments under the model-based Maximum A Posteriori (MAP) criterion:

$$\{\hat{K}, \hat{a}_1^{\hat{K}}\} = \underset{K, a_1^K}{\operatorname{argmax}} P(K, a_1^K | \mathbf{s}_1^m, \mathbf{t}_1^n) = \underset{K, a_1^K}{\operatorname{argmax}} P(\mathbf{t}_1^n, K, a_1^K | \mathbf{s}_1^m)\,. \tag{2.7}$$

We consider two different alignment strategies. The first is an instance of the widely studied family of dynamic programming sentence alignment procedures [9] [26] [84] [76] [62]. The second is a novel approach to bitext alignment by divisive clustering. We will show how these two very different alignment procedures can both be derived as MAP search procedures under the generative model. Their differences arise from changes in the form of the component distributions within the generative model.

2.3.1 Monotonic Alignment of Bitext Segments

Sentence alignment can be made computationally feasible through the imposition of alignment constraints. By insisting, for instance, that a segment in one language align to only 0, 1, or 2 segments in the other language and by requiring that the alignment be monotonic and continuous with the boundary constraints (Equation 2.2), an efficient dynamic programming alignment algorithm can be found (e.g. [26]).

We describe assumptions concerning the model introduced in the previous section that make it possible to obtain an efficient dynamic program algorithm to realize the MAP alignment. We note first that obtaining an efficient and optimal chunk alignment procedure $\hat{a}_1^{\hat{K}}$ is not straightforward in general due to the chunk count distribution $\beta(\cdot)$ and the normalization terms $Z_{m,n,K}$. A straightforward implementation

would find the optimal alignment for each K and choose the best one among them. This would require an exhaustive search of exponential complexity over all valid chunk alignments. We describe a particular model formulation under which MAP alignment by dynamic programming is possible and this exponential complexity is avoided.

Simplifying Assumptions for Efficient Monotonic Alignment

We assume that the chunk count likelihood $\beta(K|m,n)$ is proportional to the probability of finding an alignment with K chunks, i.e. that it is proportional to the normalization term $Z_{m,n,K}$. It follows therefore that

$$\beta(K|m,n) = \frac{Z_{m,n,K}}{Z_{m,n}}, \tag{2.8}$$

where $Z_{m,n} = \sum_{K=1}^{\min(m,n)} Z_{m,n,K}$. Equation (2.2) then simplifies to

$$P(\mathbf{t}_1^n, K, a_1^K|\mathbf{s}_1^m) = \frac{P(n|\mathbf{s}_1^m)}{Z_{m,n}} \prod_{k=1}^{K} p(a_k) P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k}) \tag{2.9}$$

and the best chunk alignment is defined as

$$\{\hat{K}, \hat{a}_1^{\hat{K}}\} = \underset{K,a_1^K}{\operatorname{argmax}} \prod_{k=1}^{K} p(a_k) P(\mathbf{t}_{a_k}|\mathbf{s}_{a_k})\,. \tag{2.10}$$

This alignment can be obtained via dynamic programming. Given two chunk prefix sequences $\mathbf{s}_1^i$ and $\mathbf{t}_1^j$, the likelihood of their best alignment is

$$\alpha(i,j) = \max_{k,a_1^k(i,j)} \prod_{k'=1}^{k} p(a_{k'}) P(\mathbf{t}_{a_{k'}}|\mathbf{s}_{a_{k'}})\,. \tag{2.11}$$

which can be computed recursively:

$$\alpha(i,j) = \max_{1\le x,y\le R} \alpha(i-x, j-y) \cdot p(x,y) \cdot P(\mathbf{t}_{j-y+1}^{j}|\mathbf{s}_{i-x+1}^{i}) \tag{2.12}$$

The dynamic programming procedure searches through an $m \times n$ grid as illustrated in Figure 2.1. The search is initialized with $\alpha(0,0) = 1$, and by backtracking from the final grid point (m,n) the optimal chunk alignment can be obtained. In this search, the optimum values of x and y are retained at each (i,j) along with the maximum α value.

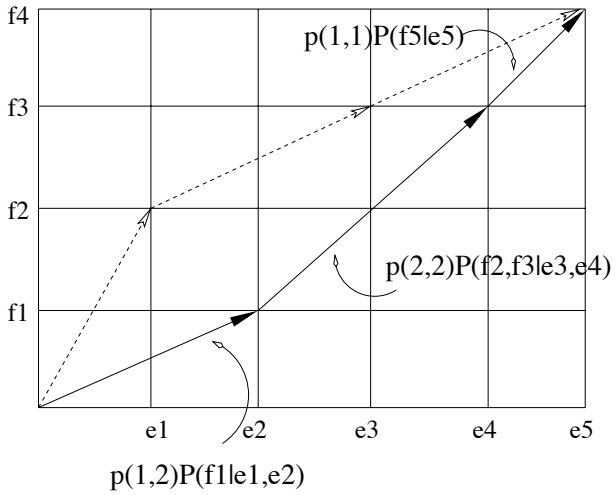


Figure 2.1: Bitext Chunking via. Dynamic Programming

We note that if we assume a flat translation table within IBM Model-1, i.e. that any word in the source language segment can be translated with equal probability as any word in the target language segment, then the algorithm is equivalent to dynamic programming based on sentence length [9, 26].

2.3.2 Divisive Clustering of Bitext Segments

The previous section describes a particular form of the chunk count distribution that leads to an efficient monotonic alignment. We now describe the alignment procedure that results when this distribution is defined so as to allow only binary segmentation of the documents, i.e. if

$$\beta(K|m,n) = \begin{cases} 1 & K = 2 \\ 0 & \text{otherwise} \end{cases} . \tag{2.13}$$

Under this distribution the segment sequences that make up each document are grouped into two chunks which are then aligned. With the range of K restricted in this way, the chunk alignment sequence contains only two terms: $a_1 a_2$. Given a parallel text $(\mathbf{s}, \mathbf{t})$, a_1^2 will split $\mathbf{s}$ into two chunks $\mathbf{s}_1\mathbf{s}_2$ and $\mathbf{t}$ into two chunks $\mathbf{t}_1\mathbf{t}_2$.

Under the model-based MAP criterion, the best split is found by

$$\hat{a}_1^2 = \underset{a_1,a_2}{\operatorname{argmax}}\, p(a_1)p(a_2)P(\mathbf{t}_1|\mathbf{s}_1)P(\mathbf{t}_2|\mathbf{s}_2)\,. \tag{2.14}$$

For simplicity, $p(a_1)$ and $p(a_2)$ are taken as uniform, although other distributions could be used. The search procedure is straightforward: all possible $m \times n$ binary alignments are considered. Given the simple form of Model-1, statistics within can be precomputed so that Equation (2.5) can be efficiently found over all these pairings.

Iterative Binary Search and Non-Monotonic Search

The above procedure is optimal for binary splitting and alignment. Of course, with lengthy documents a simple binary splitting is of limited value. We therefore perform iterative binary splitting in which the document pairs are aligned through a succession of binary splits and alignments: at each step, each previously aligned chunk pair is considered as a "document" which is divided and split by the above criteria.

The idea is to find the best split of each aligned pair into two smaller aligned chunks and then further split the derived chunk pairs as needed. As an alternative to dynamic programming alignment for bitext chunking, this divisive clustering approach is a divide-and-conquer technique. Each individual splitting is optimal, under the above criteria, but the overall alignment is of course suboptimal, since any binary splitting and pairing performed early on may prevent a subsequent operation which would be preferable. This approach is similar to other divisive clustering schemes, such as can be used in creating Vector Quantization codebooks [54] or decision trees [6], in which optimality is sacrificed in favor of efficiency.

The computational simplicity of this style of divisive clustering makes it feasible to consider non-monotonic search. In considering where to split a document, we allow the order of the resulting two chunks to be reversed. This requires a relaxation of the continuity constraint A5 given in Section 2.2, and while it does increase search complexity, we nevertheless find it useful to incorporate it within a hybrid alignment approach, described next.

An example of the divisive clustering procedure is given in Figure 2.3.2, showing the sequence of parallel binary splits.

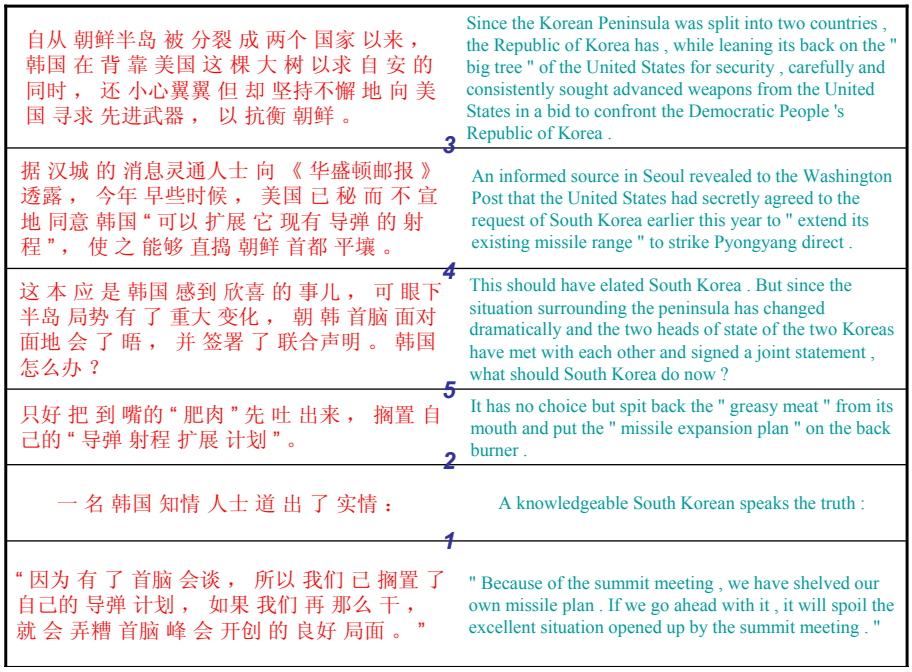


Figure 2.2: An example of the Divisive Clustering procedure applied to Chinese-English Bitext. Each line with a number on it splits a chunk pair into two smaller chunk pairs. The numbers represent the sequence of divisive clustering: i.e., "1" means the first split, so on and so forth.

2.3.3 A Hybrid Alignment Procedure

The Dynamic Programming and the Divisive Clustering algorithms arise from very different formulations of the underlying generative model. By appropriately defining the chunk count component distribution, we can obtain either of the two procedures. As a result, even though they search over the same potential chunk alignment hypotheses, the algorithms proceed differently and can be expected to yield different answers.

The two algorithms are complementary in nature. The monotonic, dynamic programming algorithm makes alignment decisions on a global scale so that the resulting

alignment is optimal with respect to the likelihood of the underlying model. The divisive clustering procedure is not globally optimal with respect to the overall likelihood, but it does divide each chunk optimally.

Efficient dynamic programming alignment procedures rely on monotonic alignment. This is not put forth as a rule, of course, but allowing for reordering would greatly complicate the search procedure and increase the size of the alignment trellis. In contrast, the relatively simple search space considered at each iteration of divisive clustering makes it feasible to incorporate simple binary reordering; this is possible because each iteration is done independently of its predecessors.

The analysis of the two algorithms suggests a hybrid alignment approach that takes advantage of the respective strengths of each procedure. We align documents by first applying the dynamic programming procedure to align documents at the sentence level. This is done to produce 'coarse chunks' containing as many as four sentences on either side. We then refine this initial alignment by divisive clustering to produce chunks with subsentential segments delimited by punctuation marks, or even by white spaces defining word boundaries. We refer to this hybrid, two stage procedure as "DP+DC".

The rationale underlying the hybrid approach is that reordering is more likely to occur at finer levels of alignment. A loose justification for this is that in translation, sentences are relatively unlikely to be moved from the beginning of a source document to the end of their target document, whereas subsentence target segments are more likely to appear within a several sentence neighborhood of their origins.

2.4 Summary

In this chapter, we have investigated statistical models of bitext chunk alignment, placing particular emphasis on the "chunk count" component distribution. Depending on how this distribution is defined, bitext alignment under a maximum likelihood criteria leads to very different types of alignment search strategies. A chunk count distribution that allows a detailed, monotone alignment of sentence segments can lead to dynamic programming alignment procedures of the sort that have been widely stud-

ied in previous work. If the distribution is defined so that only binary segmentation and alignment is allowed under the model, we obtain an iterative search procedure. We find that these two types of alignment procedures complement each other and that they can be used together to improve the overall alignment quality.

Statistical chunk alignment models are applied to extract chunk pairs at sentence or sub-sentence level. In the next chapter, we will use these derived chunk pairs as training material to build statistical word alignment models.

Chapter 3

Statistical Word Alignment Models

3.1 Introduction

One of the fundamental goals of SMT is describing word alignment. Alignment specifies how word order changes when a sentence is translated into another language, and given a sentence and its translation, alignment specifies translation at the word level. It is straightforward to extend word alignment to phrase alignment: two phrases align if their words align.

Deriving phrase pairs from word alignments is now widely used in phrase-based SMT. Parameters of a statistical word alignment model are estimated from bitext, and the model is used to generate word alignments over the same bitext. Phrase pairs are extracted from the aligned bitext and used in the SMT system. With this approach, the quality of the underlying word alignments can exert a strong influence on phrase-based SMT system performance. The common practice therefore is to extract phrase pairs from the best attainable word alignments. Currently, IBM Model-4 alignments [7] as produced by GIZA++ [70] are often the best that can be obtained, especially with large bitexts.

Despite its modeling power and widespread use, however, Model-4 has its shortcomings. Its formulation is such that maximum likelihood parameter estimation and bitext alignment are implemented by approximating, hill-climbing methods. As a consequence, parameter estimation can be slow, memory intensive, and difficult to

parallelize. It is also difficult to compute statistics under Model-4. This limits its usefulness for modeling tasks other than the generation of word alignments.

Hidden Markov Models (HMM) [74] are attractive for efficient parameter estimation algorithms and modeling various observation sequences. They have been successfully applied to and have become the mainstream models for acoustic observations in Automatic Speech Recognition (ASR) [33]. Although SMT is a different task from ASR, there are strong connections between them. They both employ the source-channel model, and both translation and transcription can be performed by source decoding algorithms. In ASR, acoustic observations need to be aligned with words/phones in training and decoding, while in SMT, word and phrase alignments are often necessary to build statistical translation models. If acoustic observations are quantized, under the HMM, both ASR and SMT model the relationships between two discrete sequences. Consequently, they share methodology and learning algorithms. Nonetheless, SMT exhibits more complicated alignment patterns than ASR; therefore, special attention should be paid to statistical modeling approaches in SMT. HMM has been explored in statistical translation models [81] [79], where words in one language are treated as states while words in the other language are regarded as observations.

We analyze model components of Model-4 and HMM-based word alignment models and identify the strengths and weaknesses of each model. As an alternative, we develop an HMM-based Word-to-Phrase (WtoP) alignment model [18] with the goal of building on the strengths of Model-4 and HMM. Basic model components are inspired by features of Model-4, but incorporated within HMM framework to allow more efficient parameter estimation and word alignment. The WtoP alignment model directly builds alignments between words and phrases.

In the word alignment and phrase-based translation experiments to be presented in Chapter 6, the WtoP model performance is comparable or improved relative to that of Model-4. Practically, we can train the model by the Forward-Backward algorithm, and by parallelizing estimation, we can control memory usage, reduce the time needed for training, and increase the bitext used for training. It will be shown in Chapter 4 that we can also compute statistics under the model in ways not practical with

Model-4, and we show the value of this in the extraction of phrase pairs from bitext.

3.2 Statistical Generative Word Alignment Models

To assign probabilities to target strings given source strings $P(\mathbf{t}|\mathbf{s})$, most statistical translation models assume a "*h*idden" random variable: word alignment $\mathbf{a}$ which specifies how words between source and target strings are to be aligned. A generative translation model describes how the target string $\mathbf{t}$ is generated from the source string $\mathbf{s}$ stochastically, which determines the conditional likelihood of "*c*omplete" data $P(\mathbf{t}, \mathbf{a}|\mathbf{s})$. The conditional likelihood of "*i*ncomplete" data, i.e., sentence translation probability, is obtained by considering all possible word alignments

$$P(\mathbf{t}|\mathbf{s}) = \sum_{\mathbf{a}} P(\mathbf{t}, \mathbf{a}|\mathbf{s}).$$

Model parameters are usually estimated from a collection of sentence pairs via the Expectation Maximization (EM) algorithm [17].

Given a pair of sentence $(\mathbf{s}, \mathbf{t})$, the best word alignment under the model is given by the Maximum A Posterior (MAP) criterion

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, P(\mathbf{a}|\mathbf{s}, \mathbf{a}) = \underset{\mathbf{a}}{\operatorname{argmax}}\, P(\mathbf{t}, \mathbf{a}|\mathbf{s}).$$

The problem of word alignment is generally defined as building word links between a bilingual sentence-aligned corpus. Formally, for each sentence pair $(\mathbf{s} = s_1^I, \mathbf{t} = t_1^J)$ in the corpus, the problem is to produce a bag of word links$\mathbf{A} = \{(i, j)\}$, where i is the word index of sentence $\mathbf{s}$ and j is the word index of sentence $\mathbf{t}$. This definition applies to general word alignment models. Statistical generative models [7] [81] [70] find a source word for each target word. This is expressed by the hidden random variable $\mathbf{a} = a_1^J$, which maps each target word t_j to the source word at position a_j, $j = 1, 2, \cdots, J$; when $a_j = 0$, t_j is generated from the empty (aka NULL) word, i.e., the target word t_j has no correspondence in the source string. Therefore, the bag of word links generated is

$$\mathbf{A} = \{(a_j, j)|j = 1, 2, \cdots, J\}.$$

This notion implies that any one target word can be linked to at most one source word.

Statistical generative word alignment models distinguish themselves by assigning the conditional likelihood function $P(\mathbf{t}, \mathbf{a}|\mathbf{s})$ differently, according to their generative procedures and underlying assumptions. We will briefly review HMM-based word alignment models [81] [70] [79] and IBM serial word alignment models [7] by describing their generative procedures and identifying their key model components.

3.2.1 HMM-based Statistical Word Alignment Models

The HMM word alignment model [81] constructs a Markov space by treating each source word s_i as a state and building fully connected transitions between states. The target sentence t_1^J is an observation sequence. Target words are emitted one by one from left to right after each state transition. Figure 3.1 shows an example of HMM state sequence for the target string. The state output probability distribution is the word-to-word translation table $t(t|s)$, called *t-table*. Given these, the conditional likelihood function reads as

$$P(\mathbf{t}, \mathbf{a}|\mathbf{s}) = P(\mathbf{a}|\mathbf{s})P(\mathbf{t}|\mathbf{s}, \mathbf{a}) = \prod_{j=1}^{J} P(a_j|a_{j-1}; I)t(t_j|s_{a_j})$$

Like HMMs in acoustic modeling, an efficient forward-backward algorithm can be applied to train parameters of HMM word alignment models. The Viterbi algorithm can be used to find the most likely state sequence, which corresponds to the best word alignment under the model. To make this procedure faster, pruning can be employed.

To handle target words for which no source words are responsible, i.e., target words aligned to the empty word, Och and Hey [70] expanded the Markov network by doubling the state space. Target words emitted by the added states are aligned to the empty word. The Markov transition matrix is adjusted accordingly to allow transitions to the added states.

Toutanova *et al* [79] extended the HMM-based word alignment models in several ways. One of them is by augmenting bilingual sentence pairs with part-of-speech

tags as linguistic constraints for word alignments. They also introduced the "stay" probability $P(\text{stay}|s)$ to model the state duration information of the Markov network.

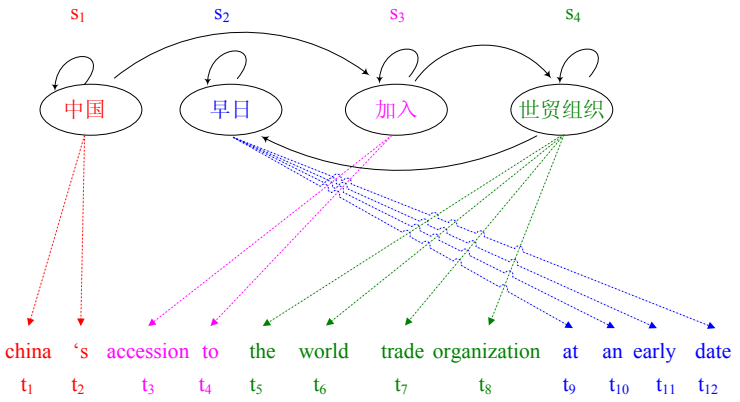


Figure 3.1: An example of HMM-based word-to-word Alignment Models. The source string is a Chinese word sequence and the target string is an English word sequence. A Markov network is established by treating source words as Markov states and target words as output sequence. A target word is emitted one time after a state transition.

3.2.2 IBM Models

The IBM word alignment serial model [7] includes five statistical translation models from the simplest Model-1 to the finest Model-5. Model complexity is increased gradually with more parameters to estimate.

Model 1 & 2

Model-1 and Model-2 assume that the target word string is generated independently from the source string. For each target position, a source position is chosen randomly; then, the target word is sampled from the chosen source word translation table. In Model-1, source positions are selected uniformly, while in Model-2, they depend on the actual position and the lengths of the two strings. Formally, let $a(i|j;I,J)$ be the probability of the j-th target word choosing the i-th source word,

then the conditional likelihood is given by

$$P(\mathbf{t}, \mathbf{a}|\mathbf{s}) = \prod_{j=1}^{J} a(a_j|j; I, J)t(t_j|s_{a_j}). \tag{3.1}$$

Model-1 is a special case of Model-2 where $a(i|j; I, J) = \frac{1}{I+l}$ regardless of i and j.

Fertility-based Models

While Model-1 and Model-2 select source words for target words, Model-3 and Model-4 operate in the other direction in a generative way. For each source word, Model-3, 4 first decide how many target words it generates, termed *fertility*, by table lookup, and then sample that amount of target words from the translation table to form a list, termed *tablet*. Let m' be the sum of fertilities of source words. A tablet is also created for the empty word. The number of entries in the empty word tablet is chosen from the Binomial distribution $B(m', p_1)$, and similarly, that amount of target words is sampled from the empty word translation table. Finally the models position the sampled words to produce a target string according to the *distortion models*. Figure 3.2 shows the generative process under fertility-based models for the same sentence pairs as in Figure 3.1. In Model-3, target positions are chosen independently for sampled target words in a tablet; while in Model-4, there are two types of distortion models: one is applied to position the first word in the tablet; the other determines the relative distance to the previous chosen position stochastically. For target words in the empty word tablet, positions are chosen from the remaining vacant positions uniformly.

Since different target words might be positioned in the same place, Model-3 and Model-4 assign non-zero probability to invalid strings, which leads to a deficient model. Model-5 is similar to Model-4 but is refined to avoid the deficiency problem. Nonetheless, Model-4 word alignments as produced by the GIZA++ Toolkit [70] have been widely used in statistical machine translation systems because of their high quality.

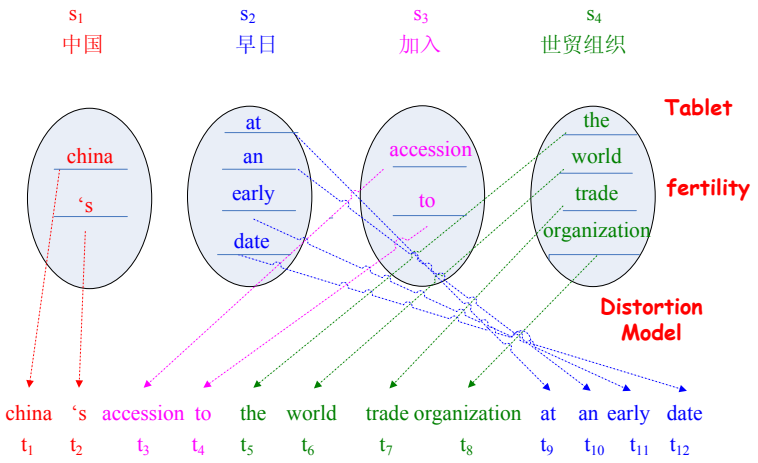


Figure 3.2: Simplified Example of Word-to-Phrase Alignments under IBM Model-4. Source word *fertility* determines how many target words are generated by each source word; word-to-word translation tables create the specified number of target words as translations of each source word; and the *distortion* model then specifies the order of the target words in the target sentence. This example presents the target sentence as a sequence of phrases, but there is no requirement that the target words generated by a source word should appear as neighbors in the target sentence.

Deriving Word Alignments

Deriving the best word alignment under Model-1 and Model-2 is almost trivial, since each target word chooses source words independently. However, finding the best word alignment under fertility-based models is quite different. Fertilities encode "global" generative information, and as a result, they cannot be determined until the whole target sequence is observed. This "global" constraint prevents efficient algorithms from being explored. Consequently, sub-optimal algorithms, say hill-climbing [7] [70], are considered. For instance, the Viterbi alignment of less complicated models, say Model-2 or HMM, is employed as a starting point to be used until a better word alignment in the 'neighborhood' of the current best word alignment is found and made to serve as an anchor for the next round; this process is repeated until no better word alignments can be found. A neighborhood of a word alignment $\mathbf{a}$ is the set of word alignments derived by performing only one move or swap operation on the word alignment $\mathbf{a}$.

Model Training

For Model-1, EM algorithm is guaranteed to find the model parameters that achieve the global maximum of the target function. Full-EM training of Model-2 is also efficient. The posterior of the j-th target word choosing the i-th source word given the sentence pair and current model θ is

$$P_\theta(a_j = i|\mathbf{s}, \mathbf{t}) = \frac{a(i|j; I, J)t(t_j|s_i)}{\sum_{i'=0}^{I} a(i'|j; I, J)t(t_j|s_{i'})} \tag{3.2}$$

Statistics for each parameter can be collected accordingly.

Similar to finding the best word alignments, training fertility-based models is also sub-optimal for practical reasons. Unlike HMM-based models, where all possible word alignments are considered, statistics are collected over a subset of all word alignments during training.

3.2.3 Discussion

HMM-based word alignment models and IBM fertility-based models are quite different. The Markov assumption in HMM generates target words locally, which enables efficient dynamic programming based algorithms to train and find the best word alignments. The distortion model and fertility information in Model-4 together produce word alignments with better quality than that of HMM, but they also make training procedure computationally complicated. However, both models are based on word to word alignments.

In this chapter, we present an HMM-based Word-to-Phrase alignment model with the goal of producing a training algorithm that is as efficient as HMM-based word-to-word alignment models but, at the same time, is also capable of producing word alignments comparable to Model-4. The idea is to make the Markov process more powerful and efficient in generating observation sequences: phrases rather than words are emitted after each state transition, as shown in Figure 3.3. We establish links between source words and target phrases explicitly during the generative procedure. A phrase is defined as one word or a consecutive word sequence. Context within a phrase is also considered in the model.

The model is formally presented in the next section.

3.3 HMM-Based Word-to-Phrase Alignment Models

One of the fundamental goals of SMT is to describe how word order changes when a sentence in one language is translated into another language. Intuitively, alignment specifies which portions of each sentence can be considered to be translations.

Our goal is to develop a generative probabilistic model of Word-to-Phrase (WtoP) alignment. We begin with a detailed description of the component distributions.

3.3.1 Component Variables and Distributions

We start with a source sentence of I words, $\mathbf{s} = s_1^I$, and its translation as a J word sentence in the target language, $\mathbf{t} = t_1^J$; this is assumed to be a pair of 'correct' translations whose word alignment is unknown. We will model the generation of the target language word sequence via an intermediate sequence of target language phrases. Here, 'phrase' refers only to continuous word sequences in the target language of variable length; any continuous subsequence of the target sentence can serve as a phrase.

We introduce only a minimal structure to describe the segmentation of target sentences into phrase sequences. We define the Phrase Count variable K, which specifies that the target language sentence is segmented into a sequence of phrases: $\mathbf{t} = v_1^K$. The central modeling assumption is that each phrase in the target phrase sequence v_1^K is generated as a translation of a single source word. The correspondence between source words and target phrases is determined by the alignment sequence a_1^K. In this way, the k^{th} target phrase is generated from the a_k^{th} source word: $s_{a_k} \rightarrow v_k$. The number of words in each target phrase is specified by the random process ϕ_k. Of course this process is constrained so that the number of words in the phrase sequence agrees with the target sentence length, i.e. $\sum_{k=1}^{K} \phi_k = J$.

It is necessary as a practical matter in modeling translation to allow for the insertion of target phrases. This need arises because the correspondence between source sentence and target sentence is not always exact, and it may be better to allow phrases to be inserted than to insist that they align to a source word. This is typically done by allowing alignments to a non-existent NULL source word. An alternative formulation is to introduce a binary 'hallucination' sequence h_1^K that determines how each phrase is generated: if $h_k = 0$, then NULL $\rightarrow v_k$; if $h_k = 1$ then $s_{a_k} \rightarrow v_k$. If the hallucination process takes a value of 0, the corresponding phrase is hallucinated rather than generated as a translation of one of the words in the source sentence.

Taken together, these quantities describe a phrase segmentation of the target language sentence and its alignment to the source sentence: $\mathbf{a} = (\phi_1^K, a_1^K, h_1^K, K)$. The modeling objective is to define a conditional distribution $P(\mathbf{t}, \mathbf{a}|\mathbf{s})$ over these alignments. With the assumption that $P(\mathbf{t}, \mathbf{a}|\mathbf{s}) = 0$ if $\mathbf{t} \neq v_1^K$, we write $P(\mathbf{t}, \mathbf{a}|\mathbf{s}) = P(v_1^K, K, a_1^K, h_1^K, \phi_1^K|\mathbf{s})$ and

$$\begin{aligned} P(v_1^K, K, a_1^K, h_1^K, \phi_1^K|\mathbf{s}) &= P(K|J, \mathbf{s}) \times P(a_1^K, \phi_1^K, h_1^K|K, J, \mathbf{s}) \\ &\quad \times P(v_1^K|a_1^K, h_1^K, \phi_1^K, K, J, \mathbf{s}) \end{aligned}$$

These are the natural dependencies of the component variables in the Word-to-Phrase alignment model. We now describe the simplifying assumptions made in their realization. The objective is to not to define the ideal realization of each component; many of the assumptions are admittedly simplistic and may be improved upon in the future. However simplicity is preferred wherever possible so as to control the complexity of the overall model.

Phrase Count Distribution $P(K|J, \mathbf{s})$ specifies the distribution over the number of phrases in the target sentence given the source sentence and the number of words in the target sentence. We use a simple, single parameter distribution

$$P(K|J, \mathbf{s}) = P(K|J, I) \propto \eta^K.$$

The scalar $\eta \geq 1$ controls segmenting of the target sentence into phrases; for example, larger η values favor target sentence segmentations with many short phrases. In

practice, we use η as a tuning parameter to roughly control the length of the target phrases hypothesized over training data.

Word-to-Phrase Alignment Distribution Before the words of the target language phrases are generated, the alignment of the target phrases to the source words is described. The alignment is modeled as a Markov process that specifies the lengths of phrases and the alignment of each one to one of the source words

$$\begin{aligned} P(a_1^K, h_1^K, \phi_1^K | K, J, \mathbf{s}) &= \prod_{k=1}^{K} P(a_k, h_k, \phi_k | a_{k-1}, \phi_{k-1}, h_{k-1}, K, J, \mathbf{s}) \\ &= \prod_{k=1}^{K} p(a_k | a_{k-1}, h_k; I) \cdot d(h_k) \cdot n(\phi_k; s_{a_k}) \end{aligned}$$

The actual word-to-phrase alignment (a_k) is a Markov process over the source sentence word indices, as in HMM-based word-to-word alignment [81]. It is formulated with a dependency on the hallucination variable so that target phrases can be inserted without disrupting the Markov dependencies of phrases aligned to actual source words

$$p(a_j | a_{j-1}, h_j; I) = \begin{cases} 1 & a_j = a_{j-1},\ h_j = 0 \\ 0 & a_j \neq a_{j-1},\ h_j = 0 \\ p_a(a_j | a_{j-1}; I) & h_j = 1 \end{cases}$$

The target phrase length model $n(\phi; s)$ is a form of source word fertility. It specifies the probability that a source word s produces a target phrase with ϕ words. The distribution $n(\phi; s)$ is maintained as a table for each source word for $\phi = 1, \cdots, N$. Similarly, the model requires a table of Markov transition probabilities $p_a(i'|i; I)$ for all source sentence lengths I.

The hallucination sequence is a simple i.i.d. process, where $d(0) = p_0$ and $d(1) = 1 - p_0$. Specified in this way, p_0 acts as a tuning parameter that controls the tendency towards the insertion of target phrases.

Word-to-Phrase Translation The translation of words to phrases is given as

$$P(v_1^K | a_1^K, h_1^K, \phi_1^K, K, J, \mathbf{s}) = \prod_{k=1}^{K} p(v_k | s_{a_k}, h_k, \phi_k),$$

so that target phrases are generated independently by individual source words. We define two models of word-to-phrase translation.

The simplest model of word-to-phrase translation is based on context-independent, word-to-word translation: target phrase words are translated independently from the source word via fixed translation tables

$$p(v_k|s_{a_k}, h_k, \phi_k) = \prod_{j=1}^{\phi_k} t_1(v_k[j] \mid h_k \cdot s_{a_k})$$

where the notation $h_k \cdot s_{a_k}$ is shorthand for

$$h_k \cdot s_{a_k} = \begin{cases} s_{a_k} & h_k = 1 \\ \text{NULL} & h_k = 0 \end{cases}$$

In this way specialized translation tables can be dedicated to hallucinated phrases in case their statistics differ from the phrases that arise from a direct translation of source words.

A more complex realization of word-to-phrase translation captures word context within the target language phrase via *bigram translation probabilities*

$$p(v_k|s_{a_k}, h_k, \phi_k) = t_1(v_k[1] \mid h_k \cdot s_{a_k}) \prod_{j=2}^{\phi_k} t_2(v_k[j] \mid v_k[j-1], h_k \cdot s_{a_k}).$$

Here, $t_1(t|s)$ is the usual context independent word-to-word translation probability described above. The bigram translation probability $t_2(t|t', s)$ specifies the likelihood that the target word t is to follow t' in a phrase generated by the source word s. Note that the conditioning is on words within the target phrase.

Summary The parameter set θ of this formulation of the HMM-based Word-to-Phrase alignment model consists of the Markov transition matrix p_a, the phrase length table n, the hallucination parameter p_0, the unigram word-to-word translation table t_1, and the bigram translation probabilities t_2:

$$\theta = \{p_a(i|i'; I), n(\phi; s), p_0, t_1(t|s), t_2(t|t', s)\}$$

The stochastic process by which a source string $\mathbf{s}$ generates a target string $\mathbf{t}$ of J words is as follows :

1. The number of phrases in the target sentence is chosen under $P(K|I, J)$.

2. For each of the K target phrases to be produced :

   (a) The alignment a_1^K is generated along with the hallucination process h_1^K.

   (b) With the alignment of the k^{th} phrase to the a_k^{th} source word set, the number of words in the k^{th} phrase is then chosen under distribution $n(\phi_k; s_{a_k})$. The ϕ_k are constrained to satisfy $\sum_{k=1}^{K} \phi_k = J$.

   (c) The words in the target phrase v_k are chosen under $P(v_k|s_{a_k}, h_k, \phi_k)$, where the hallucination process controls the insertion of target phrases.

3. The target sentence is formed from the target phrase sequence: $\mathbf{t} = v_1^K$.

Although it relies on target phrases, the Word-to-Phrase alignment model is very much a model of word translation in that it produces target phrases as sequences of words generated by a single source word. Phrase-level information is used primarily to influence the translation of individual words. The alignment of source and target words can easily be derived from the word-to-phrase alignments: words in a target phrase are aligned to the source word that generated the phrase.

3.3.2 Comparisons with other Models

The formulation of the Word-to-Phrase (WtoP) alignment model was motivated by both the HMM word alignment model [81] and IBM fertility-based models with the goal of building on the strengths of each.

The relationship with the word-to-word HMM alignment model is straightforward. For example, constraining the phrase length component $n(\phi; s)$ to permit only one word phrases would give a word-to-word HMM alignment model. The extensions introduced here include the phrase count, the phrase length models, and the bigram translation distribution. The hallucination process is motivated by the use of NULL alignments in the Markov alignment models [70].

The phrase length model is motivated by [79], who introduced "stay" probabilities in HMM alignment as an alternative to word fertility. By comparison, Word-to-Phrase

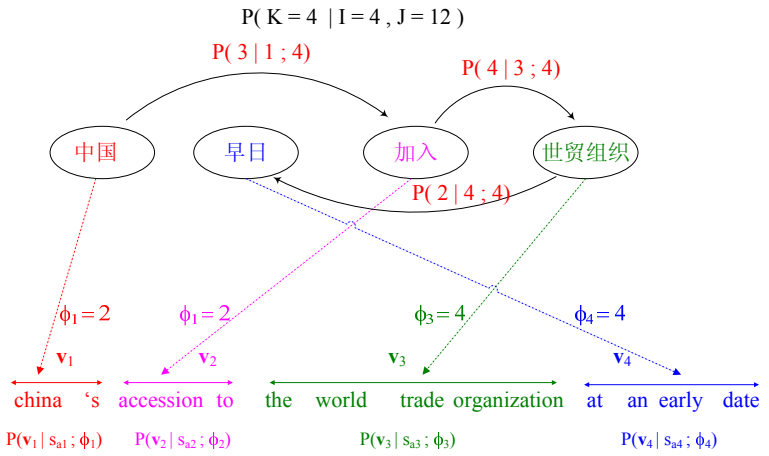


Figure 3.3: Simplified Example of HMM-Based Word-to-Phrase Alignment Model. The source string is a Chinese word sequence and the target string is an English word sequence. A Markov network is established by treating source words as Markov states, with the state dependent observation distributions defined over phrases of target words.

HMM alignment models contain detailed models of state occupancy, motivated by the IBM fertility model, which are more powerful than any single duration parameter (the "stay" probability). In fact, the WtoP model is similar to a segmental Hidden Markov Model [71], in which states emit observation sequences.

Comparison with Model-4 is less straightforward. The main features of Model-4 are NULL source words, source word fertility, and the distortion model. The WtoP alignment model includes the first two of these. However, distortion, which allows hypothesized words to be distributed throughout the target sentence, is difficult to incorporate into a model that supports efficient DP-based search. We preserve efficiency in the WtoP model by insisting that target words form connected phrases; this is not as general as Model-4 distortion. This constraint is somewhat offset by a more powerful (Markov) alignment process and the phrase count distribution.

Despite these differences, the WtoP alignment model and Model-4 allow similar alignments. For example, in Fig. 3.4, Model-4 would allow t_1, t_3, and t_4 to be generated by s_1 with a fertility of 3. Under the WtoP model, s_1 could generate t_1 and $t_3 t_4$ with phrase lengths 1 and 2, respectively: source words can generate more than one

phrase. If the state represented by a source word is not revisited, the phrase length of the target phrase aligned to the source word indeed is equal to its fertility.

The phrase length model is not exactly equivalent to the fertility model in Model-4, but it has a similar descriptive model. It is inspired by certain features of Model-4, but incorporated within HMM to allow efficient estimation and alignment.

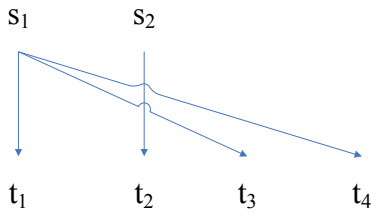


Figure 3.4: Word-to-Word and Word-to-Phrase Links

There is, of course, much prior work in translation that incorporates phrases. Sumita *et al* [78] developed a model of phrase-to-phrase alignment, which, while based on the HMM alignment process, appears to be deficient. Marcu and Wong [57] proposed a model to discover lexical correspondences at the phrase level.

The idea of explicitly aligning source words to target phrases has been explored for statistical natural language understanding [24] [73]. Epstein *et al* [24] proposed a statistical word-to-clump generative model with a uniform alignment distribution as in IBM Model-1. The "clump" can be understood as a phrase. Alignment distortions were suggested and studied in [23]. Della Pietra *et al* [73] extended the concept of fertility [7] to the generation of phrases and proposed improved word to phrase translation probabilities by utilizing context. While our model shares similar idea of generating a buck of target words from a source word to these models [24] [23] [73], we capture alignment distortion with a Markov stochastic process.

3.3.3 Embedded Model Parameter Estimation

We now discuss estimation of the WtoP model parameters by the EM algorithm. Since the WtoP model can be treated as an HMM, albeit with a somewhat complicated state space, it is straightforward to apply Baum-Welch parameter estimation.

The recursion runs forward, i.e. word by word, over the target sentence and gathers statistics relative to the alignment of target phrases to source words.

Forward-Backward Procedure

Given a sentence pair (s_1^I, t_1^J), a state space

$$(i, \phi, h) : 1 \leq i \leq I, 1 \leq \phi \leq N, h = 0 \text{ or } 1$$

is created over which the Forward-Backward algorithm will be carried out. The forward statistic $\alpha_j(i, \phi, h)$ is defined as the probability that the complete source sentence generates the first j target words, with the additional constraint that the last ϕ target words form a phrase generated by source word s_i . Including the influence of the hallucination process, this is written as

$$\alpha_j(i, \phi, h) = \begin{cases} P(t_1^j,\ t_{j-\phi+1}^j \leftarrow s_i \mid s_1^I) & h = 1 \\ P(t_1^j,\ t_{j-\phi+1}^j \leftarrow \text{NULL} \mid s_1^I) & h = 0 \end{cases} .$$

The forward statistics can be calculated recursively as

$$\begin{aligned} \alpha_j(i, \phi, h) = & \Big\{ \sum_{i', \phi', h'} \alpha_{j-\phi}(i', \phi', h') p(i|i', h; I) \Big\} \cdot \eta \cdot n(\phi; h \cdot s_i) \\ & \cdot t_1(t_{j-\phi+1}|h \cdot s_i) \cdot \prod_{j'=j-\phi+2}^{j} t_2(t_{j'}|t_{j'-1}, h \cdot s_i) . \end{aligned} \tag{3.3}$$

This recursion is over a trellis of $2NIJ$ nodes.

Similarly, the backward probability $\beta_j(i, \phi, h)$ is defined as the probability that the complete source sentence generates the final $I - j$ target words, given that the target words $t_{j-\phi+1}^j$ form a phrase aligned to $h \cdot s_i$:

$$\beta_j(i, \phi, h) = \begin{cases} P(t_{j+1}^I \mid t_{j-\phi+1}^j \leftarrow s_i\ ,\ s_1^I) & h = 1 \\ P(t_{j+1}^I \mid t_{j-\phi+1}^j \leftarrow \text{NULL}\ ,\ s_1^I) & h = 0 \end{cases} .$$

It can be calculated recursively over the same trellis as

$$\begin{aligned} \beta_j(i, \phi, h) = & \sum_{i', \phi', h'} \beta_{j+\phi'}(i', \phi', h') \cdot p(i'|i, h'; I) \cdot \eta \cdot n(\phi'; h' \cdot s_{i'}) \\ & \cdot t_1(t_{j+1}|h' \cdot s_{i'}) \cdot \prod_{j'=j+2}^{j+\phi'} t_2(t_{j'}|t_{j'-1}, h' \cdot s_{i'}) \end{aligned} . \tag{3.4}$$

Word to Phrase Translation Statistics

At the completion of the Forward recursion, the conditional probability of sentence $\mathbf{t}$ given $\mathbf{s}$ can be found as

$$P(\mathbf{t} \,|\, \mathbf{s}) = \sum_{i',h',\phi'} P(t_1^J,\, t_{J-\phi'+1}^J \leftarrow h' \cdot s_{i'} \mid \mathbf{s}) \,. \tag{3.5}$$

The corresponding relationship holds for the backward probability, as usual.

The probability that a phrase $t_{j-\phi+1}^j$ is generated by any of the words in the source sentence can be found as

$$\begin{aligned} P(\mathbf{t},\, t_{j-\phi+1}^j \leftarrow h \cdot s_i \mid \mathbf{s}) &= P(t_{j+1}^I \mid t_{j-\phi+1}^j \leftarrow h \cdot s_i \,,\, s_1^I) \;\cdot \\ &\qquad P(t_1^j,\; t_{j-\phi+1}^j \leftarrow h \cdot s_i \mid s_1^I) \\ &= \alpha_j(i,\phi,h)\; \beta_j(i,\phi,h) \end{aligned}$$

With these quantities computed, the posterior probability that target words $t_{j-\phi+1}^j$ form a phrase aligned to the source word $h \cdot s_i$ can be found as

$$\gamma_j(i,\phi,h) = P(t_{j-\phi+1}^j \leftarrow h \cdot s_i \mid \mathbf{s},\mathbf{t}) = \frac{\alpha_j(i,\phi,h)\beta_j(i,\phi,h)}{\sum_{i',h',\phi'} \alpha_J(i',\phi',h')}.$$

Finally, re-estimation of the Markov transition matrix requires the posterior probability of observing *pairs* of word-to-phrase translation instances. The probability that a phrase $t_{j-\phi'+1}^j$ is generated by $h' \cdot s_{i'}$ and that the next phrase $t_{j+1}^{j+\phi}$ is generated by $h \cdot s_i$ can be found as

$$\begin{aligned} &P(\mathbf{t},\, t_{j-\phi'+1}^j \leftarrow h' \cdot s_{i'} \,, t_{j+1}^{j+\phi} \leftarrow h \cdot s_i \mid \mathbf{s}) = \\ &\quad \alpha_j(i',\phi',h') \cdot P(i|i',h;I) \cdot \eta \cdot n(\phi; h \cdot s_i) \cdot P(t_{j+1}^{j+\phi}|s_i,h,\phi) \cdot \beta_{j+\phi}(i,\phi,h) \,. \end{aligned} \tag{3.6}$$

The posterior probability can be found as the ratio of Equation 3.6 to Equation 3.5 :

$$\gamma_j(i',\phi',h',i,\phi,h) = P(t_{j-\phi'+1}^j \leftarrow h' \cdot s_{i'} \,, t_{j+1}^{j+\phi} \leftarrow h \cdot s_i \,|\mathbf{t},\; \mathbf{s}) \,.$$

Parameter Update Equations

The update equations for the context independent translation table t_1 and the Markov transition probability p_a are given here; the remaining model parameters are updated in a similar manner.

Let $\mathbf{T}$ denote the collection of sentence pairs in the bitext training set. Let (s,t) be the posterior counts accumulated over all training bitext, as follows

$$c(s,t) = \sum_{(\mathbf{s},\mathbf{t})\in\mathbf{T}} \sum_{\substack{i,j,\phi,\\ s_i=s}} \gamma_j(i,\phi,h=1)\ \#_j(t,\phi)$$

where $\#_j(t,\phi) = \sum_{j'=j-\phi+1}^{j} 1_t(t_{j'})$ is the number of times word t appears in the phrase $t_{j-\phi+1}^{j}$. The updated estimate of the unigram translation probability is then found as

$$\hat{t}_1(t|s) = \frac{c(s,t)}{\sum_{t'} c(s,t')}\ .$$

The word-to-phrase translation pair statistics are gathered as

$$c(i',i;I) = \sum_{\substack{(\mathbf{s},\mathbf{t})\in\mathbf{T},\\ |\mathbf{s}|=I}} \sum_{j,\phi',h',\phi} \gamma_j(i',\phi',h',i,\phi,h=1),$$

where $|\mathbf{s}|$ is the number of words in $\mathbf{s}$. The re-estimated transition probability is then computed as

$$\hat{p}_a(i|i';I) = \frac{c(i',i;I)}{\sum_{i''} c(i'',i;I)}\ . \tag{3.7}$$

Iterative Estimation Procedures

As with training the IBM fertility-based models [7, 70], the WtoP model parameters are estimated incrementally so that model complexity increases only as training progresses. Any number of training scenarios for the Word-to-Phrase Alignment HMM are possible, however the experiments that will be reported later in this paper were based on the followed recipe, which has proven to be fairly reliable.

Model parameters are trained from a flat-start, without use of any prior alignment information. The final model complexity is determined by the maximum phrase length N_{max}, which is decided upon beforehand, or verified subsequently through testing of the models.

- Translation and transition tables are initialized as uniform distributions.
- Model-1 parameters are estimated with 10 iterations of EM.

- Model-2 parameters are estimated with 5 iterations of EM.
- The parameters of a word-to-word HMM alignment model are initialized by word alignment counts from Model-2 Viterbi alignments of the bitext.
- Word-to-word HMM parameters are estimated with 5 iterations of EM.
- For $N = 2, \ldots, N_{max}$:
  - Parameters of Word-to-Phrase alignment HMMs with maximum phrase length of N are estimated with 5 iterations of EM.
- Bigram translation tables t_2 are cloned from unigram tables t_1.
- Word-to-Phrase Alignment HMMs with bigram-translation tables are estimated with 5 iterations of EM

This strategy of gradually increasing model complexity as training progresses is motivated by experience in estimating the parameters of large language processing systems, notably the 'incremental build' approach to building mixture of Gaussian distribution models in automatic speech recognition [92].

Robust WtoP HMM Parameter Estimation

The component distributions that make up the Word-to-Phrase HMM come together to form an extremely complex system. Even with large amounts of training bitext, there is significant risk of overtraining unless preventative steps are taken. We now discuss simple parameter smoothing techniques for robust estimation of the Word-to-Phrase HMM transition matrices and the bigram translation probabilities.

Estimation of Markov Alignment Transition Matrices When estimated in the usual way (via Equation 3.7), the transition probabilities, $P_a(i|i'; I)$, are based on statistics of the (conditional) expectation that consecutive target phrases are generated by source words in positions i' to i within source sentences of length I. Clearly,

this level of modeling specificity could easily suffer from observation sparsity within the available bitext.

To address this particular problem, Vogel *et al* [81] suggested the use of 'jump dependent' Markov transition probabilities, which we adopt here in modified form. The jump transition probability $p_a^{(jump)}(i|i';I))$ is a function only of the 'jump' $i - i'$ made in the alignment sequence. In estimating $p_a^{(jump)}(i|i';I)$, all accumulators corresponding to state transitions with a jump of $i - i'$ contribute to estimating the jump transition probability. The goal is to improve robustness by sacrificing some of the descriptive power of this component so that there are few parameters to estimate.

We employ a simple interpolation scheme to obtain transition probabilities $\hat{p}_a(i|i';I)$ after each iteration of EM as a linear interpolation of the Maximum Likelihood estimate $p_a(i|i';I)$, the 'jump' transition probabilities $p_a^{(jump)}(i|i';I)$ and the uniform distribution $1/I$

$$\tilde{p}_a(i|i';I) = \lambda_1 \cdot \hat{p}_a(i|i';I) + \lambda_2 \cdot p_a^{(jump)}(i|i';I) + \lambda_3 \cdot \frac{1}{I} \tag{3.8}$$

with $\hat{p}_a$ estimated by the unsmoothed EM estimate of Equation 3.7. The interpolation parameters λ_1, λ_2, and λ_3 are positive, sum to 1, and are tuned over held-out development data.

Performing parameter interpolation in this way does improve robustness, but it is less effective than estimation strategies that control the overall model complexity relative to the amount of relevant training data. We next investigate the use of such techniques for estimation of the bigram translation probabilities.

Estimation of Bigram Translation Probabilities The bigram translation probability assigns likelihood to a target word t which follows another target word t' in a phrase generated as a translation of a given source word s. This probability has the form of a predictive bigram language, $t_2(t|t',s)$, and we borrow techniques from statistical language modeling for its robust estimation. Any of the many backoff schemes for n-gram language modeling could be used, and here we investigate Witten-Bell [83] smoothing.

Let $k(t',t,s)$ be the expected number of occurrences of t given that t follows t'

in a phrase translated from source word s. We choose a threshold L such that the conditional bigram is treated as an unseen event if $k(t', t, s) < L$. For those less frequent events, we back off to the word-to-word translation probabilities, t_1. The total count of seen events is defined as $N(t', s) = \sum_{t:k(t',t,s)\geq L} k(t', t, s)$ and the total seen event types as $T(t', s) = \sum_{t:k(t',t,s)\geq L} 1$. Using these quantities we define

$$\lambda_{t',s} = \frac{T(t', s)}{T(t', s) + N(t', s)}$$

as the total probability mass to be assigned to all unseen events. This probability mass is distributed according to the "unigram" distribution, which is a word-to-word translation probability.

$$t_2(t|t', s) = \begin{cases} (1 - \lambda_{t',s})\frac{k(t',t,s)}{N(t',s)} & k(t', t, s) \geq L \\ \lambda_{t',s}\frac{t(t|s)}{\gamma_{t',s}} & \text{otherwise} \end{cases} \tag{3.9}$$

where $\gamma_{t',s} = \sum_{t:k(t',t,s)<L} t(t|s)$ is introduced for normalization.

3.3.4 Deriving Word Alignments

Although the WtoP alignment model is more complex than the word-to-word HMM alignment model, the Viterbi algorithm can still be used. If we replace the "summation" operation with the "max" operation in Equation (3.3) of the Forward procedure, we obtain the partial HMM likelihood of the best path to the target word j at each state (i, ϕ, h). We then store the preceding incoming state in the best path. At the end of the target string, it is possible to trace stored records to find out the best word-to-phrase alignments. To make the algorithm faster, pruning could be applied.

Alternative alignment algorithms are also possible. For instance, post processing of the state occupancy network with heuristics or additional knowledge resources could possibly produce better alignments under alternative criteria to maximum likelihood.

Word-to-word alignments are generated directly from the most likely word-to-phrase alignments: if $s_{a_k} \rightarrow v_k$, the source word s_{a_k} is linked to all the words in the target phrase v_k.

3.3.5 Discussion

After transitioning to a state in the Markov network, a decision must be made about emitting a word or a phrase. The balance between word-to-word and word-to-phrase alignments is set by the phrase count distribution parameter η. As η increases, alignments with shorter phrases are favored, and for very large η, the model allows only word-to-word alignments. It is desirable to have a balanced distribution that leads to the best overall word alignment quality as measured by Alignment Error Rate [70]. Indeed, our experimental results support this position.

The unigram word-to-phrase translation probability is a bag-of-words model. Permutating words within a phrase does not affect the probability. However, the bigram translation probability, which relies on word context, has been known to help in translation [3] to improve the identification of target phrases. Word order within a phrase is captured by bigram or higher-order models. As an example, "世贸组织" is the Chinese word for "World Trade Organization". Table 3.1 shows how the likelihood of the correct English phrase is improved with bigram translation probabilities.

Table 3.1: An example showing that bigram translation probabilities may assign higher likelihood to correct phrase translations than unigram probabilities by utilizing word context information within a phrase.

Model	unigram	bigram
P(World|世贸组织)	0.06	0.06
P(Trade|World, 世贸组织)	0.06	0.99
P(Organization|Trade, 世贸组织)	0.06	0.99
P(World Trade Organization|世贸组织, 3)	0.0002	0.0588

The basic model components described in section 3.3.1 leave much room for refinements. It is straightforward to extend Markov process to higher order. Moreover, it is also possible to make transition probabilities depend on actual source words or their classes [70]. This will allow modeling of source word context. Currently, target phrase lengths are determined by source words. When the max phrase length N increases, more parameters need to be estimated. Making the phrase length ϕ dependent on classes of source words would allow statistics to be shared among linguistically similar source words and lead to robust parameter estimation as well. Source word classes

can be learned from monolingual corpora with clustering algorithms or from bilingual data [69].

3.4 Summary

We have discussed HMM-based word-to-word alignment models and IBM fertility-based models, identifying their model components and analyzing their strengths and weaknesses. Building on the strengths of each, we have proposed and developed an HMM-based Word-to-phrase alignment model. The model architecture is inspired by features of Model-4, such as fertility and distortion, but care is taken to ensure that dynamic programming procedures, such as EM and Viterbi alignment, can still be performed. There is practical value in this: training and alignment are easily parallelized.

Chapter 4

Statistical Phrase Alignment Models

4.1 Introduction

It has been shown that phrase-based machine translation outperforms word-based machine translation [68]. Phrase-based statistical machine translation systems typically require a phrase translation table, which provides a list of foreign translations and their probabilities for English phrases. A typical translation process would segment input foreign sentences into phrases, translate each foreign phrase into English, and finally reorder English phrases to produce output [40] [42]. Phrases can be inserted or deleted during translation.

Phrase-based translation offers several advantages to word-based translation. First, it naturally captures local context and uses that context in the translation. Secondly, because a phrase is defined as a consecutive sequence of words and is not necessarily limited to linguistically motivated segments, phrase-based models allow the translation of non-compositional phrases [53], for example "kick the bucket", "chew the fat". Finally, phrase translations are learned from data in an unsupervised way. Linguistic resources are not required to assist the learning procedure, which makes the methodology generally applicable to any language pairs, and so the more data in the training corpora, the longer the phrase translations that can be learned.

Statistical phrase translation models are usually induced from word alignments [40] [68] [80]. A static word aligned bitext is formed first, and phrase pairs are extracted according to heuristics. In [40], word alignments within a phrase pair are ignored, while in [68], phrase pairs are replaced by alignment templates, which are generalized and alignment-annotated phrase pairs. Alignment templates model alignments of word classes rather than words. Consequently, the phrase translation table is small with reliable statistics. However, this can complicate the decoding procedure.

Phrase translation models can be learned directly from phrase alignment models [57] [94]. In the joint phrase model proposed by Marcu and Wong [57], the generative procedure creates a number of concepts first; then, each concept generates a foreign and English phrase; finally, the English phrases are reordered. A concept can be understood as abstraction of phrase types. Both foreign and English sentences are generated jointly. Phrase translation distribution is internally part of the model.

Phrase pairs can also be found by sequential pattern mining algorithms from parallel strings through co-occurrence analysis [89]. A phrase does not have to be a sequence of consecutive words; instead, gaps are allowed.

Rather than a pool of phrase-to-phrase translation entries, phrase translation models can be in the form of synchronous context-free grammar [13]. Phrases are presented in a hierarchial format: they can contain words and subphrases. The model uses phrases in a higher level to reorder phrases in a lower order; therefore, no particular phrase reordering mechanism is required.

4.2 Word Alignment Induced Phrase Translation Models

In this work, we extract phrase pairs based on word alignments and their models. We will discuss the three steps in constructing a statistical phrase translation model: (1) establishing static word alignments for training bitexts; (2) using a phrase extracting procedure to create a pool of phrase pairs; a phrase pair inventory (PPI); (3)

the building of the phrase translation table. We then present the model-based phrase pair posterior and show its value in augmenting PPI for a better translation system.

4.2.1 Word Alignments

Statistical word alignment models produce asymmetrical word alignments from source strings to target strings; they consist of only one-to-many alignments. In real data, many-to-many are naturally observed. To compensate for the model assumption and improve word alignment quality, statistical word alignment models, say Model-4 or Word-to-Phrase HMM, are usually trained in both translation directions, e.g. F→E and E→F; and then two sets of word alignments $A_{F\to E}$ and $A_{E\to F}$ are generated by the Viterbi algorithm for each set of models.

The intersection of the two alignments $A_I = A_{E\to F} \cap A_{F\to E}$ yields high precision of word-to-word alignment but low recall. On the contrary, the union of the two alignments $A_U = A_{E\to F} \cup A_{F\to E}$ shows high recall but low precision. To achieve a balance between precision and recall, starting from A_I, word links in A_U but not in A_I are added to the final alignment set if they satisfy some heuristic constraints [40] [70].

The heuristic postprocessing step combines the word alignments in two translation directions. The final alignment set falls between the intersection A_I and the union A_U. It has been shown that a higher recall is more important for statistical machine translation [70]. In our experiments, we simply merge the alignments in two directions and form a static word aligned bitext before phrase pair extraction.

4.2.2 Phrase Pair Extraction

In the *phrase-extract* algorithm [65], phrase pairs are learned from word alignments. Phrases align if their words align. There is a hard constraint in the algorithm: words within the phrase pairs are not allowed to align to words outside the phrase pairs. More formally, let $A = \{(i, j)\}$ be the word alignment matrix, $(e_{i_1}^{i_2}, f_{j_1}^{j_2})$ form a phrase pair if for an $(i, j) \in A : i_1 \leq i \leq i_2$ iif $j_1 \leq j \leq j_2$. This definition applies to a general word alignment matrix.

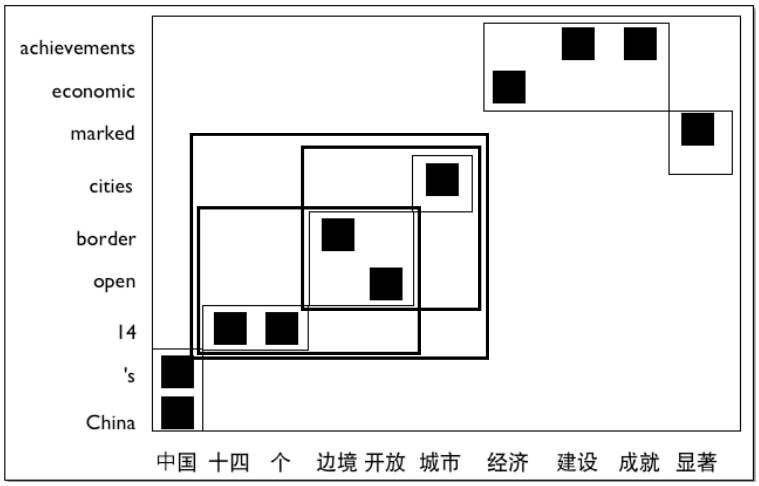


Figure 4.1: Phrase pair extracting based on word alignments

All phrase pairs up to a certain length on foreign side [1] within bitext can be derived using the *phrase-extract* algorithm. We refer to the result as Viterbi Phrase-Extract PPI. As Figure 4.1 shows, each black dot represents a word link. Each block is a phrase pair.

4.2.3 Phrase Translation Table

Phrase pair extracting produces a pool of English and foreign phrase pairs. Once the PPI phrase pairs are set, the phrase translation probabilities can be defined as the number of times each phrase pair is extracted from a sentence pair, i.e. from relative frequencies. Let $c(\mathbf{u}, \mathbf{v})$ be the count that the English phrase $\mathbf{u}$ paired with the foreign phrase $\mathbf{v}$, the maximum likelihood estimation is given by the relative frequency without smoothing.

$$P^{(ML)}(\mathbf{v}|\mathbf{u}) = \frac{c(\mathbf{u}, \mathbf{v})}{\sum_{\mathbf{v}'} c(\mathbf{u}, \mathbf{v}')} \tag{4.1}$$

The alternative way to assign the translation probability is by lexical weighting

[1] Due to memory and computational constraints, the maximum phrase length has been chosen to be 5.

[40]. Let $\mathbf{u} = e_1^n$ be the English phrase and $\mathbf{v} = f_1^m$ the foreign phrase. Let $\mathbf{a} = \{(i,j) : 0 \leq i \leq n, 1 \leq j \leq m\}$ be a set of word links between the phrase pairs. The lexical weighting is given by:

$$P^{(LW)}(\mathbf{v}|\mathbf{u},\mathbf{a}) = \prod_{j=1}^{m} \frac{1}{|\{i|(i,j) \in \mathbf{a}\}|} \sum_{i:(i,j)\in\mathbf{a}} t(f_j|e_i) \tag{4.2}$$

When there are multiple word alignments $\mathbf{a}$ between the phrase pairs, the highest lexical weight is used:

$$P^{(LW)}(\mathbf{v}|\mathbf{u}) = \max_{\mathbf{a}} P^{(LW)}(\mathbf{v}|\mathbf{u},\mathbf{a}) \tag{4.3}$$

4.2.4 Discussion

We have discussed how a phrase translation table can be created from word aligned bitexts. Constructed in this way, the PPI is limited to phrase pairs which can be found in the static word alignment set. Figure 4.2 shows an example where the 5th Chinese word is incorrectly aligned to the second "politics" in the English sentence. Consequently, a perfect phrase pair covered by the dot-dash line in the figure is not identified by the *phrase-extract* algorithm.

To avoid this situation, caused by the hard word alignment constraint, a metric is needed, indicating how strongly any phrase pairs within a parallel sentence pairs are aligned. For example, Venugopal *e*t al [80] assigns scores to phrase pairs with information from word alignments and a translation lexicon as a confidence measurement to identify the phrase translation hypothesis. We next define a probability distribution over phrase pairs which allows more control over the generation of phrase pairs and enables alternative phrase translation extraction strategies.

4.3 Model-based Phrase Pair Posterior

Given a sentence pair $(\mathbf{s},\mathbf{t})$, and the word alignment model θ, we define and compute the posterior of the target phrase $t_{j_1}^{j_2}$ aligned to the source phrase $s_{i_1}^{i_2}$.

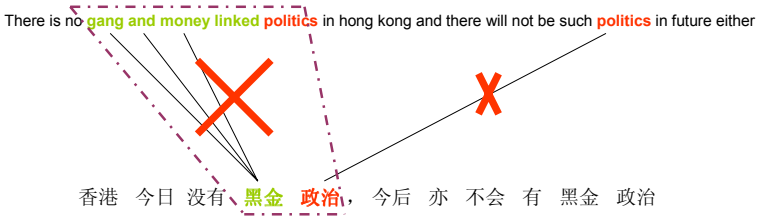


Figure 4.2: An example of Model-4 word alignments showing that incorrect word alignments prevent perfect phrase pairs from being extracted.

There are I^J possible word alignments from the source string $\mathbf{s}$ to $\mathbf{t}$, ignoring empty word alignments for simplicity. Among these, we are interested in word alignments which link the words in the target phrase $t_{j_1}^{j_2}$ to the words in the source phrase $s_{i_1}^{i_2}$, and other target words in $\mathbf{t}$ to other source words in $\mathbf{s}$. More formally, we concern ourselves with and define the following set of word alignments:

$$A(i_1, i_2; j_1, j_2) = \left\{ \mathbf{a} = a_1^J : a_j \in [i_1, i_2] \text{ if and only if } j \in [j_1, j_2] \right\} \tag{4.4}$$

The likelihood of the target phrase aligned to the source phrase is obtained by considering all "valid" alignments in the set:

$$P(\mathbf{t}, A(i_1, i_2; j_1, j_2)|\mathbf{s}; \theta) \quad = \quad \sum_{\mathbf{a} \in A(i_1, i_2; j_1, j_2)} P(\mathbf{t}, \mathbf{a}|\mathbf{s}; \theta) \tag{4.5}$$

Applying the Bayes rule, we obtain the phrase pair posterior

$$P(A(i_1, i_2; j_1, j_2)|\mathbf{s}, \mathbf{t}; \theta) \quad = \quad P(\mathbf{t}, A(i_1, i_2; j_1, j_2)|\mathbf{s}; \theta) / P(\mathbf{t}|\mathbf{s}; \theta). \tag{4.6}$$

The sentence translation probability in the denominator considers all possible word alignments $P(\mathbf{t}|\mathbf{s}; \theta) = \sum_{\mathbf{a}} P(\mathbf{t}, \mathbf{a}|\mathbf{s}; \theta)$. Therefore, the posterior defined in this way is normalized between 0 and 1.

The phrase pair posterior definition (Equ. 4.6) applies to any statistical word alignment model. Whether the posterior computation is tractable depends on the complexity of the model.

4.3.1 IBM Model 1 & 2

Calculating phrase pair posterior under IBM Model-1 and Model-2 has a closed form solution. Let $P_\theta(i|j, \mathbf{t}, \mathbf{s})$ be the posterior of $a_j = i$ (Equ. 3.2). Since in Model 1 and 2 there is no dependency between a_j's, the posterior can be directly computed.

Let $J_2 = \{j_1, j_1 + 1, \cdots, j_2\}$ be the set of word index of concerned target phrase, and $J_1 = \{1, 2, \cdots, j_1 - 1, j_2 + 1, \cdots, m\}$ the set of other target words. Let $I_2 = \{0, i_1, i_1 + 1, \cdots, i_2\}$ be the set of word index of concerned source phrase plus NULL word, and $I_1 = \{0, 1, 2, \cdots, i_1 - 1, i_2 + 1, \cdots, l\}$ the set of other source words plus NULL word; then the posterior is easily found:

$$P(A(i_1, i_2; j_1, j_2)|\mathbf{s}, \mathbf{t}; \theta) = \prod_{j \in J_1} \sum_{i \in I_1} P_\theta(i|j, \mathbf{t}, \mathbf{s}) \times \prod_{j \in J_2} \sum_{i \in I_2} P_\theta(i|j, \mathbf{t}, \mathbf{s}) \tag{4.7}$$

4.3.2 IBM Fertility-based Models

We point out that finding phrase pair posteriors under IBM fertility-based models, for instance Model-4, faces the same challenge as that of parameter estimation. There is no efficient way to calculate the likelihood of the source phrase generating the target phrase as defined in Equ. (4.5).

4.3.3 HMM-based Alignment Models

Computing phrase pair posteriors under HMM-based alignment models, for both word-to-word and word-to-phrase, has an efficient implementation using a modified Forward algorithm. During the forward procedure, when computing the phrase pair likelihood (Equ. 4.5), the recursive definition needs to follow the word alignment constraint defined in the alignment set (Equ. 4.4). For each target word, the alignment set A in Equation (4.4) specifies the source words that it is allowed to align. We define a test function $f_j(i, \phi)$ which is 1 if the target word sequence $t_{j-\phi+1}^{j}$ is allowed to align to the source word s_i under the alignment set A, and 0 otherwise. A modified Forward procedure can be implemented to calculate the phrase generating probability

(Equ. 4.5) under the WtoP model with the recursive definition as follows:

$$\alpha_j(i,\phi,h) = \begin{cases} 0 & f_j(i,\phi) = 0 \\ \Big\{ \sum_{i',\phi',h'} \alpha_{j-\phi}(i',\phi',h') p(i|i',h;I) \Big\} \eta n(\phi; h \cdot s_i) & \\ \cdot t(t_{j-\phi+1}|h \cdot s_i) \cdot \prod_{j'=j-\phi+2}^{j} t_2(t_{j'}|t_{j'-1}, h \cdot s_i) & f_j(i,\phi) = 1 \end{cases}$$

There is a tradeoff between model performance and computational efficiency in IBM fertility-based models. However, the Word-to-Phrase alignment model does not appear to suffer from this tradeoff - it is a good model of word alignment under which statistics such as the phrase-to-phrase posterior can be calculated.

4.4 PPI Induction Strategy

As we have mentioned before, due to word alignment errors, there are foreign phrases which do appear in the training bitext which will not be included in the PPI because suitable English phrases cannot be found by the heuristic alignment of search [65]. To add these to the PPI, we use the phrase-to-phrase posterior distribution to find English phrases as candidate translations. This augments phrases to the Viterbi Phrase-Extract PPI and can increase the test set coverage. A somewhat *ad hoc* PPI Augmentation algorithm is given below.

For each foreign phrase v not in the Viterbi PPI :

For all pairs (f_1^m, e_1^l) and j_1, j_2 s.t. $f_{j_1}^{j_2} = v$:

For $1 \le i_1 \le i_2 \le l$, find:

$$b(i_1,i_2) = P_{F\to E}(\, A(i_1,i_2;j_1,j_2) \,|\, e_1^l, f_1^m)$$

$$f(i_1,i_2) = P_{E\to F}(\, A(i_1,i_2;j_1,j_2) \,|\, e_1^l, f_1^m)$$

$$g(i_1,i_2) = \sqrt{f(i_1,i_2)\, b(i_1,i_2)}$$

$$(\hat{i}_1,\hat{i}_2) = \underset{1\le i_1,i_2\le l}{\operatorname{argmax}}\ g(i_1,i_2) \text{ , and set } u = e_{\hat{i}_1}^{\hat{i}_2}$$

Add (u,v) to the PPI if any of A, B, or C hold :

$b(\hat{i}_1,\hat{i}_2) \ge T_g$ and $f(\hat{i}_1,\hat{i}_2) \ge T_g$ (A)

$b(\hat{i}_1, \hat{i}_2) < T_g$ and $f(\hat{i}_1, \hat{i}_2) > T_p$ (B)

$f(\hat{i}_1, \hat{i}_2) < T_g$ and $b(\hat{i}_1, \hat{i}_2) > T_p$ (C)

PPI Augmentation via Phrase-Posterior Induction

Condition (A) extracts phrase pairs based on the geometric mean of the E→F and F→E posteriors ($T_g = 0.01$ throughout). The threshold T_p selects additional phrase pairs under a more forgiving criterion: as T_p decreases, more phrase pairs are added and PPI coverage increases. A balance between coverage and phrase translation quality can be achieved by setting up thresholds properly. Note that this algorithm is constructed specifically to improve a Viterbi PPI; it is certainly not the only way to extract phrase pairs under the phrase-to-phrase posterior distribution.

4.5 Summary

Word alignment induced phrase translation models have been introduced. We discussed how to form a static word aligned bitext, extract phrase pairs from those alignments, and build the phrase translation table with maximum likelihood estimation and lexical weighting. We discussed the limitation of the quality of word alignments to possible phrase pairs that can be extracted in the training bitext and in turn proposed a model-based phrase pair posterior distribution that enables alternative phrase translation extraction strategies. We devised a simple augmenting strategy that aims to improve phrase coverage on test sets by using the phrase pair posterior. With the definition under the Word-to-Phrase HMM alignment model, the posterior can be calculated efficiently with DP-based implementation, whereas under Model-4 it is intractable.

Chapter 5

Experimental Results of Bitext Chunk Alignment

In Chapter 2, we formally presented the statistical chunk alignment model and developed a hybrid modeling approach. In this chapter we report the results of experiments in bitext word alignment and statistical machine translation whose purpose is to investigate the behavior and evaluate the quality of the alignment procedures we have proposed. We begin with a description of the training and test data.

5.1 Corpora

5.1.1 Chinese-English

The training corpora for Chinese-English systems are from the LDC collections [55]. They consist of FBIS [49], Hong Kong Hansards [50], Hong Kong News [51], XinHua News [47], Sinorama [46], Chinese Treebank [48], and UN chapters [52] from the year 1993 to 2002. The number of documents and English words for each corpus are tabulated in Table 5.1. Before bitext chunking, Chinese documents are segmented into words using the LDC segmenter [45].

The test sets for Chinese-English systems are the NIST Machine Translation 2001, 2002, 2003 and 2004 evaluation sets [63]. The task is to translate Chinese sentences

Table 5.1: Statistics of Chinese-English Parallel (document pairs) Corpora

Corpus	# of Document Pairs	# of English words (in Millions)
FBIS	11,537	10.71
HongKong Hansards	713	39.87
HongKong News	44,649	17.04
XinHua News	19,140	4.13
Sinorama	2,373	3.78
Chinese Treebank	325	0.13
UN (1993∼2002)	44,248	144.43
Total	122,985	220.09

Table 5.2: Statistics of NIST Chinese-English MT Evaluation Sets

test set	# of documents	# of sentences
Eval01	105	993
Eval02	100	878
Eval03	100	919
Eval04	200	1788

into English sentences. Table 5.2 shows the number of documents and total Chinese sentences for each test set. There are four English reference translations for each Chinese sentence in these sets.

In Eval01∼03, the test documents are "news" stories drawn from several kinds of sources, including newswire, broadcast news, and the web. Eval04 contains three types of genres, including "speech" and "editorial" pieces in addition to "news" stories (there are 100 "news" documents with 901 Chinese sentences).

For bitext word alignments, we use the test set of the NIST 2001 dry-run MT-eval set [63], which consists of 124 parallel Chinese/English sentences. Chinese sentences are segmented into words manually. All sentence pairs are word aligned manually.

5.1.2 Arabic-English

The training corpora for Arabic-English systems are also taken from the LDC collections [55]. They consist of News corpora, and UN chapters from the year 1993 to 2002. The number of documents and English words for each corpus are tabulated

Table 5.3: Statistics of Arabic-English Parallel (document pairs) Corpora

Corpus	# of Document Pairs	# of English words (in Millions)
News	10,265	3.59
UN (1993∼2002)	-	131.38
Total	-	134.97

Table 5.4: Statistics of NIST Arabic-English MT Evaluation Sets

test set	# of documents	# of sentences
Eval02	141	1043
Eval03	100	663
Eval04	200	1353

in Table 5.3. The UN corpora are already sentence aligned. Before bitext chunking, all Arabic documents are preprocessed by a modified Buckwalter tokenizer [44].

The test sets for Arabic-English systems are the NIST Machine Translation 2002, 2003 and 2004 evaluation sets [63]. The task is to translate Arabic sentences into English sentences. Table 5.4 shows the number of documents and total Arabic sentences for each test set. There are four English reference translations for each Arabic sentence in these sets.

Like the Chinese-English systems in Eval02∼03, the test documents also include "news" stories drawn from several kinds of sources, such as newswire, broadcast news, and the web. In Eval04, there are three types of genres, including "speech" and "editorial" pieces in addition to "news" stories (there are 100 "news" documents with 707 Arabic sentences).

5.2 Unsupervised Bitext Sentence Alignment

Our initial experiments investigate the quality of automatic sentence alignments produced by different model configurations and alignment strategies. We use a collection of 122 document pairs selected at random from the FBIS Chinese/English parallel corpus [63]; the Chinese sentences were segmented using the LDC word seg-

menter [45]. The documents were aligned at the sentence level by bilingual human annotators, resulting in a collection of 2,200 aligned Chinese-English sentence pairs.

These human alignments serve as the reference against which the quality of automatically generated alignments is measured. Both alignment precision and alignment recall relative to the human references will be reported, and of these results, only exactly corresponding alignments will be counted as correct. For instance, a many-to-one alignment will not be judged as correct even if it covers a one-to-one reference alignment.

5.2.1 Monotonic Sentence Alignment Using Sentence Length Statistics

We generate initial sentence alignments using the monotonic dynamic programming procedure described in Section 2.3.1. In this, as well as in the other experiments described in this section, the boundary markers are defined so that the chunking and alignment procedures operate at the sentence level.

The initial alignment is based on sentence length statistics, i.e. with flat Model-1 word translation tables. The global length ratio c in Equation 2.6 is set based on document-level statistics: we count the total number of words in the Chinese and English documents and set c to be their ratio. We also set the parameters of Equation 2.3 to be $\lambda = 3.0$ and $\alpha = 0.9$; these were found to be generally robust values in experiments not reported here. These parameters, c, λ, and α, are all that is needed to perform sentence alignment under Equation 2.12. The resulting sentence alignment precision and sentence alignment recall are 81% and 83%, shown as Iteration 0 of Table 5.5.

5.2.2 Iterative Alignment and Translation Table Refinement

We use these initial length-based sentence alignments as the basis for more refined alignments [91]. Since this alignment procedure is 'self-organizing' and does not make use of any sentence aligned training data, we adopt a strategy that uses the model

Table 5.5: Bitext used at Each Iteration of Unsupervised Sentence Alignment. At Iteration 0, the entire 122 document bitext is used. At iterations 1 through 4 the chunk pairs found at the previous iteration are sorted by likelihood and only those with likelihoods above the specified threshold are retained for the estimation of the Model-1 translation table.

Iteration	Threshold	Surviving Chunk Count	Total Words (Ch/En)
0	-	-	64K/86K
1	0.8	1278	34K/44K
2	0.005	1320	35K/45K
3	0.001	1566	42K/55K
4	0.001	1623	44K/58K

to produce a reliably aligned subset of the training data. From the aligned pairs we selected those with likelihoods higher than 0.8 under Equation 2.6. Approximately 58% of the initial alignments (44K English words / 34K Chinese words) survive this filtering.

With these aligned sentences, we can use the EM algorithm to refine the IBM Model-1 translation lexicon; eight iterations of EM are performed to train Chinese-to-English distribution $t(\cdot|\cdot)$. With these distributions incorporated into Equation 2.5, replacing the flat translation table as used in Section 5.2.1, monotone sentence alignment performance over the entire corpus increases in both precision and recall by approximately 4% relative to the initial length-based sentence alignments.

This forms the basis for an unsupervised sentence alignment procedure that allows us to iteratively refine the translation tables. We relax the inclusion threshold over the likelihood of aligned sentence pairs (Equation 2.5), which gradually increases the size of the bitext used to estimate the translation tables.

After each reduction in the threshold, we re-estimate the Model-1 translation table using eight iterations of EM. Table 5.5 shows the amount of bitext incorporated at each stage, and the corresponding sentence alignment precision and sentence alignment recall are plotted in Figure 5.1, marked with line 'E'; for iterations 5 and 6, no filtering is performed and the entire bitext is used.

5.2.3 Length Distributions, Divisive Clustering, and Alignment Initialization

We now investigate the main components of the sentence alignment procedures. These alignment results and search configurations are detailed in Figure 5.1. Each alignment iteration that succeeds iteration 0 involves eight EM iterations to estimate the Model-1 Chinese-to-English word translation tables; in each scheme, the aligned chunks are filtered at each iteration following the schedule of Table 5.5.

These procedures are initialized 'naturally': for example, Procedure A is initialized by monotonic sentence alignment based on sentence-length statistics with $\lambda = 0$, and Procedure C is initialized by a single binary split also based on sentence-length statistics. Procedures 'F' and 'G', which incorporate uniform chunk length distributions, are exceptions; they are initialized with the translation table produced by the first iteration of DP+DC($\lambda = 3.0$, $\alpha = 0.9$).

We first note that the hybrid search procedure of monotonic sentence alignment to produce coarse alignments that are subsequently refined by division clustering (DP+DC) is the procedure that produces the best overall sentence alignments in terms of sentence alignment precision and recall. Performance is sensitive to the chunk length distribution, and performance suffers if flat length distribution is used. Monotone alignment (DP) performs nearly as well under the informative length distribution, although the final alignment recall is slightly worse than that of the DP+DC procedure.

Iterative binary search as a stand-alone alignment algorithm (DC) performs relatively poorly, although it does improve with iterative translation table refinement. Comparison of plots C and G shows that DC alignment is extremely sensitive to initialization, which is not surprising given the suboptimal nature of its search.

We observe that in nearly all cases the precision and recall increase steadily as the iterations proceed. Procedures 'E' and 'F' show the influence of a good translation table on alignment. When initialized with a translation table estimated under slightly stronger models (models with non-uniform chunk-length distributions) the DP+DC and DC procedures both perform better, even when limited by uniform chunk-length

Table 5.6: Performance of Sentence Alignment Procedures Over the FBIS Sentence-Alignment Corpus. Procedures *a, b, c* are unsupervised; Champollion is provided with a Chinese-English translation lexicon; the 'Oracle' version of DP+DC uses Model-1 translation tables trained over the human-aligned sentences.

Alignment Procedure		Precision	Recall
a	Gale-Church	0.763	0.776
b	Moore'02	0.958	0.441
c	DP+DC($\lambda = 3.0$, $\alpha = 0.9$)	0.901	0.910
d	Champollion	0.937	0.940
e	DP+DC($\lambda = 3.0$, $\alpha = 0.9$) Oracle	0.960	0.971

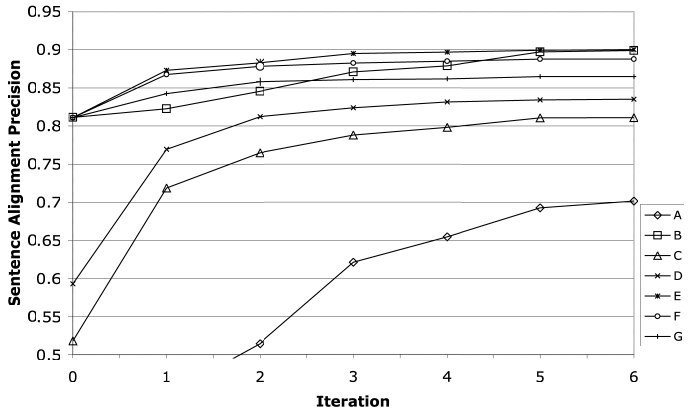
distributions. That they perform very similarly implies that when a reasonable translation lexicon is incorporated in alignment, the chunk length distribution plays less of a role. Loosely speaking, the translation table is more important than the chunk length distributions.

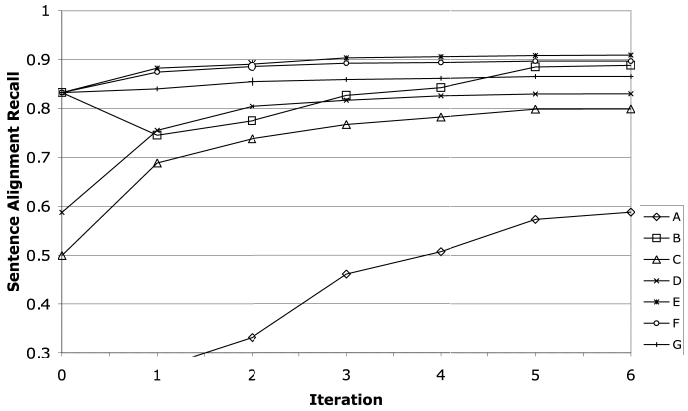
The results of Figure 5.1 are obtained in an unsupervised manner. No linguistic knowledge is required. This is important and useful in circumstances where linguistic resources such as a bilingual dictionary are not available. We note also that the alignment procedure achieves a good balance between precision and recall.

5.2.4 Comparable Sentence Alignment Procedures and Performance Upper Bounds

To place these results in context we present the sentence alignment performance obtained on this task by several other well-known algorithms, namely the sentence alignment procedures of Moore [62], Gale-Church [26], and the Champollion Toolkit [55]. The results are shown in Table 5.6. The Champollion aligner requires a bilingual dictionary; we use the 41.8 K entry Chinese-English dictionary distributed with the toolkit.

To estimate an upper bound on the performance that might be achieved by sentence alignment procedures based on word-to-word translation, we take the sentence pairs as aligned by humans and use them in estimating the IBM Model-1translation





Plot	Alignment Procedure	Length Distribution Parameters (DP)
A	DP	$\lambda = 0.0$ (uniform)
B	DP	$\lambda = 3.0$, $\alpha = 0.9$
C	DC	n/a
D	DP+DC	$\lambda = 0.0$ (uniform)
E	DP+DC	$\lambda = 3.0$, $\alpha = 0.9$
F	DP+DC	$\lambda = 0.0$ (uniform)
G	DC	n/a

Figure 5.1: Precision and Recall of Automatically Sentence Alignment Procedures Over the FBIS Sentence-Alignment Corpus with Different Initialization and Search Strategies. Alignment procedures 'F' and 'G' were initialized from iteration 0 of the DP+DC($\lambda = 3.0$, $\alpha = 0.9$) alignment procedure.

table. We then align the whole collection under this model using one iteration of the DP+DC procedure. This translation table is very heavily biased towards this particular corpus; for instance, many translations that would normally appear within the translation table, for example those due to different word senses, will not be present unless they happen to occur within this small bitext sample. We therefore call this the 'Oracle' DP+DC condition, and it yields a precision of 96% and a recall of 97%. The upper bound confirms that the IBM Model 1 can indeed be useful for sentence alignment tasks, although this claim must be qualified as above, noting that the translation tables are refined for the task. However, it is clear that an interdependence exists between sentence alignment quality and the translation lexicon, and if we use a translation lexicon estimated over human-aligned sentences, we can obtain better sentence alignment.

5.3 Evaluation via Statistical Machine Translation

Statistical Machine Translation (SMT) systems rely on high quality sentence pairs for training translation models [7, 65, 42]. We now present experiments to evaluate the influence of bitext chunking alignment algorithms on word alignment and translation performance in SMT.

The Translation Template Model [42] 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. We utilize the *phrase-extract* algorithm [65] to extract a library of phrase-pairs from bitext word alignments. We first obtain word alignments of chunk-pairs using IBM-4 word level translation models [7] trained in both translation directions (IBM-4 F and IBM-4 E), and then we form the union of these alignments (IBM-4 $E \cup F$). Next, we use the algorithm to identify pairs of phrases in the target and source language that align well according to a set of heuristics [65]. We will report the word alignment performance of the underlying IBM-4 models and the translation performance of the TTM system initialized from these models.

5.3.1 Bitext Chunking and Alignment

We present experiments on the NIST Chinese-to-English Translation task [63]. The goal of this task is the translation of news stories from Chinese to English. The bitext used for model parameter estimation is the FBIS Chinese-English parallel corpus [63].

As in the previous section, we investigate the DP+DC hybrid alignment procedure. We align the bitext by first performing monotonic alignment (DP) under the sentence length model (Equation 2.6) ($\lambda = 3.0$, $\alpha = 0.9$). In this stage we consider only end-of-sentence marks as segment boundaries and insist that each chunk contain at most 4 sentences in either language. From the resulting aligned chunks, we select those chunk pairs with a maximum of 100 words in their English and Chinese segments; chunks with longer segments are discarded. This yields an aligned bitext collection of 7.5M Chinese words and 10.1M English words; approximately 10% of the bitext is discarded. Each aligned chunk pair contains 28 Chinese words and 38 English words on average; see entry 1 of Table 5.7. We next apply divisive clustering to the chunk pairs obtained by DP. In this step, we consider all punctuations, such as commas and other markers of pauses, as segment boundary markers. This allows for a much finer alignment of sentence segments (Table 5.7, entry 2).

Using the chunk pairs produced by length based model with divisive clustering (Table 5.7, entry 2), we train IBM Model 1 word translation models. Although it departs from the strict model formulation, we have found it beneficial to training IBM Model 1 translation tables in both translation directions, i.e. from English-to-Chinese and from Chinese-to-English. A single translation table is formed by finding $\sqrt{P(t|s)P(s|t)}$ and then normalizing appropriately.

We then repeat the DP and DP+DC procedures incorporating these IBM Model-1 translation tables from Step 2; during DP monotone alignment, we set the parameter $\lambda = 0$ in Equ. (2.3) to allow chunk pairs to align freely.

We observe that the training bitext in system 2 is derived from that of system 1 by divisive clustering. System 2 retains all the bitext aligned by system 1, but produces pairs of shorter chunks. A similar relationship holds between systems 3 and

Table 5.7: Aligned Chunk Pair Statistics Over Contrastive Alignment Configurations. Step 1: initial chunk alignments obtained by DP monotone alignment using sentence length statistics. Step 2: divisive clustering of aligned chunks from Step 1 under sentence-length statistics. The aligned chunks at Step 2 are used in training a Model-1 translation table; this table is held fixed for Steps 3 and 4. Step 3: chunk alignments obtained by DP monotone alignment using Model-1 translation table. Step 4: divisive clustering of aligned chunks from Step 1 under Model-1 translation table.

	Alignment Procedure	Chunk Translation Model	Words (M) Ch/En	Average words per chunk Ch/EN	IBM-4 Training Time (CPU hrs)
1	DP	length-based	7.5/10.1	28/38	20
2	DP+DC	length-based	7.5/10.1	20/27	9
3	DP	Model 1	7.2/9.7	29/40	21
4	DP+DC	Model 1	7.2/9.7	16/22	8

4. We will use the aligned bitext collections produced by these alignment strategies in training SMT systems that will be used for word alignment and translation.

5.3.2 Bitext Word Alignment and Translation Performance

For each collection of bitext produced by the four alignment strategies, we use the GIZA++ Toolkit [67] to train IBM-4 translation models [7] in both translation directions. The IBM-4 training time is also displayed in Table 5.7. We observe that after applying DC, average chunk size on both sides is reduced, which significantly speeds up the MT training procedure. This is an extremely valuable practical benefit of divisive clustering at the subsentence level relative to monotone sentence alignment.

We now measure the word alignment performance of the resulting IBM-4 word translation models. Our word alignment test set consists of 124 sentences from the NIST 2001 dry-run MT-eval set [63] that are word aligned manually, and word alignment performance is measured using the Alignment Error Rate (AER) metric [67]. For each system described in Table 5.7, Table 5.8 shows the AER of IBM-4 models trained in both translation directions. We observe that the chunk pairs extracted using IBM Model 1 translation tables in bitext alignment yield lower AER than the sentence length based alignment procedures. We also note that in some cases, divisive

Table 5.8: Word Alignment and Translation Performance Corresponding to IBM-4 Models Estimated over Bitext Collections Produced by Contrastive Alignment Configurations. Alignment Error Rates are provided in both translation directions. Translation performance is given as BLEU(%) scores of phrase-based SMT systems based on phrases extracted from the word aligned bitext.

Collection	Alignment Error Rate (%)		Translation Performance		
	$E \rightarrow C$	$C \rightarrow E$	Eval01	Eval02	Eval03
1	38.6	35.3	25.1	23.1	22.2
2	38.1	35.1	24.7	23.1	22.1
3	38.0	33.6	25.3	23.3	22.3
4	37.1	33.8	25.1	23.5	22.7

clustering yields some minor improvement relative to monotonic sentence alignment, and that performance is otherwise comparable.

We next measure translation performance of a TTM system trained on the four bitext collections. We report performance on the NIST 2001, 2002 and 2003 evaluation sets, and translation performance is measured using the BLEU metric [72].

We use a trigram word language model estimated using modified Kneser-Ney smoothing, as implemented in the SRILM toolkit[1]. 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*[2] (16.9M words) and the English side of the Xinhua Chinese-English parallel corpus [63] (4.3M words). The total language model corpus size is 21M words.

For each of the word-aligned bitext collections, we show the translation performance of the phrase-based SMT system built on the word alignments (Table 5.8). We observe that the IBM Model-1 yields improvements over the length-based model on every one of the test sets. Divisive clustering yields performance comparable to that of sentence level alignment, but with greatly reduced training times. We conclude that the DP+DC procedure has practical benefits relative to sentence-length based alignment.

[1]http://www.speech.sri.com/projects/srilm/
[2]http://www.english.people.com.cn

5.4 Maximizing the Aligned Bitext Available for Training

The controlled experiments in Section 5.3 show that applying divisive clustering to derive shorter chunk pairs significantly reduces MT training time while still maintaining MT system performance as evaluated by bitext word alignment and translation performance. In this section, we show another advantage of the two stage bitext chunking procedure: its ability to align almost all available bitext available for training MT systems. We present experiments on an Arabic-English MT system to show that bitext chunking makes the most of available bitext usable for MT training, demonstrating how this can effectively improve system performance.

The bitext used for this experiment includes all document pairs of news and UN parallel corpora released by LDC [63]. We set the maximum number of tokens on both Arabic and English sides to be 60 in GIZA++ model training. If any side has more than 60 tokens or chunk pair length ratio is more than 9, the sentence pairs would have to be discarded. These are practical constraints which prevent the GIZA++ training procedure from running out of memory.

Two iterations of the monotonic DP sentence alignment algorithm (as in Figure 5.1, plot E) are applied to the Arabic-English document pairs. In the sentence aligned bitext that results, we find that about 60% and 74% (in terms of English tokens) of all bitext can be used in training for the News and UN corpora, respectively (Table 5.9). This is simply because Arabic sentences tend to be very long. We then apply divisive clustering to these sentence pairs. On the English side, all punctuation marks are considered as boundary markers. On the Arabic side, two boundary marker sets are investigated. In one configuration (DP+DC(I)), punctuation serves as boundary marks; in the second configuration (DP+DC(II)), all Arabic tokens are considered as potential boundaries, i.e. white space is used as boundary marks.

When applying DC with boundary definition DP+DC(I), the statistics of Table 5.9 show that relatively little aligned bitext is extracted relative to the initial sentence alignment. However, under the more aggressive segmentation scheme of DP+DC(II),

Table 5.9: Percentage of Usable Arabic-English Bitext. English tokens for Arabic-English news and UN parallel corpora under different alignment procedures.

Bitext	DP	DP+DC(I)	DP+DC(II)
News	60%	67%	98%
UN	74%	78%	98%

Table 5.10: Translation Performance of TTM Arabic-English Systems Based on Bitext Collections Extracted by the Alignment Procedures.

Bitext	Alignment Procedure	Eval02	Eval03	Eval04
News	DP+DC(I)	33.00	35.43	32.31
	DP+DC(II)	33.86	36.06	32.79
News+UN	DP+DC(I)	35.39	37.41	33.71
	DP+DC(II)	35.81	37.82	34.02

almost all available bitext can be extracted and aligned for use in MT training.

We also show the advantage of having more bitext in statistical machine translation. The test sets are NIST 2002, 2003 and 2004 Arabic/English MT evaluation sets. As with the Chinese-English MT systems, we perform decoding by the Translation Template Model (TTM). The English language model is a trigram word language model estimated using modified Kneser-Ney smoothing with 266M English words.

Phrase translations are extracted from the News and from the News and the UN collections. Performance of the resulting translation systems are shown on each evaluation set in Table 5.10. Over all test sets, DP+DC(II) performs better than DP+DC(I). Its greater performance is due to retaining bitext in training that otherwise would have to be discarded. We also note that significant improvements over all test sets are obtained when UN bitext is included in model training.

5.5 Improved Subsentence Alignment Can Improve Word Alignment

Word and sentence alignment are typically addressed as distinct problems to be solved independently, with sentence alignment sometimes even regarded as merely as a text preprocessing step to be done prior to word alignment. The two tasks are of course quite different. As discussed here and in earlier work, sentences in parallel documents can be accurately aligned using algorithms based on relatively simple models, such as IBM Model-1. However, word alignment algorithms require more sophisticated alignment models based on Hidden Markov Models [81] or IBM fertility-based models [7]. An intuitive explanation for this difference is that capturing alignment variation in bitext is more challenging as the granularity of the problem becomes smaller. However the interaction between the two types of alignment procedures has not been widely studied.

The experiments reported here investigate the extent to which sub-sentence chunk alignment can improve word alignment. Rather than deriving word alignments directly from manually aligned sentence pairs, we first identify and align chunks at the sub-sentence level and then align the words within the chunks.

There is of course a risk in this approach. If chunks are aligned incorrectly, then some correct word alignments are ruled out from the start, since words cannot be aligned across chunk pairs. In this situation, we say that a word alignment is prevented from crossing a chunk alignment boundary. However, if the automatic chunking procedure does a good job both in deciding where and when to split the sentences, then the sub-sentence aligned chunks may actually help guide the word alignment procedure that follows.

Our training bitext is the complete FBIS Chinese/English parallel corpus, and the test set is the same as that used in the experiments of Section 5.3.2. To generate the Model 1 Chinese-to-English translation lexicons needed by the alignment procedures we run GIZA++[70] with 10 iterations of EM over the training bitext. In aligning the test set, boundary points are set at punctuation marks for both the monotone (DP)

Table 5.11: Influence of Subsentence Alignment on Alignment Error Rate

		Sentence Aligned Test Set	Automatically Aligned Subsentence Chunks	
			DP	DC
Average Ratio of Aligned Segment Lengths (Ch/En words)		24/33	14/19	10/14
	Model-4 Word Alignment Performance			
En→Ch	Precision	67.6	72.2	75.6
	Recall	46.3	48.7	49.6
	AER	45.0	41.8	40.1
Ch→En	Precision	66.5	69.3	72.6
	Recall	59.4	60.4	60.1
	AER	37.3	35.4	34.2

and divisive clustering (DC) alignment procedures. For the DP procedure, we set $\lambda = 3.0$ and $\alpha = 0.9$ and perform chunk alignment as specified by Equation 2.10. In DC alignment, we proceed by Equation 2.14. The recursive parallel binary splitting stops when neither chunk can be split or when one side has less than 10 words and the other side has more than 20 words.

The word alignment performance resulting from these procedures is shown in Table 5.11. We see first that the divisive clustering procedure generates the shortest subsentence segments of all the procedures. We also see that in all instances except one, DC chunk alignment leads to better quality Model-4 word alignment than the other two procedures, and that both subsentence alignment procedures improve the quality of Model-4 word alignments.

This result suggests that the proposed two-stage word alignment strategy can indeed improve word alignment quality relative to the usual word alignment procedure in which word links are established directly from given sentence pairs. To explain where the improvements come from, we inspect the English→Chinese word alignments and analyze the distribution of word links that cross the segment boundaries found by divisive clustering, the most aggressive of the segmentation procedures. The results are presented in Table 5.12.

Table 5.12: English-to-Chinese Word Alignment Links Accuracy Relative to Chunk Alignment Boundaries Found by Divisive Clustering

	Word Alignment Links Relative to DC Chunk Alignment Boundaries	Total Links	Correct Links	Alignment Precision
Manual Word Alignment	Crossing Boundaries	91		
	Within Aligned Chunks	3655		
	All	3746		
DC + Model-4	Within Aligned Chunks	2455	1857	75.6%
Sentence Aligned Test Set + Model-4	Crossing Boundaries	150	34	22.6%
	Within Aligned Chunks	2415	1701	70.4%
	All	2565	1735	67.6%

First, we note that only ∼2.4% of the reference (manual) word alignment links cross the chunk alignment boundaries found by the divisive clustering procedure. This small fraction further confirms that the DC procedure yields good alignments of sub-sentence chunks: nearly all of the manually generated alignment links between the words in the sentences can be found in these chunk pairs. It follows that an automatic word alignment system is not necessarily handicapped if it aligns only these subsentence chunks and not the original sentence pairs.

We now look at the performance of the Model-4 word alignments relative to the chunk alignment boundaries produced by divisive clustering. Obviously, in the DC case, all word alignments that are generated respect these boundaries, and the precision of 75.6% agrees with the result of Table 5.11. When applied to sentence pairs, Model-4 is free to generate word alignments that cross the DC chunk alignment boundaries. However, when it does so, errors are likely: there are 150 cross-boundary links, and most of them (77.3%) are incorrect. In fact, if we remove these cross-boundary links, we can improve the alignment precision to 70.4% and reduce the AER to 44.8% from 45.0%.

However, simply applying Model-4 Viterbi word alignment to the subsentence chunks is more effective than discarding links that cross DC chunk alignment boundaries. The result is a great number of correct word alignment links (1857 vs. 1701),

higher precision (75.6%) and recall (1857/3746 = 49.6%), and a lower overall AER of 40.1%

These results support the notion that the power of the underlying alignment model should be matched to the alignment task to which it is applied. Model-4 is certainly a better word alignment model that Model-1, yet we still find that chunk alignment procedures based on Model-1 can be used to guide Model-4 word alignment. We take this as evidence that, from the point of view of building statistical models, word and sentence alignment are not independent tasks.

5.6 Translation Lexicon Induction

As a final experiment into the properties of these chunk alignment procedures, we evaluate the quality of the probabilistic translation lexicons they produce. Translation lexicons serve as a bridge between languages and thus play an important role in cross lingual applications. For example, statistical methods of extracting lexicons from parallel [59] and non-parallel corpora [56] have been investigated in the literature, and in Section 5.2 we have shown that the quality of sentence alignment improves with the quality of the lexicon used.

We created three subsets of the FBIS Chinese/English bitext, consisting of 100, 300, 500 document pairs. Over each collection, and over the full FBIS bitext, we performed the iterative unsupervised sentence alignment procedure of Section 5.2. We then used each collection of aligned bitext in performing 8 EM iterations to produce an IBM Model 1 Chinese-to-English lexicon.

We measure the precision of these lexicons against the LDC Chinese-to-English dictionary[3]. In doing so, we apply a pruning threshold to the translation probability: if the probability of a translation is below the threshold, it is discarded. In Figure 5.2, we plot the precision of induced translation lexicon against its size as the pruning threshold varies. The results are consistent with intuition about how these procedures should behave. Overall precision increases with the size of the bitext used

[3]LDC Catalog Number LDC2002L27

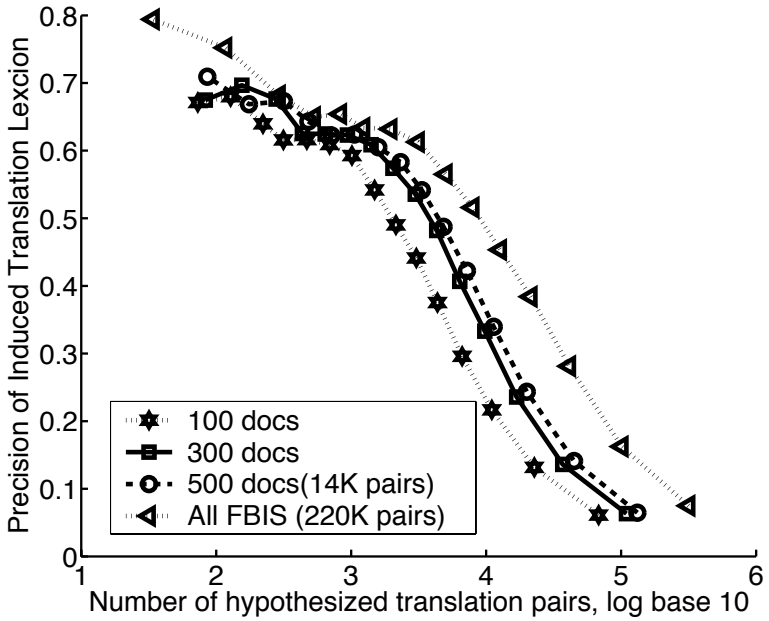


Figure 5.2: Precision of induced IBM Model 1 lexicons measured against the LDC Chinese-to-English bilingual dictionary. Each curve is associated with a single alignment of the bitext. DP+DC algorithm is applied to 100, 300, 500 and all document pairs from FBIS Chinese/English parallel corpus. From each set of alignments eight iterations of EM are used to induce an IBM Model 1 lexicon. Each curve is obtained by pruning the lexicon by a sequence of thresholds on the translation probability. Each point on each curve represents a pruned lexicon. The precision of each of these is plotted versus its number of entries.

in training; as the bitext size increases, more translations are generated at a fixed posterior pruning threshold; and overall precision tracks the posterior level fairly closely. While we observe that it is possible to generate a small, accurate lexicon with 500 document pairs, these experiments also show the limitations of the overall approach: if a translation precision of 0.7 is desired, training with the entire FBIS collection itself still yields fewer than 1000 entries.

5.7 Summary

The hybrid statistical chunk alignment modeling approach was developed with the goal of aligning large amounts of bitext for Statistical Machine Translation parameter estimation. A series of experiments have been conducted to evaluate the model and alignment approach in the context of unsupervised sentence alignment, bitext word alignment, and translation evaluation.

We find the approach to be robust in these applications, and when assessed in terms of sentence alignment on a manually annotated test set, we find balanced performance in precision and recall. An important feature of the approach is the ability to segment at the sub-sentence level as part of the alignment process. We find that this does not degrade translation performance of the resulting systems, even though the sentence segmentation is done with a weak translation model. The practical benefits of this are faster training of MT systems and the ability to retain more of the available bitext in MT training.

Beyond the practical benefits of better text processing for SMT parameter estimation, we observed interesting interactions between the word-level and sentence-level alignment procedures we studied. Although the models used in coarse, sentence-level alignment are relatively simple models of translation, they can still guide the alignment of long stretches of text by more powerful translation models based on complicated models of word movement. This suggests that sentence alignment and word alignment in bitext are not entirely independent modeling problems, and this work is intended to provide a framework, and the motivation, within which the joint modeling of both problems can be studied.

Chapter 6

Experimental Results of Word and Phrase Alignment

In this Chapter, we report experiments designed to evaluate performance of statistical word and phrase alignment models in word alignment and translation tasks. We begin with a description of training bitext for both Chinese-English and Arabic-English systems.

6.1 Data

6.1.1 Chinese-English

The training material of word and phrase alignment models are obtained by bitext chunking procedure as statistical chunk alignment models described in Chapter 2. We start from parallel document pairs released by LDC (statistics are shown in Table 5.1). For each corpus, we perform the hierarchial chunking process with "DP+DC" procedure as described in section 5.3.1 (System 4) to derive chunk pairs at sub-sentence level. Table 6.1 shows the number of chunk pairs and distribution of Chinese/English tokens for each corpus. These chunk pairs will serve as training bitext for statistical word/phrase alignment models.

The test data are NIST Chinese-English MT evaluation sets as described in section

Table 6.1: Statistics of Chinese-English Parallel (chunk pairs) Corpora

Corpus	# of Chunk Pairs	# of Chinese/English tokens (in Millions)
FBIS	368,191	7.83/10.45
HongKong Hansards	1,426,848	30.75/35.31
HongKong News	615,874	15.22/16.35
XinHua News	137,064	3.71/3.91
Sinorama	138,434	3.25/3.68
Chinese Treebank	4,726	0.097/0.13
UN (1993∼2002)	4,936,551	114.84/137.52
Total	7,627,688	175.70/207.36

Table 6.2: Statistics of Arabic-English Parallel (chunk pairs) Corpora

Corpus	# of Chunk Pairs	# of Arabic/English tokens (in Millions)
News	136,834	2.90/3.50
UN (1993∼2002)	4,982,431	120.06/129.04
Total	5,119,265	122.97/132.55

5.1.1.

6.1.2 Arabic-English

To prepare training data for statistical word alignment models for Arabic-English systems, we conduct similar bitext chunking procedure as in Chinese-English systems for news corpora starting with document pairs. The LDC-release of Arabic-English UN corpora have been sentence aligned. We use short sentence pairs (no more than 60 tokens on each side) to estimate IBM Model-1 translation lexicons and then perform divisive clustering procedure ("DP+DC(II)" in section 5.4) on long sentence pairs. The number of the final resulting chunk pairs (at sub-sentence level) and the distribution of Arabic/English tokens are presented in Table 6.2.

Similar to Chinese-English evaluations, we use the NIST Arabic-English MT evaluation sets (described in section 5.1.2) as the test data.

6.2 Chinese-English Bitext Word Alignment

We now investigate the Chinese-English bitext word alignment performance of the statistical word alignment models described in Chapter 3. We compare the word alignment quality of the HMM-based word-to-phrase alignment model to that of IBM Model-4.

6.2.1 Alignment Evaluation

The alignment test set consists of 124 sentences from the NIST 2001 dry-run MT-eval set [63] that are manually word aligned. We analyze the distribution of word links within these manual alignments. Of the Chinese words which are aligned to more than one English word, 82% align with consecutive English words (phrases). In the other direction, among all English words which are aligned to multiple Chinese words, 88% align to Chinese phrases. In this collection, at least, word-to-phrase alignments are plentiful.

Alignment performance is measured by the Alignment Error Rate (AER) [70]

$$AER(B;B') = 1 - 2 \times |B \cap B'|/(|B'| + |B|)$$

where B is a set of reference word links and B' are the word links generated automatically. AER is defined as the complement of F-measure with precision and recall weighted equally.

AER gives a general measure of word alignment quality. We are also interested in how the model performs over the word-to-word and word-to-phrase alignments it supports. We split the reference alignments into two subsets: B_{1-1} contains word-to-word reference links (e.g. 1→1 in Fig 3.4); and B_{1-N} contains word-to-phrase reference links (e.g. 1→3, 1→4 in Fig 3.4). The automatic word alignments B' are partitioned similarly. We define additional AERs that measure word-to-word and word-to-phrase alignment separately: $AER_{1-1} = AER(B_{1-1}, B'_{1-1})$ and $AER_{1-N} = AER(B_{1-N}, B'_{1-N})$.

Table 6.3: FBIS Bitext Alignment Error Rate.

	Models	AER_{1-1}	AER_{1-N}	AER
C⟶E	Model-4	37.9	68.3	37.3
	WtoW HMM	42.8	72.9	42.0
	WtoP HMM, N=2	38.3	71.2	38.1
	WtoP HMM, N=3	37.4	69.5	37.8
	WtoP HMM, N=4	37.1	69.1	37.8
	+ bigram t-table	37.5	65.8	37.1
E⟶C	Model-4	42.3	87.2	45.0
	WtoW HMM	45.0	90.6	47.2
	WtoP HMM, N=2	42.7	87.5	44.5
	+ bigram t-table	44.2	85.5	45.1

6.2.2 Initial Experiments on FBIS Corpus

We present word alignment experiments on the FBIS Chinese/Englsih parallel corpus that consists of 11,537 parallel documents with approximately 10M English/7.5M Chinese words.

Table 6.3 presents the three AER measurements for the WtoP alignment models trained as described in Section 3.3.3. GIZA++ Model 4 alignment performance is also presented for comparison. We note first that the Word-to-Word (WtoW) HMM alignment model is worse than Model 4, as expected. For the WtoP models in the C→E direction, we see reduced AER for phrases lengths up to 4, although in the E→C direction, AER is reduced only for phrases of length 2; performance for $N > 2$ is not reported. We also note WtoP HMM produces comparable word alignments to Model-4 in terms of AER.

From Table 6.3, we observe that in introducing the bigram phrase translation (the bigram t-table) there is a tradeoff between word-to-word and word-to-phrase alignment quality. As has been mentioned, the bigram t-table increases the likelihood of word-to-phrase alignments. In both translation directions, this reduces the AER_{1-N}. However, it also causes increases in AER_{1-1}, primarily due to a drop in recall: fewer word-to-word alignments are produced. For C→E, this is not severe enough to cause an overall AER increase; however, in E→C, AER does increase.

Fig. 6.1 (C→E, N=4) shows how Word-to-Word (1-1) and Word-to-Phrase (1-N) alignment behavior are balanced by the phrase count parameter. As η increases, the model favors alignments with more word-to-word links and fewer word-to-phrase links; the overall Alignment Error Rate (AER) suggests a good balance at $\eta = 8.0$.

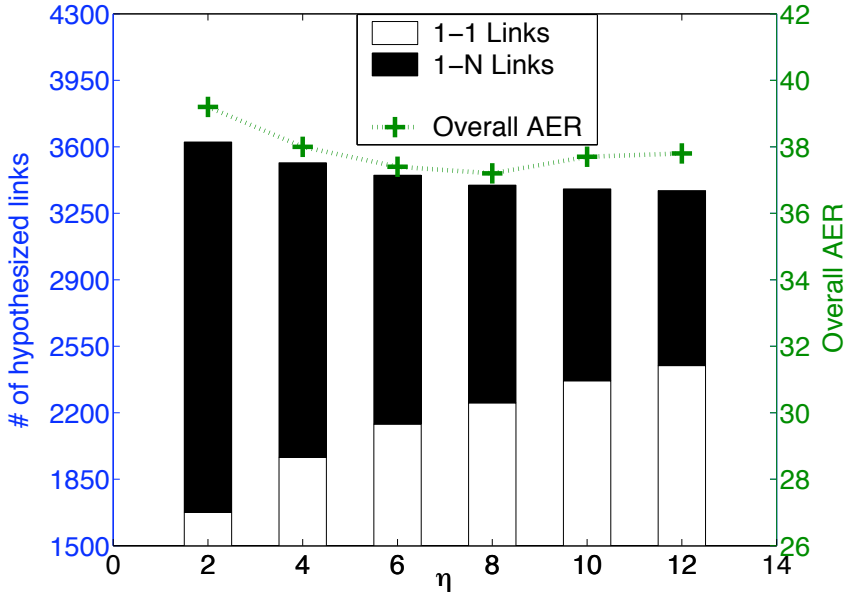


Figure 6.1: Balancing Word and Phrase Alignments.

6.2.3 Aligning Large Bitexts

In the previous section, we observed that the WtoP model performs comparably with Model-4 over the FBIS C-E bitext. In this section, we extend the alignment model to large bitexts. We investigate performance over these large bitexts :

- "NEWS" containing non-UN parallel Chinese/English corpora from LDC (mainly FBIS, Xinhua, HongKong News, HongKong Hansards, Sinorama, and Chinese Treebank).

- "NEWS+UN01-02" also including UN parallel corpora from the years 2001 and

Table 6.4: AER Over Large C-E Bitexts.

Bitext	English Words	Model	$AER_{C\rightarrow E}$	$AER_{E\rightarrow C}$
NEWS	71M	M-4	37.1	45.3
		WtoP	36.1	44.8
NEWS+ UN01-02	96M	M-4	36.1	43.4
		WtoP	36.4	44.2
ALL C-E	200M	WtoP	36.8	44.7

2002.

- "ALL C-E" refers to all the C-E bitext available from LDC as of his submission; this consists of the NEWS corpora with the UN bitext from all years (1993-2002).

Over all these collections, WtoP alignment performance (Table 6.4) is comparable to that of Model-4. We do note a small degradation in the E→C WtoP alignments. It is quite possible that this one-to-many model suffers slightly with English as the source and Chinese as the target, since English sentences tend to be longer. Notably, simply increasing the amount of bitext used in training does not necessarily improve AER. However, larger aligned bitexts can give improved phrase pair coverage of the test set.

One desirable feature of HMMs is that the Forward-Backward steps can be run in parallel: bitext is partitioned; the Forward-Backward algorithm is run over the subsets on different CPUs; statistics are merged to re-estimate model parameters. Partitioning the bitext also reduces memory usage, since different co-occurrence tables can be kept for each partition. With the "ALL C-E" bitext collection, a single set of WtoP models (C→E, N=4, bigram t-table) can be trained over 200M words of Chinese-English bitext by splitting training over 40 CPUs; each Forward-Backward process takes less than 2GB of memory; and the training run finishes in five days.

By contrast, the 96M English word NEWS+UN01-02 is about the largest C-E bitext over which we can train Model-4 with our GIZA++ configuration and computing infrastructure.

Based on these and other experiments, we set a maximum value of $N = 4$ for F→E; in E→F, we set N=2 and omit the bigram phrase translation probability; η is set to 8.0.

6.3 Translation Evaluation

We evaluate the quality of phrase pairs extracted from the bitext through translation performance, which is measured by BLEU [72]. The BLEU metric is defined as the geometric mean of n-gram precisions weighted by sentence length penalty. In our setup, there are four references for each test sentence, and we measure up to 4-grams. We also report the coverage of a PPI over a test set, which is the percentage of foreign phrases up to length 5 that have English translations in the PPI. We present Chinese-English translation and Arabic-English translation results and compare with Model-4.

6.3.1 Chinese-English Translations

We report performance on the NIST Chinese/English 2002, 2003 and 2004 (News only) MT evaluation sets consisting of 878, 919, and 901 sentences, respectively.

We evaluate two C→E translation systems. The smaller system is built on the FBIS C-E bitext collection. The language model used for this system is a trigram word language model estimated with 21M words taken from the English side of the bitext; all language models in this article are built with the SRILM toolkit using Kneser-Ney smoothing [77].

The larger system is based on alignments generated over all available C-E bitext (the "ALL C-E" collection of Section 6.2.3). The language model is an equal-weight interpolated trigram model trained over 373M English words taken from the English side of the bitext and the LDC English Gigaword corpus.

6.3.2 Arabic-English Translations

We also evaluate our WtoP alignment models in Arabic-English translation, reporting results on small and large systems. We test our models on NIST Arabic/English 2002, 2003 and 2004 (News only) MT evaluation sets that consist of 1043, 663 and 707 Arabic sentences, respectively.

In the small system, the training bitext is from A-E News parallel text, with ∼3.5M

Table 6.5: Chinese→English Translation Analysis and Performance of Viterbi PPI Extraction (V-PE) and WtoP Posterior Induction Procedures

		V-PE	WtoP	eval02		eval03		eval04	
		Model	T_p	cvg	BLEU	cvg	BLEU	cvg	BLEU
FBIS System	1	M-4	-	20.1	23.8	17.7	22.8	20.2	23.0
	2		0.7	24.6	24.6	21.4	23.7	24.6	23.7
	3	WtoP	-	19.7	23.9	17.4	23.3	19.8	23.3
	4		1.0	23.1	24.0	20.0	23.7	23.2	23.5
	5		0.9	24.0	24.8	20.9	23.9	24.0	23.8
	6		0.7	24.6	24.9	21.3	24.0	24.7	23.9
	7		0.5	24.9	24.9	21.6	24.1	24.8	23.9
Large System	8	M-4	-	32.5	27.7	29.3	27.1	32.5	26.6
	9	WtoP	-	30.6	27.9	27.5	27.0	30.6	26.4
	10		0.7	38.2	28.2	32.3	27.3	37.1	26.8

words on the English side. We follow the same training procedure and configurations as in Chinese/English system in both translation directions. The language model is an equal-weight interpolated trigram built over ∼400M words from the English side of the bitext, including UN text, and the LDC English Gigaword collection. The large Arabic/English system employs the same language model. Alignments are generated over all A-E bitext available from LDC as of this submission; this consists of approx. 130M words on the English side.

6.3.3 WtoP Model and Model-4 Comparison

We first look at translation performance of the small A→E and C→E systems, where alignment models are trained over the smaller bitext collections. The baseline systems (Table 6.5 and 6.6, line 1) are based on Model-4 Viterbi Phrase-Extract PPIs.

We compare WtoP alignments directly to Model-4 alignments by extracting PPIs from the WtoP word alignments using the Viterbi Phrase-Extract procedure. In C→E translation (Table 6.5, line 3), performance is comparable to that of Model-4; in A→E translation (Table 6.6, line 3), performance lags slightly. As we add phrase pairs to the WtoP Viterbi Phrase-Extract PPI via the Phrase-Posterior Augmentation

Table 6.6: Arabic→English Translation Analysis and Performance of Viterbi PPI Extraction (V-PE) and WtoP Posterior Induction Procedures

		V-PE	WtoP	eval02		eval03		eval04	
		Model	T_p	cvg	BLEU	cvg	BLEU	cvg	BLEU
News System	1	M-4	-	19.5	36.9	21.5	39.1	18.5	40.0
	2		0.7	23.8	37.6	26.6	40.2	22.4	40.3
	3	WtoP	-	18.4	36.2	20.6	38.6	17.4	39.2
	4		1.0	21.8	36.7	24.3	39.3	20.4	39.7
	5		0.9	23.2	37.2	25.8	39.7	21.8	40.1
	6		0.7	23.7	37.2	26.5	39.7	22.4	39.9
	7		0.5	24.0	37.2	26.9	39.7	22.7	39.8
Large System	8	M-4	-	26.4	38.1	28.1	40.1	28.2	39.9
	9	WtoP	-	24.8	38.1	26.6	40.1	26.7	40.6
	10		0.7	30.7	39.3	32.9	41.6	32.5	41.9

procedure (Table 6.5, Table 6.6, lines 4-7), we obtain a ∼1% improvement in BLEU; the value of $T_p = 0.7$ gives improvements across all sets. In C→E translation, this yields good gains relative to Model-4, while in A→E we match or improve the Model-4 performance.

The performance gains through PPI augmentation are consistent with increased PPI coverage of the test set. We tabulate the percentage of test set phrases that appear in each of the PPIs (the 'cvg' values in Table 6.5, Table 6.6). The augmentation scheme is designed specifically to increase coverage, and we find that BLEU score improvements track the phrase coverage of the test set. This is further confirmed by the experiment of Table 6.5 and Table 6.6, line 2 in which we take the PPI extracted from Model-4 Viterbi alignments, and add phrase pairs to them using the Phrase-Posterior augmentation scheme with $T_p = 0.7$ and WtoP alignment models. We find that the augmentation scheme under the WtoP models can be used to improve the Model-4 PPI itself.

We also investigate C→E and A→E translation performance with PPIs extracted from large bitexts. Performance of systems based on Model-4 Viterbi Phrase-Extract PPIs is shown in Table 6.5 and Table 6.6, line 8.

To train Model-4 using GIZA++, we split the bitexts into two (A-E) or three

(C-E) partitions, train models for each division separately, and find word alignments for each division separately with their models; otherwise, we find that memory usage is too great. These serve as a single set of alignments for the bitext, as if they had been generated under a single alignment model.

When we translate with Viterbi Phrase-Extract PPIs taken from WtoP alignments created over all available bitext, we find comparable performance to the Model-4 baseline (Table 6.5, Table 6.6, line 9). Using the Phrase-Posterior augmentation scheme with $T_p = 0.7$ yields further improvement (Table 6.5, Table 6.6, line 10).

Table 6.7: Translation results on the merged test sets

Model	PPI	$BLEU_{C-E}$	$BLEU_{A-E}$
Model-4	baseline	$27.29^{\pm 0.5}$	$39.39^{\pm 0.6}$
WtoP HMM	augmented	$27.47^{\pm 0.5}$	$40.48^{\pm 0.6}$

We also perform tests to see if the improvements under the BLEU metric are statistically significant [66]. Pooling all three test sets of eval02, eval03, and eval04, we form large test sets for C→E and A→E translations. We compare the translation performance of two setups: one is the Model-4 word alignments with the baseline PPI (Viterbi Phrase-Extract) (as in Table 6.5 and 6.6, line 8), while the other is the Word-to-Phrase word alignments with the augmented PPI (as in Table 6.5 and 6.6, line 10). We show their BLEU scores as well as their 95% confidence intervals in Table 6.7. We find that the WtoP alignment model leads to equivalent C→E system performance as that of Model-4, while A→E system improvements are significant at a 95% level [66].

6.4 Effect of Language Model

We now investigate the effects of language models on statistical machine translation systems. Specifically, we test and compare performance on the large Arabic-English translation system described in section 6.3.2. We train three different n-gram language models using modified Kneser-Ney smoothing as implemented in the SRILM [77] toolkit, studying how each influences translation results.

The English sources used for each language model are tabulated in the Table 6.8, with the number of English words in million shown for each source. The training text for the small 3-gram LM consists of data from XinHua and AFP of the English Gigawords released by LDC and the English side of News bitext. For each of the three sources, a 3-gram language model is trained separately, and the linear interpolation of them with weights 0.4, 0.4, and 0.2, respectively, gives the small 3-gram LM.

The big 3-gram LM has additional training data from English side of the UN bitext. Like the small 3-gram LM, a 3-gram language model is trained for each of the four sources and the big 3-gram LM is the linear interpolation of them with the equal weight of 0.25. The big 4-gram LM is trained similarly to the big 3-gram LM but with 4-gram language models.

Table 6.8: The number of English text (in millions) used to train language models.

Source	XinHua	AFP	UN	News	Total
small 3-gram	63.1	200.8	-	2.1	266.0
big 3-gram	83.0	210.0	131.0	3.6	428.0
big 4-gram	83.0	210.0	131.0	3.6	428.0

We run the TTM translation system with the three language models and report results on NIST MT evaluation 2002 and 2003 test set. The performance is measured by BLEU metric [72]. As the Table 6.9 shows, the big 3-gram language model gives more than 2 BLEU points improvement than the small 3-gram language model. This is achieved by having more training data in building a language model. The big 4-gram language model is applied to re-rank the 1000-best hypothesis generated with the big 3-gram language model. We observe additional 1 BLEU point gain out of

these.

Table 6.9: Language model effects on the large Arabic-English translation system measured by BLEU score.

Language Model	eval02	eval03
small 3-gram	35.81	37.82
big 3-gram	38.14	40.08
big 4-gram	39.14	41.38

6.5 Summary

We have shown that word-to-phrase alignment models are capable of producing good quality bitext word alignment. In Chinese-English word alignment tasks, they compare well to Model-4, even with large bitexts. Efficient DP-based training algorithms and parallel implementation enables the building of a single model with all Chinese-English bitext released by LDC.

In Arabic-English and Chinese-English translation, word-to-phrase alignment models compare well to Model-4, even with large bitexts. The model architecture was inspired by features of Model-4, such as fertility and distortion, but care was taken to ensure that dynamic programming procedures, such as EM and Viterbi alignment, could still be performed. There is practical value in this: training and alignment are easily parallelized. A single model can be built with all training bitext. With increasingly availability of training data, this practical advantage is becoming even more desirable.

Working with HMMs also makes it straightforward to explore new modeling approaches. We show an augmentation scheme that adds to phrases extracted from Viterbi alignments; this improves translation with both the WtoP and the Model-4 phrase pairs, even though it would not be feasible to implement the scheme under Model-4 itself. We note that these models are still relatively simple, and we anticipate further alignment and translation improvement as the models are refined.

Part II

Language Modeling

Chapter 7

Latent Semantic Analysis

7.1 Introduction

In section 6.4, we found that language model is a very important component in selecting the best English word sequence among all possible hypotheses in the output during translation. In this section, we study statistical language modeling techniques. We begin with an introduction of Latent Semantic Analysis (LSA), which has been shown to be effective in improving language model performance to capture multi-span dependency [2] [15] [19].

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text [43]. It provides a method by which to calculate the similarity of meaning of given words and documents. Measures of the relationships between words and documents produced by LSA have been found to correlate well with several human cognitive phenomena involving association or semantic similarity. As a practical method of characterizing word meaning, LSA has been successfully applied to information retrieval [16], statistical langauge modeling [2] [19], and automatic essay assessment [61].

LSA induces representations of the meaning of words and documents from text alone using statistical analysis techniques. No linguistic knowledge, semantic net-

[1]The work in this chapter was done under Prof. S. Khudanpur's supervision.

work, or other resources are required. The extraction of meaning is performed in an unsupervised manner from a collection of texts wherein document boundaries are marked. Word co-occurrence counts or correlations in usage within documents are analyzed. Much deeper relations between words (thus the phrase "Latent Semantic") are inferred by means of a powerful factor analysis technique, Singular Value Decomposition (SVD). SVD conducts matrix factorization to identify patterns in data and expresses these findings in such a way as to highlight their similarities and differences. It can be used to identify important and informative features and reduce the dimension of data representation.

LSA is a bag-of-word model. Word order within a document is ignored. The underlying assumption is that words that appear in the same document are semantically consistent. LSA is meant to address two fundamental language phenomena: synonymy and polysemy. In synonymy, the same idea can be described in different ways while in polysemy the same word can have multiple meanings.

LSA can be regarded as a preprocessing step used before document classification or other tasks. The starting point of LSA is the construction of a matrix describing word-document co-occurrence. By performing singular value decomposition of this matrix, a short vector representation is derived for each word and document. One advantage of the resulting word and document representation is that they all live in the same low-dimensional continuous vector space, enabling one to quantitatively measure closeness or similarity between words and documents. The cosine of the angle between two vectors is a standard measure of similarity in this framework. The similarity between words and documents can be measured and utilized by applications, such as information retrieval and language modeling.

In this chapter, we describe each stage of the LSA process, starting from matrix construction, to factorization by SVD, to final low-dimensional representation of words and documents, and similarity measurements for them.

7.2 Word-Document Frequency Matrix W

LSA requires a corpus separated into semantically coherent documents as well as a vocabulary to cover words found in these documents. It is assumed that the co-occurrence of any two words within a document at a rate much greater than chance is an indication of their semantic similarity. The notation and exposition in this chapter closely follows that of Bellegarda [2].

Before statistics collecting, documents are usually processed into space delimited sequences. For instance, in information retrieval, words in a document are lowercased and stemmed. The basic indexing unit is called a "term". While in language modeling applications, documents are tokenized, we define a document, without ambiguity, as a collection of words separated by spaces.

The first step in LSA is to represent co-occurrence information by a large sparse matrix. Let $\mathbf{V}$, $|\mathbf{V}| = M$ be the underlying task vocabulary and $\mathbf{T}$ a text corpus, with document boundaries marked, comprised of N documents relevant to some domain of interest. Typically, M and N are of the order of 10^4 and 10^5, respectively. $\mathbf{T}$, the training corpus, may thus have hundreds of millions of words. The construction of the $M \times N$ matrix W of co-occurrences between words and documents ignores word order within the document; it is accumulated from $\mathbf{T}$ by simply counting how many times a word appears in a document.

There are several choices for the matrix entry $[W]_{ij}$. In information retrieval, when indexing terms and documents, the commonly used entry is the Term Frequency weighted by the Inverse Document Frequency, called *TF-IDF*. Let c_{ij} be the raw count of a word $w_i \in \mathbf{V}$ in a document $d_j \in \mathbf{T}$, which is the term frequency (TF) that indicates the local importance of the word w_i to the document. Let N_i be the number of documents which contain the word w_i; then the IDF is defined as $\log \frac{N}{N_i}$, which measures the importance of the word globally. A function word, e.g., "a" or "the", which is very likely to appear in any documents, will have a lower IDF value, while a content word, say "Markov", which would be found only in documents relating to a certain topic, therefore has a higher IDF value.

Alternative implementations of matrix entry that have been studied in information

retrieval can be found in [22]. Empirical evidence has shown that a realization which is normalized by document length and word global weight leads to good performance. To be specific, in constructing the word-document co-occurrence matrix W, the row count c_{ij} is weighted by:

- the "relevance" of a word in the vocabulary to the topic of a document, function words being given less weight than content words, and
- the size of the document, a word with a given count in a longer document being given less weight than in a shorter one.

To accomplish the former, assume that a single (unknown) document in our collection **T** is relevant for some task, and our goal is to guess which document it is. Let the *a prior* probability of a document being relevant be uniform ($\frac{1}{N}$) on the collection and, further, let an orale draw a single word at random from the relevant document and reveal it to us. The conditional probability of d_j being the relevant document, given that the relevant document contains the word w_i, is clearly $\frac{c_{ij}}{c_i}$, where $c_i = \sum_{j=1}^{N} c_{ij}$. The ratio of the average *conditional entropy* of the relevant document's identity, given w_i and its *a prior entropy* is thus a measure of the (un)informativeness of w_i. Highly informative words w_i have small values of

$$\epsilon_i = \epsilon_{w_i} = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{ij}}{c_i} \log \frac{c_{ij}}{c_i} \,. \tag{7.1}$$

Since $0 \leq \epsilon_i \leq 1$, the raw counts in the i-th row of W are weighted by $(1 - \epsilon_i)$.

To achieve the latter effect, the counts in the j-th column of W are weighted by the total length $c_j = \sum_{i=1}^{M} c_{ij}$ of the document d_j. In summary,

$$[W]_{ij} = (1 - \epsilon_i) \frac{c_{ij}}{c_j} \,, \tag{7.2}$$

is the resulting ij-th matrix entry.

7.3 Singular Value Decomposition of W

Each column of the matrix W represents a document and each row represents a word. Typically, W is very sparse. To obtain a compact representation, singular

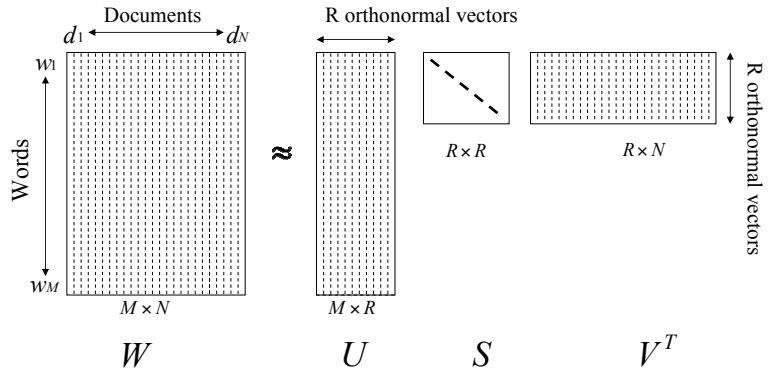


Figure 7.1: Singular Value Decomposition of the Sparse Matrix W.

value decomposition (SVD) is employed (cf. Berry et al [4]) to yield

$$W \approx \hat{W} = U \times S \times V^T, \tag{7.3}$$

as Figure 7.1 shows, where, for some order $R \ll \min(M, N)$ of the decomposition, U is a $M \times R$ left singular matrix with rows u_i, $i = 1, \cdots, M$, S is a $R \times R$ diagonal matrix of singular values $s_1 \geq s_2 \geq \ldots \geq s_R \gg 0$, and V is $N \times R$ a right singular matrix with rows v_j, $j = 1, \cdots, N$. Note that both U and V are column-orthonormal, i.e., $U^T U = V^T V = I_R$ (the identity matrix of order-R). For each i, the scaled R-vector $u_i S$ may be viewed as representing w_i, the i-th word in the vocabulary, and similarly the scaled R-vector $v_j S$ as representing d_j, j-th document in the corpus. Note that the $u_i S$'s and $v_j S$'s both belong to $\mathbb{R}^R$, the so-called LSA-space.

The approximation of W with $\hat{W}$ is optimal in the sense that the matrix $\hat{W}$ is the best rank-R approximation to the word-document matrix W, for any unitarily invariant norm. This implies that for any matrix A of rank R

$$\min_{A:\text{rank}(A)=R} ||W - A|| = ||W - \hat{W}|| = s_{R+1} \tag{7.4}$$

where $||\cdot||$ refers to the L_2 norm and s_{R+1} is the $R+1$-th smallest singular value of W. Clearly, the rank of W is R implies no approximation and that s_{R+1} is zero.

7.4 Similarity Measurements

SVD projects each word in the vocabulary into the LSA-space. Before the SVD, the word w_i in the vocabulary is represented by a N dimensional row vector (the i-th row of W). After the matrix decomposition, it is represented by a R dimensional row vector ($u_i S$). The process of SVD generates a compact and meaningful representation of each word. The similarity between words can be captured by their vector cosine distance in the low-dimensional space, sometimes referred to as the LSA-space. The following similarity measure between the i-th and i'-th words w_i and $w_{i'}$ is frequently used:

$$K(w_i, w_{i'}) = \cos(u_i S, u_{i'} S) = \frac{u_i S^2 u_{i'}^T}{\|u_i S\| \times \|u_{i'} S\|} . \tag{7.5}$$

It can be understood by examining the (i, i') entry of the matrix WW^T. WW^T characterizes all co-occurrences between words. The (i, i') cell of WW^T implies the extent to which words w_i and $w_{i'}$ are likely to appear in the same document across the entire training set. As the approximation of WW^T, $\hat{W}\hat{W}^T$ provides a better derivation. Expanding $\hat{W}$ with Equ. (7.3) and considering S to be diagonal and V to be column-orthonormal, $\hat{W}\hat{W}^T = US^2U^T$, it becomes obvious that the (i, i') cell is the dot product of $u_i S$ and $u_{i'} S$. The similarity measurement in Equ. (7.5), then, is the normalized dot product, which captures the cosine of the angles between the two word vectors in the LSA-space.

Words can be grouped into clusters using algorithms such as K-means with the definition of Equ. (7.5) as a measure of similarity between words. Word clustering is necessary and beneficial for class-based language models [8]. Semantically motivated word clustering is very useful for query expansion in information retrieval to improve recall [1].

Replacing u_i's with v_j's in the definition above, a corresponding measure $K(d_j, d_{j'})$

$$K(d_j, d_{j'}) = \cos(v_j S, v_{j'} S) = \frac{v_j S^2 v_{j'}^T}{\|v_j S\| \times \|v_{j'} S\|} . \tag{7.6}$$

of similarity between the j-th and j'-th documents is obtained and has been used for document clustering, filtering and topic detection. Similar to words, the document

similarity defined in Equ. (7.6) can be understood by examining (j, j') entry of the matrix $\hat{W}^T\hat{W}$.

Since words and documents are projected into the same low-dimensional space, the LSA-space, the distance between a word and a document can also be measured. Inspecting the (i, j) cell of the matrix $\hat{W} = US^{\frac{1}{2}}(VS^{\frac{1}{2}})^T$ reveals how closely the word w_i and the document d_j stay together in the LSA-space. Their similarity can also be measured by the cosine of the angle between their vectors.

$$K(w_i, d_j) = \cos(u_i S^{\frac{1}{2}}, v_j S^{\frac{1}{2}}) = \frac{u_i S v_j^T}{\|u_i S^{\frac{1}{2}}\| \times \|v_j S^{\frac{1}{2}}\|}, \tag{7.7}$$

7.5 Representing Pseudo-Documents

LSA is accomplished on a fixed training corpus. It represents words and documents in the training corpus by assigning them to points in the low-dimensional LSA-space. It is very important to compute appropriate comparison quantities for objects which are not in the original analysis. In information retrieval, LSA is usually performed on a collection of documents. During retrieval, a user supplies a query which is compared against documents in the database. Documents are then ranked by their similarities to the input query. To measure the closeness between the query and each document, it is necessary to represent the query in the LSA-space. The query is regarded as a pseudo-document.

A pseudo-document is a collection of words which is not in the training corpus of the LSA process. Let d_q be the raw $M \times 1$ vector representing a pseudo-document and v_q be a row vector, the pseudo-document's low-dimensional representation in the LSA-space. From the definition of entry in the spare matrix W, d_q can be computed. To derive v_q from d_q, imagine adding the pseudo-document into the training corpus as the $N + 1$-th document. This implies appending W with d_q and appending V by v_q in Equ. (7.3) of SVD:

$$[W\ d_q] \approx [\hat{W}\ \hat{d}_q] = U \times S \times [V^T\ v_q^T] \tag{7.8}$$

It is assumed the approximation in Equ. (7.8) is still optimal in the sense of the

best-R approximation as in Equ. (7.4). The assumption leads to approaching d_q by $\hat{d}_q$: $d_q = USv_q^T$; since U is column-orthonormal, a little algebra shows that

$$v_q = d_q^T US^{-1} \tag{7.9}$$

Note that this representation is just like a row of V. With appropriate scaling by $S^{\frac{1}{2}}$ or S, it can be used like a usual document's representation in the LSA-space for making comparisons against words or documents, respectively.

7.6 Word Clustering Examples

For the purpose of illustration, we report word clustering experiments on the Wall Street Journal corpus. The training data consists of 86,602 articles from the years 1987 to 1989 totaling 40.5M English words. The most frequent 23K words are chosen as the vocabulary.

We construct the word-document matrix with TF-IDF entry. After SVD [4], low dimension ($R = 153$) representation for each word in the vocabulary is obtained. We use the K-means algorithm to cluster words. The initial cluster number is 600, and the final cluster number is 588. Two representative clusters are illustrated in the following table.

Cluster I

---

acura, aries, audi, beretta, bonneville, brakes, braking, buick, cadillac, car, car's, aravan, cars, cavalier, chevrolet, chevy, compact, conditioning, corsica, coupe, cutlass, daihatsu, ealership, dealerships, dodge, eagle, fiero, geo, hahn, honda, honda's, horsepower, yundai, incentives, infiniti, isuzu, jeeps, lebaron, lexus, luxury, maserati, mazda, mazda's, mercedes, midsized, minivan, minivans, model, models, nissan, nissan's, oldsmobile, optional, pickup, pickups, pontiac, pony, porsche, prelude, prix, proton, rebate, rebates, regal, reliant, riviera, sedan, showroom, showrooms, sporty, styling, subaru, subcompact, subcompacts, suzuki, toyota, toyota's, truck, trucks, turbo, vehicles, volkswagen's, volvo, wagon, wagons, warranties, warranty, wheel, windshield, yugo

Cluster II
Alabama, Arkansas, Carolina, Clinton, Columbus, Dakota, Georgetown, Georgia, Illinois, Illinois's, Kentucky, Langley, Maryland, Michigan, Michigan's, Midwestern, Minnesota, Mississippi, Missouri, Nationally, Nebraska, Ohio, Oregon, Pennsylvania, Rhode, Springfield, Tennessee, Vermont, Virginia, Wisconsin

Clearly, cluster I collects words related to "vehicles", while cluster II is about "U.S. states". Indeed, words within a group are semantically consistent.

7.7 Summary

We discussed Latent Semantic Analysis (LSA) as a technique for extracting meaningful and compact representations for words and documents via Singular Value Decomposition (SVD). One advantage of LSA inheres in how it projects words and documents into the same LSA-space. Therefore, similarities between words and documents can be measured by the cosine of angles between their vectors. As a preprocess step, LSA quantitatively finds semantic features of words and documents, and so it can be applied to information retrieval and classification tasks, to give some examples. In the next chapter, we will discuss its application to statistical langauge modeling by showing how language models can be improved by incorporating semantic features derived from LSA.

Chapter 8

Latent Semantic Analysis in Language Models

8.1 Introduction

In Chapter 7, we discussed Latent Semantic Analysis (LSA) techniques as a method to extract and represent the meanings of words and documents in a low dimensional vector space. We now investigate the application of LSA to statistical language modeling.

Statistical language modeling benefits greatly from the augmentation of standard N-gram statistics with information about the syntactic structure of the sentence and the semantic context of the segment or story being processed, as witnessed by the improved performance of automatic speech recognition systems that use such models. In highly constrained settings such as a telephone-based interactive voice-response system, sometimes called a dialogue system, it may be reasonable to limit the notion of syntax to finite state grammars, while the notion of semantics may be adequately captured by a dialogue-state variable representing the type of sentence that may be spoken next by a user. In less constrained speech recognition tasks, e.g. transcription of Broadcast News or conversational telephone speech, the incorporation of syntactic information is usually via a statistical left-to-right parser, while semantic information

[1]The work in this chapter was done under Prof. S. Khudanpur's supervision.

is usually brought in through some notion of topicality or "aboutness" of the sentence being processed. It is this latter notion of semantics in statistical language modeling that is the subject of this work.

Collocation or N-gram statistics have proven to be one of the best predictors of words in a sentence, and all attempts to augment a language model (LM) with semantic information also aim to conform to N-gram statistics in one form or another. The straightforward technique [31, 14] is to

1. group documents or stories from a putatively large LM training corpus into semantically cohesive clusters using an information retrieval based notion of document similarity,
2. estimate N-gram LMs for each cluster, and
3. interpolate the topic-specific N-gram model with an N-gram model estimated from the undivided LM training corpus.

Alternatives to this method fall into two broad categories, one based on latent semantic analysis (LSA), *e.g.*, Coccaro and Jurafsky [15] and Bellegarda [2], and another based on maximum entropy, *e.g.*, Chen and Rosenfeld [12] and Khudanpur and Wu [35]. In this work, we attempt to find a bridge between these two techniques.

For language modeling, a pseudo-document is constructed from (possibly all) the words preceding a particular position in an utterance and the resulting vector is projected into the above-mentioned low-dimensional vector space, sometimes referred to as the LSA space. Intuition suggests that words with vectors close to the pseudo-document vector are more likely to follow than those far away from it. This is used to construct a conditional probability on the task-vocabulary. This probability, which depends on a long span of "history" is then suitably combined with an N-gram probability.

An alternative to first constructing a conditional probability on the task-vocabulary independently of the N-gram model and then seeking ways to combine the two probabilities, is directly modeling the pseudo-document as yet another conditioning event — on par with the preceding $N-1$ words — and finding a single probability distribution conditioned on the entire "history." Note that the co-occurrence of the

predicted word with, say, the immediately preceding word in the history is a discrete event and amenable to simple counting. By contrast, the pseudo-document is a continuous-valued vector, and simply counting how often a word follows a particular vector in a training corpus is meaningless. Consequently, we must employ a parametric model for word-history co-occurrence, possibly together with quantization of the pseudo-document vector.

The remainder of this chapter explores these main themes as follows. We briefly describe how to induce an LSA probability distribution from history in Section 8.2. We then describe the standard LSA language modeling techniques we implemented 8.3, presenting several realizations of combining N-grams with LSA probability. Finally, we describe the maximum entropy alternative for combining N-gram and latent semantic information in Section 8.4.

8.2 Calculating Word-Probabilities Using LSA

During LSA process, we construct the word-document co-occurrence matrix W with 'cell' defined as word frequency weighted by document size and entropy-related global weight as in Equ. (7.2). After SVD, words and training documents are projected into the low-dimensional LSA space.

Statistical language models specify a probability distribution of words to predict the following words based on current history. Histories are regarded as pseudo-documents and need to be projected into the LSA-space to be compared against vectors of words. Then, a conditional probability on the vocabulary, the so-called LSA probability, can be induced from similarities between the history and words in the vocabulary.

Given a sequence $w_1, w_2, \ldots, w_T$ of words in a document, the semantic coherence between w_t, the word in the t-th position, and $\tilde{d}_{t-1} \equiv \{w_1, \ldots, w_{t-1}\}$, its predecessors, is used to construct a conditional probability on the vocabulary.

Specifically, a $M \times 1$ pseudo-document vector $\tilde{d}_{t-1}$ is constructed by weighting the frequency of the preceding words in accordance with (7.2), and its scaled R-vector

representation $\tilde{v}_{t-1}S = \tilde{d}_{t-1}^T U$ is used in Equ. (7.6) to obtain

$$P_{\text{LSA}}(w_t|\tilde{d}_{t-1}) = \frac{\left[K(w_t, \tilde{d}_{t-1}) - K_{\min}(\tilde{d}_{t-1})\right]^\gamma}{\sum_w \left[K(w, \tilde{d}_{t-1}) - K_{\min}(\tilde{d}_{t-1})\right]^\gamma}, \tag{8.1}$$

where $K_{\min}(\tilde{d}) = \min_w K(w, \tilde{d})$ is an offset to make the probabilities non-negative, and $\gamma >> 1$ is chosen experimentally, as by Coccaro and Jurafsky [15], to increase the otherwise small dynamic range of K as w varies over the vocabulary.

As one processes successive words in a document, the pseudo-document $\tilde{d}_{t-1}$ is updated incrementally:

$$\tilde{d}_t = \frac{t-1}{t}\tilde{d}_{t-1} + \frac{1-\epsilon_{w_t}}{t}\mathbf{e}_{w_t}, \tag{8.2}$$

where $\mathbf{e}_{w_t}$ is a $M \times 1$ vector with a 1 in the position corresponding to w_t and 0 elsewhere. Consequently, the vector $\tilde{v}_{t-1}S$ needed for the similarity computation of (7.7) towards the probability calculation of (8.1) is also incrementally updated:

$$\tilde{v}_t S = \lambda \frac{t-1}{t}(\tilde{v}_{t-1}S) + \frac{1-\epsilon_{w_t}}{t}u_{w_t}, \tag{8.3}$$

where a positive "decay" coefficient $\lambda < 1$ is thrown in to accommodate dynamic shifts in topic.

8.3 Combining LSA probabilities with N-grams

Several strategies have been proposed [15] [2] for combining the LSA-based probability described above with standard N-gram probabilities, and we list those which we have investigated for conversational speech.

Linear Interpolation: For some experimentally determined constants α and $\overline{\alpha} = 1 - \alpha$,

$$P(w_t|w_{t-1}, w_{t-2}, \tilde{d}_{t-1}) = \alpha P_{\text{LSA}}(w_t|\tilde{d}_{t-1}) + \overline{\alpha} P_{N\text{-gram}}(w_t|w_{t-1}, w_{t-2}).$$

Similarity Modulated N-gram: With the similarity (7.7) offset to be nonnegative, as done in (8.1),

$$P(w_t|w_{t-1}, w_{t-2}, \tilde{d}_{t-1}) = \frac{K(w_t, \tilde{d}_{t-1}) P_{N\text{-gram}}(w_t|w_{t-1}, w_{t-2})}{\sum_w K(w, \tilde{d}_{t-1}) P_{N\text{-gram}}(w|w_{t-1}, w_{t-2})}.$$

Information Weighted Arithmetic Mean: Setting $\lambda_w = \frac{1-\epsilon_w}{2}$ and $\overline{\lambda}_w = 1 - \lambda_w$,

$$P(w_t|w_{t-1}, w_{t-2}, \tilde{d}_{t-1}) = \frac{\lambda_{w_t} P_{\text{LSA}}(w_t|\tilde{d}_{t-1}) + \overline{\lambda}_{w_t} P_{N\text{-gram}}(w_t|w_{t-1}, w_{t-2})}{\sum_w \lambda_w P_{\text{LSA}}(w|\tilde{d}_{t-1}) + \overline{\lambda}_w P_{N\text{-gram}}(w|w_{t-1}, w_{t-2})}.$$

Information Weighted Geometric Mean: With the same λ_w and $\overline{\lambda}_w$ as above,

$$P(w_t|w_{t-1}, w_{t-2}, \tilde{d}_{t-1}) = \frac{P_{\text{LSA}}^{\lambda_{w_t}}(w_t|\tilde{d}_{t-1}) \cdot P_{N\text{-gram}}^{\overline{\lambda}_{w_t}}(w_t|w_{t-1}, w_{t-2})}{\sum_w P_{\text{LSA}}^{\lambda_w}(w|\tilde{d}_{t-1}) \cdot P_{N\text{-gram}}^{\overline{\lambda}_w}(w|w_{t-1}, w_{t-2})}.$$

8.4 Exponential Models with LSA Features

The ad hoc construction of $P_{\text{LSA}}(w|\tilde{d}_{t-1})$ to somehow capture $K(w, \tilde{d}_{t-1})$, and its combination with N-gram statistics described above, is a somewhat unsatisfactory aspect of the LSA-based models. We propose an alternative family of exponential models

$$\begin{aligned} & P_{\underline{\alpha}}(w_t|\tilde{d}_{t-1}, w_{t-2}, w_{t-1}) \\ &= \frac{\alpha_{w_t}^{f_1(w_t)} \alpha_{w_{t-1},w_t}^{f_2(w_{t-1},w_t)} \alpha_{w_{t-2},w_{t-1},w_t}^{f_3(w_{t-2},w_{t-1},w_t)}}{Z_{\underline{\alpha}}(\tilde{d}_{t-1}, w_{t-2}, w_{t-1})} \\ &\quad \times \alpha_{\tilde{d}_{t-1},w_t}^{f_{\text{LSA}}(\tilde{d}_{t-1},w_t)} \end{aligned} \tag{8.4}$$

in which semantic coherence between a word w_t and its long-span history $\tilde{d}_{t-1}$ is treated as a feature, just like the standard N-gram features $f_1(w_t)$, $f_2(w_{t-1}, w_t)$ and $f_3(w_{t-2}, w_{t-1}, wt)$. *E.g.*,

$$f_{\text{LSA}}(\tilde{d}_{t-1}, w_t) = K(w_t, \tilde{d}_{t-1})\,. \tag{8.5}$$

We then find the maximum likelihood estimate of the model parameters given the training data. Recall that the resulting model is also the *maximum entropy* (ME) model among models which satisfy constraints on the marginal probabilities or expected values of these features [75].

An important decision that needs to be made in a model such as (8.4) regards the parameterization $\underline{\alpha}$. In a traditional ME language model, in the absence of LSA-based features each N-gram feature function is a $\{0,1\}$-valued indicator function, and there is a parameter associated with each feature: an α_w for each unigram constraint, an $\alpha_{w',w}$ for each bigram constraint, *etc.* In extending this methodology to the LSA features, we note that $K(w_t, \tilde{d}_{t-1})$ is continuous-valued. That in itself is not a problem; the ME framework does not require the $f(\cdot)$'s to be binary. What is problematic, however, is the fact that, almost surely, no two pseudo-documents $\tilde{d}_t$ and $\tilde{d}_{t'}$ will ever be the same. Therefore, assigning a distinct parameter $\alpha_{\tilde{d},w}$ for each pseudo-document – word pair $(\tilde{d}, w)$ is counterproductive. At least three alternatives present themselves naturally:

$$\alpha_{\tilde{d},w} = \alpha_{\text{LSA}} \qquad \forall\, w \in \mathbf{V} \text{ and } \tilde{d} \in \mathbb{R}^R\,, \tag{8.6}$$

which makes (8.4) comparable to the similarity modulated N-gram model above, except for the data-driven choice of α_{LSA} *jointly* with the other α's;

$$\alpha_{\tilde{d},w} = \alpha_{\text{LSA},w} \qquad \forall\, \tilde{d} \in \mathbb{R}^R\,, \tag{8.7}$$

which makes (8.4) comparable to the geometric interpolation described above, again except for the data-driven choice of $\alpha_{\text{LSA},w}$ *jointly* with the other α's;

$$\alpha_{\tilde{d},w} = \alpha_{\hat{d},w} \qquad \forall\, \tilde{d} \in \Phi(\hat{d}) \subset \mathbb{R}^R\,, \tag{8.8}$$

where $\Phi(\hat{d})$ represents a finite partition of $\mathbb{R}^R$ indexed by $\hat{d}$. We choose to pursue this alternative.

We use a standard K-means clustering of the representations $v_j S$ of the training documents d_j, with $K(d_j, d_{j'})$ in the role of distance, to obtain a modest number of clusters. We then pool documents in each cluster together to form *topic-centroids*

$\hat{d}$, and the partition $\Phi(\cdot)$ of $\mathbb{R}^R$ is defined by the *Voronoi regions* around the topic-centroids:

$$\Phi(\hat{d}) = \left\{ \tilde{d} \,:\, K(\tilde{d}, \hat{d}) \leq K(\tilde{d}, \hat{d}') \,\forall \,\text{centroids}\, \hat{d}' \right\}.$$

We also make two approximations to the feature function of (8.5). First, we *approximate* the pseudo-document $\tilde{d}_{t-1}$ in $K(\cdot)$ with its nearest topic-centroid $\hat{d}_{t-1} = \hat{d}$ whenever $\tilde{d}_{t-1} \in \Phi(\hat{d})$. This is motivated by the fact that we often deal with very small pseudo-documents $\tilde{d}$ in speech recognition, and $\hat{d}$ provides a more robust estimate of semantic coherence than $\tilde{d}$. Furthermore, keeping in mind the small dynamic range of the similarity measure of (7.7), as well as the interpretation (7.1) of ϵ_w, we *approximate* the feature function of (8.5) with

$$\hat{f}_{\text{LSA}}(\tilde{d}_{t-1}, w_t) = \begin{array}{ll} 1 & \text{if } K(w_t, \hat{d}_{t-1}) > \eta \text{ and } \epsilon_w < \tau, \\ 0 & \text{otherwise.} \end{array} \tag{8.9}$$

This pragmatic approximation results in a simplified implementation, particularly for the computation of feature-expectations during parameter estimation. More importantly, when there is a free parameter α for each $(\hat{d}, w)$ pair, *e.g.* (8.8), $\hat{f}_{\text{LSA}}(\hat{d}, w) = 1$ and $\hat{f}_{\text{LSA}}(\hat{d}, w) = K(w, \hat{d})$ yield equivalent model families. Therefore, using

$$\alpha_{\hat{d}_{t-1}, w_t}^{1 \text{ or } 0} \text{ instead of } \alpha_{\hat{d}_{t-1}, w_t}^{K(w_t, \hat{d}_{t-1})} \tag{8.10}$$

in (8.4) simply amounts to doing feature selection.

For all pairs $(\hat{d}, w)$ with $\hat{f}_{\text{LSA}}(\hat{d}, w) = 1$ in (8.9), the model-expectation of $\hat{f}$ is constrained to be the relative frequency of w within the cluster of training documents whose centroid is $\hat{d}$. By virtue of their semantic coherence, it is usually higher than the relative frequency of w in the entire corpus.

Another interesting way of parameterizing (8.4) which we have *not* investigated here is

$$\alpha_{\tilde{d}, w} = \alpha_{\hat{d}, \hat{w}} \quad \forall\, w \in \Psi(\hat{w})\,,\, \forall\, \tilde{d} \in \Phi(\hat{d})\,, \tag{8.11}$$

where $\Psi(\hat{w})$ is a finite, possibly $\hat{d}$-dependent, partition of the vocabulary. This parameterization may be particularly beneficial when, due to a very large vocabulary, small training corpus, or other factors, we do not have sufficient counts to constrain

the model-expectations of $\hat{f}_{\text{LSA}}(\hat{d}, w)$ for all words w bearing high semantic similarity with a topic-centroid $\hat{d}$.

8.4.1 A Similar ME Model from the Past

An interesting consequence of (8.9) is that it makes the model of (8.4) identical in form to the model described by Khudanpur and Wu [35]. Two significant ways in which (8.4) is novel include

- clustering of documents d_j to obtain topic-centroids $\hat{d}$ during training, and assignment of pseudo-documents $\tilde{d}_{t-1}$ to topic-centroids $\hat{d}_{t-1}$ during recognition, is based on similarity in LSA-space $\mathbb{R}^R$, not document-space $\mathbb{R}^M$, and

- the set of words with active semantic features (8.9) for any particular topic-centroid $\hat{d}$ is determined by a threshold η on LSA similarity, not by a difference in within-topic v/s corpus-wide relative frequency.

The former novelty results in considerable computational savings during both clustering and on-line topic assignment. The latter may result in a different choice of topic-dependent features.

Chapter 9

Experimental Results of LSA-Based Language Models

We now present the experimental results of the use of a language model on the Switchboard corpus of conversational speech.

9.1 Corpus

We conducted experiments on the Switchboard corpus of conversational telephone speech [32], dividing the corpus into an LM training set of approximately 1500 conversations (2.2M words) and a test set of 19 conversations (20K words). The task vocabulary was fixed to 22K words, with an out-of-vocabulary rate under 0.5% on the test set. Acoustic models trained on roughly 60 hours of Switchboard speech, a bigram LM was used to generate lattices for the test utterances, and then a 100-best list was generated by rescoring the lattice using a trigram model. All results in this chapter are based on rescoring this 100-best list with different language models.

We treated each conversation-side as a separate document and created W of (7.2) with $M \approx 22,000$ and $N \approx 3000$. Guided by the fact that one of 70-odd topics was prescribed to a caller when the Switchboard corpus was collected, we computed the SVD of (7.3) with R=73 singular values. We implemented the LSA model of (8.1)

[1]The work in this chapter was done under Prof. S. Khudanpur's supervision.

with $\gamma = 20$, and the four LSA + N-gram combinations of Section 8.3.

To obtain the document clusters and topic-centroids $\hat{d}$ required for creating the partition $\Phi(\cdot)$ of (8.8), we randomly assigned the training documents to one of 50 clusters and used a K-means algorithm to iteratively (i) compute the topic-centroid $\hat{d}$ of each cluster by pooling together all the documents in the cluster and then (ii) reassigning each document d_j to a cluster to whose centroid the document in question bore the greatest LSA similarity $K(d_j, \hat{d})$. Each cluster was required to have a minimum number of 10 documents in it, and if the number of documents in a cluster fell below this threshold following step (ii) the cluster was eliminated and each of its documents reassigned to the nearest of the remaining centroids. The iteration stopped when no reassignments resulted in step (ii). This procedure resulted in 25 surviving centroids, and we checked to be certain that the clusters were reasonably coherent by conducting a cursory examination of the documents.

For each topic-centroid $\hat{d}$ we chose, according to (8.9), a set of words that activates an LSA feature. We used $\tau = 0.4$ to first eliminate stop-words and then set a $\hat{d}$-specific η to yield ∼800 vocabulary-words above threshold per $\hat{d}$. However, not all these words actually appeared in training documents in $\Phi(\hat{d})$. Only the seen words were chosen, obtaining an average of 750 topic-dependent features for each topic-centroid. The resulting model had 19K $\alpha_{\hat{d},w}$ parameters associated with the semantic features in addition to about 22K unigram α_w's, 300K bigram $\alpha_{w',w}$'s and 170K trigram $\alpha_{w'',w',w}$'s. An ME language model was trained with these parameters using the toolkit developed by Wu [87].

9.2 Perplexity: LSA + N-gram Models

We used the CMU-CU LM toolkit to implement a baseline trigram model with Good-Turing discounting and Katz back-off. We then measured the perplexity of the reference transcription of the test conversations for the trigram and the four LSA + N-gram models of Section 8.3. The pseudo-document $\tilde{d}_{t-1}$ was updated according to (8.3) with $\lambda = 0.97$ for all four models. We used $\alpha = 0.1$ for the linear interpolation of the LSA and N-gram models. The other three combination techniques

require no additional parameters. The relative performance of the four schemes,

Language Model	Perplexity
CMU-CU Standard Trigram	81.1
LSA + Trigram Linear Interpolation	81.8
Similarity Modulated Trigram	79.1
Info Weighted Arithmetic Mean	81.8
Info Weighted Geometric Mean	75.8

Table 9.1: Perplexities: N-gram + LSA Combination

reported in Table 9.1, is consistent with the results of Coccaro and Jurafsky [15], with the information-weighted geometric interpolation showing the greatest reduction in perplexity. However, the reduction in perplexity is much smaller on this corpus than, *e.g.*, that reported by Bellegarda [2] on a text corpus.

9.3 Effect of Replacing $\tilde{d}_{t-1}$ with $\hat{d}_{t-1}$

We next describe our attempt to gain some understanding of the effect of replacing the pseudo-document $\tilde{d}_{t-1}$ with the closest topic-centroid $\hat{d}_{t-1}$ before the similarity computation in (8.9). For several of our test conversation-sides, we computed $K(w_t, \tilde{d}_{t-1})$ and $K(w_t, \hat{d}_{t-1})$, $t = 1, \ldots, T$, where w_t denotes the word in the t-th position and T denotes the number of words in the conversation-side.

For a *typical* conversation side in our test set, these similarities are plotted as a function of t in the box at the top of Figure 9.1. The second box shows the difference $K(w_t, \hat{d}_{t-1}) - K(w_t, \tilde{d}_{t-1})$. It is clear from the second box that $\hat{d}_{t-1}$ bears a greater similarity to the next word than $\tilde{d}_{t-1}$, confirming the beneficial effect of replacing $\tilde{d}_{t-1}$ with $\hat{d}_{t-1}$. We also computed $K(w_t, \hat{d}_T)$, the similarity of w_t with the topic-centroid most similar to the entire conversation side, and the box at the bottom of Figure 9.1 depicts the difference $K(w_t, \hat{d}_T) - K(w_t, \hat{d}_{t-1})$. We note with some satisfaction that as the conversation proceeds, the dynamically computed topic-centroid $\hat{d}_{t-1}$ converges to $\hat{d}_T$. Our conversation-sides are 470 words long on average, and we observe a convergence of roughly 110 words into the conversation side.

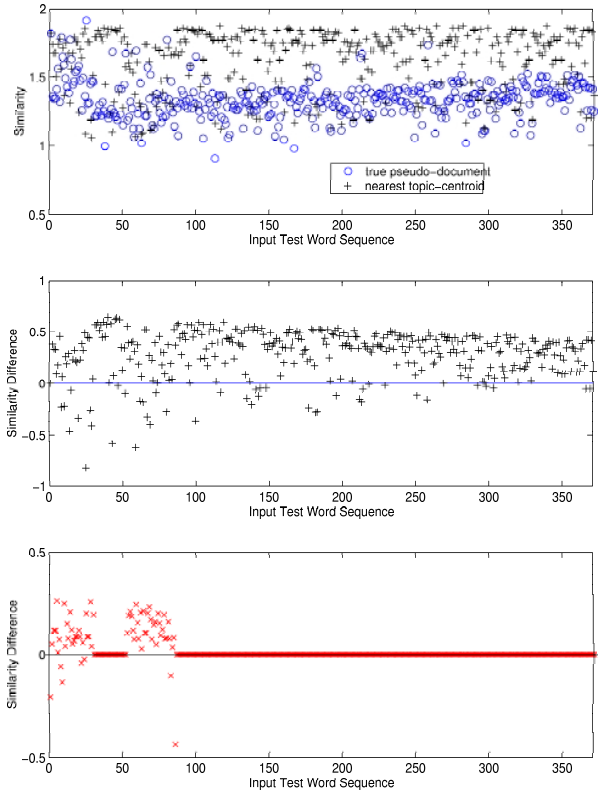


Figure 9.1: $K(w_t, \hat{d}_{t-1})$ and $K(w_t, \tilde{d}_{t-1})$ through a conversation (TOP), $K(w_t, \hat{d}_{t-1}) - K(w_t, \tilde{d}_{t-1})$ (MIDDLE), and $K(w_t, \hat{d}_T) - K(w_t, \hat{d}_{t-1})$ (BOTTOM).

9.4 Perplexity: ME Model with LSA Features

In the process of comparing our ME model of (8.4) with the one described by Khudanpur and Wu [35], we noticed that they built a baseline trigram model using the SRI LM toolkit. Other than this, our experimental setup – training and test set definitions, vocabulary, etc. – matches theirs exactly. We report the perplexity of our ME model against their baseline in Table 9.2, where the figures in the first two lines are quoted directly from Khudanpur and Wu [35]. A single topic-centroid $\hat{d}_T$ selected for an entire test conversation-side was used in these experiments. The last line of Table 9.2 shows the best perplexity obtainable by any topic-centroid, suggesting that the automatically chosen Voronoi region based topic-centroids are quite adequate.

A comparison of Tables 9.1 and 9.2 also shows that the maximum entropy model is more effective in capturing semantic information than the information-weighted

Language Model	Perplexity
SRI Trigram	78.8
ME Trigram	78.9
ME + LSA Features (Closest $\hat{d}_T$)	73.6
ME + LSA Features (Oracle $\hat{d}_T$)	73.0

Table 9.2: Perplexities: Maximum Entropy Models

geometric mean of the LSA-based unigram model and the trigram model. The correspondence of the information-weighted geometric mean with the parameterization of (8.7) and the corresponding richer parameterization of (8.8) perhaps adequately explain this improvement.

9.5 Word Error Rates for the ME Model

We rescored the 100-best hypotheses generated by the baseline trigram model using the ME model with LSA features. In order to assign a topic-centroid $\hat{d}$ to an utterance, we investigated taking the best, 10-best, or 100-best first-pass hypotheses of each utterance in the test set, computed $\hat{d}$ once per test utterance, and found the performance of the 10-best hypotheses to yield a slightly lower word error rate (WER). This is perhaps the optimal trade-off between the robustness in topic assignment that results from the consideration of additional word hypotheses and the noise introduced by the consideration of erroneous words. We also investigated assigning topics for the entire conversation side based on the first-pass output and found that to yield a further reduction in WER. We report the results in Table 9.3 where the top two lines are, again, quoted directly from Khudanpur and Wu [35]. We performed a paired sentence-level significance test on the outputs of these systems and found that the WER improvement of the ME model with

- only N-gram features over the baseline trigram model is somewhat significant at p=0.019;
- LSA features and utterance-level topic assignment over the ME model with only N-gram features is significant at p=0.005;

Language Model ($\hat{d}_T$ Assignment)	WER
SRI Trigram	38.47%
ME Trigram	38.32%
ME+LSA (per utterance via 10-best)	37.94%
ME+LSA (per conv-side via 10-best)	37.86%

Table 9.3: Error Rates: Maximum Entropy Models

- LSA features and conversation-level topic assignment over the ME model with only N-gram features is significant at p=0.001.

The difference between the ME models with utterance-level and conversation-level topic assignment is not significant (p=0.036), and differences between using the 1-v/s 10- v/s 100-best hypotheses for topic assignment were insignificant.

9.5.1 Benefits of Dimensionality Reduction

It was pointed out in Section 8.4.1 that the model proposed here differs from the model of Khudanpur and Wu [35] mainly in its use of R-dimensional LSA-space for similarity comparison rather than direct comparison in M-dimensional document-space. We present in Table 9.4 a summary comparison of the two modeling techniques. While owing to the sparse nature of the vectors the 22K-dimensional space

Attribute	Model A	Model B
Similarity measure	cosine	
Document clustering	K-means	
Vector-space dimension	22K	73
Num. topic-centroids	67	25
Avg. # topics/topic-word	1.8	1.3
Total # topic-parameters	15500	19000
ME + topic perplexity	73.5	73.6
ME + topic WER	37.9%	

Table 9.4: A comparison between the model (A) of Khudanpur and Wu [35] and our model (B).

does not entail a proportional growth in similarity computation relative to the 73-dimensional space, the LSA similarities are still expected to offer faster computing. Furthermore, the LSA-based model yields comparable perplexity and WER performance with considerably fewer topic-centroids, resulting in fewer comparisons during run time for determining the nearest centroid. Of lesser note is the observation that the η-threshold based topic-feature selection of (8.9) results in a content word being an active feature for fewer topics than it does when topic-features are selected based on differences in within-topic and overall relative frequencies.

9.6 Summary

We have presented a framework for incorporating latent semantic information together with standard N-gram statistics in a unified exponential model for statistical language modeling. This framework permits varying degrees of parameter tying depending on the amount of training data available. We have drawn parallels between some conventional ways of combining LSA-based models with N-grams and the parameter-tying decisions in our exponential models, and our results suggest that incorporating semantic information using maximum entropy principles is more effective than ad hoc techniques.

We have presented perplexity and speech recognition accuracy results on the Switchboard corpus which suggest that LSA-based features, while not as effective on conversational speech as on newspaper text, make produce modest but statistically significant improvements in speech recognition performance.

Finally, we have shown that the maximum entropy model presented here performs as well as a previously proposed maximum entropy model for incorporating topic-dependencies, even as it is computationally more economical.

Part III

Conclusions and Future Work

Chapter 10

Conclusions and Future Work

In this chapter, we summarize the work of this thesis and highlight its contributions. We also point to several possible directions for new research.

10.1 Thesis Summary

In the first part of this thesis, we investigated string-to-string bitext alignment models for statistical machine translation.

In Chapter 2, we addressed chunk alignment at the sentence or sub-sentence level, proposing a generative chunk alignment model to model the relationship between document pairs. Under this framework, two very different alignment algorithms were derived straightforwardly by varying the component distribution. One was the widely used Dynamic Programming (DP) algorithm, which finds the global optimal monotone chunk alignments. The other was the Divisive Clustering (DC) algorithm, which derives short chunk pairs by parallel binary splitting iteratively. DC is a divide and conquer approach. Swapping is allowed in DC to capture non-monotone orders. After analyzing and comparing the two algorithms, we proposed a hybrid approach called DP+DC, where the DP algorithm is applied to identify chunk alignments at sentence level and the DC algorithm is applied to extract sub-sentential alignments.

Chapter 3 investigated statistical word alignment models. We compared the HMM-based word-to-word alignment models and IBM fertility-based models by dis-

tinguishing their generative procedures and identifying their strengthes and weaknesses. We presented the word-to-phrase HMM alignment model, which has training and alignment algorithms as efficient as HMM-based model and at the same time exhibits word alignment quality comparable to that of the IBM Model-4. Features of Model-4 were incorporated within HMM to allow efficient training and alignment algorithms. Word context within phrases can be captured without losing algorithmic efficiency. An incremental training procedure was proposed to estimate model parameters gradually from scratch.

We studied phrase alignment models in Chapter 4 with a focus on extracting phrase pair inventory (PPI) from word aligned bitext. Since replying on one-best word alignments would exclude valid phrase pairs, we proposed a model-based phrase pair posterior distribution which allows more control over phrase pair extraction. With the goal of improving phrase pair coverage, we presented a simple PPI augmenting scheme based on the phrase pair posterior distribution.

A series of experiments have been designed to systematically study the usefulness and effects of the chunk alignment model on machine translation systems in Chapter 5. We found that better sentence alignments can be achieved with better translation lexicons. In the unsupervised sentence alignment experiment, we obtained balanced performance in precision and recall. With the DC procedure, we were able to derive chunk pairs at sub-sentence level. We found that this maintains translation performance of the resulting systems. The practical benefits of the approach include the faster training of MT systems and the capability to retain more of the available bitext in MT training.

Chapter 6 presented the experimental results of the statistical word-to-phrase (WtoP) HMM alignment model. In Chinese-English bitext word alignment experiments, the WtoP model performed comparably to Model-4 when measured by Alignment Error Rate (AER), even over large training bitexts. Increasing max phrase length N improved word alignment quality in the Chinese to English direction. We found that a balance between word-to-word and word-to-phrase link distributions could be obtained by setting model parameters properly to achieve overall optimal performance. In Arabic-English and Chinese-English translation, the WtoP model

compared well to Model-4, even with large bitexts. The model-based phrase pair posterior distribution enabled more control over phrase pair extraction than inducing just over word alignments. The augmented phrase pair inventory (PPI) improved coverage on test sets and translation performance. We found that the WtoP model can even be used to improve the PPI extracted from Model-4 word alignments. On the large Arabic-English translation system, the WtoP model performs significantly better than Model-4 with about 1∼2% absolute BLEU points improvements.

In Chapter 7, we reviewed the Latent Semantic Analysis technique, describing steps of extracting meaningful and compact representations of words and documents. In Chapter 8, we studied applications of LSA to statistical language modeling. We proposed a novel tight integration of latent semantic information with local n-gram under the log-linear model. In Chapter 9, we presented the experimental results of perplexity evaluation and speech recognition accuracy on the Switchboard conversational tasks. Our results showed that incorporating semantic information using maximum entropy principles is more effective than the ad hoc techniques. LSA-based features produce modest but statistically significant improvements in speech recognition performance.

10.2 Suggestions for Future Work

10.2.1 Word-to-Phrase HMM

We note that the WtoP model is still relatively simple, and we anticipate further alignment and translation improvement as such models are refined.

HMM is attractive insofar as it provides efficient parameter estimation and inference algorithms for modeling various sequences. Still, the framework leaves much space for further exploitation. Experience from automatic speech recognition can be borrowed in machine translation applications. For instance, alignment lattice exploitation and translation model adaptation are two immediate possibilities. Discriminative training techniques that have been successfully applied in automatic speech recognition can also be examined in statistical word alignment models under the same

HMM framework.

The Markov assumption implies locality, consequently enabling efficient dynamic-programming based training and alignment algorithms. However, its limited memory impedes the exploitation of global constraints. For instance, there is no guarantee that every state will be visited in the generation of target sequences. When a state is not visited, it is not penalized in the conditional likelihood function. On the contrary, under the IBM Model-4 when a source word s aligns to no target words, the fertility contribution from that word is $n(\phi = 0; s)$, which is part of the conditional likelihood function. Therefore, IBM Model-4 encourages the connection of source words to target words. This feature can not be introduced into an HMM framework directly without disturbing the Markov property. However, it can be expressed along with other features during post-processing on word alignment lattice.

10.2.2 Phrase-to-Phrase HMM

Probably the most obvious extension of this thesis work would be to model phrase-to-phrase alignments directly within HMM architecture. Like the joint model in [57], such modeling would allow a phrase translation table to be generated directly from model training rather than be induced from word alignments only.

In Appendix A, we formulate a phrase to phrase alignment model under Markov framework. One difficulty therein inheres in deciding the phrase vocabulary for source and target languages. This is a practical issue also challenging robust parameter estimation. Possible choices can come from frequent n-grams or linguistically motivated constituents. It can be expected that severe data sparseness problem would arise in estimating the phrase-to-phrase translation table. To address these, internal structures between words or word clustering technologies can be explored, as in alignment template [65].

10.2.3 Beyond String-to-String Alignments

In this thesis, we focused on string-to-string alignments at different granularity. Models are formulated without acknowledging many linguistic constraints. Several

extensions may permit further study of the impact of linguistic features. In the word-to-phrase model, any possible phrase segmentations of target sentences are considered; instead of this procedure, a shallow parsing on the target sentence can be performed initially to identify noun/verb/propositional phrases, and during the Markov transition procedure, phrase boundaries can be obeyed.

Rather than string-to-string alignments, alignments between strings and trees pose interesting and challenging tasks to syntax-based machine translation. Sequence alignment models in this thesis can serve as starting points for alignments between structure and sequence or structure and structure, for instance, as in bilingual synchronized parsing [13].

10.2.4 Language Modeling

Language models (LM) play an important role in statistical machine translation (SMT) and automatic speech recognition (ASR). In section 6.4, we showed how SMT system can be improved with a more powerful language model. As in ASR, language models are typically trained and optimized separately from translation models in SMT. A systematic study of language model in SMT tasks would provide guidelines on how to tune a statistical language model into a SMT system. It is desirable to exploit domain or topic information specially designed for SMT tasks.

Appendix A

HMM-based Phrase-to-Phrase Alignment Models

Since word-to-phrase HMM alignment model is only half way toward a phrase-to-phrase alignment model, we formulate a general HMM-based Phrase-to-Phrase alignment model to capture phrase alignments between bitext.

Let $\mathbf{s} = (s_1, s_2, \cdots, s_I)$ and $\mathbf{t} = (t_1, t_2, \cdots, t_J)$ be the given parallel word strings in two language. Since we are interested in building a model that will assign probability to $\mathbf{t}$ given $\mathbf{s}$ and find words in $\mathbf{s}$ that are responsible to generate words in $\mathbf{t}$, the alignments between them are from $\mathbf{s}$ to $\mathbf{t}$. Without ambiguity, we shall call $\mathbf{s}$ the word string in source language and $\mathbf{t}$ the word string in target language.

Let the random variable $\mathbf{u}^{(p)} = (u_1, u_2, \cdots, u_p)$ be a phrasal segmentation of the source word string $\mathbf{s}$. Each u_k is a word or word sequence. The concatenation of the word sequence of each u_p would yield the word string of $\mathbf{s}$. Similarly, let the random variable $\mathbf{v}^{(q)} = (v_1, v_2, \cdots, v_q)$ be a phrasal segmentation of target word string $\mathbf{t}$. Let $\mathbf{a}^{(p,q)} = (a_1, a_2, \cdots, a_q)$ be an alignment from $\mathbf{u}^{(p)}$ to $\mathbf{v}^{(q)}$, where $0 \leq a_k \leq p$ is the index of source phrase in $\mathbf{u}^{(p)}$ that is responsible to generate phrase v_k, $1 \leq k \leq q$.

The likelihood of incomplete data is the summation of conditional likelihood function over all possible hidden variables,

$$P(\mathbf{t}|\mathbf{s}) = \sum_{p,q,\mathbf{u}^{(p)},\mathbf{v}^{(q)},\mathbf{a}^{(p,q)}} P(\mathbf{t}, p, q, \mathbf{u}^{(p)}, \mathbf{v}^{(q)}, \mathbf{a}^{(p,q)}|\mathbf{s}). \tag{A.1}$$

Applying the chain rule, the likelihood of the complete data is

$$
\begin{aligned}
P(\mathbf{t}, p, q, \mathbf{u}^{(p)}, \mathbf{v}^{(q)}, \mathbf{a}^{(p,q)}|\mathbf{s}) &= P(p|\mathbf{s}) && \text{(A.2)}\\
&\quad \times P(\mathbf{u}^{(p)}|\mathbf{s}, p) && \text{(A.3)}\\
&\quad \times P(q|\mathbf{s}, p, \mathbf{u}^{(p)}) && \text{(A.4)}\\
&\quad \times P(\mathbf{a}^{(p,q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q) && \text{(A.5)}\\
&\quad \times P(\mathbf{v}^{(q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q, \mathbf{a}^{(p,q)}) && \text{(A.6)}\\
&\quad \times P(\mathbf{t}|\mathbf{s}, p, \mathbf{u}^{(p)}, q, \mathbf{a}^{(p,q)}, \mathbf{v}^{(q)}) && \text{(A.7)}
\end{aligned}
$$

Each component, along with its assumptions, is constructed in the following way.

- **Source Phrase Number Model** $P(p|\mathbf{s})$ is the probability that source string $\mathbf{s}$ is segmented into p phrases. Valid p takes value from 1 to I. We assume that p depends on the number of total source words only: $P(p|\mathbf{s}) = P(p|I)$

- **Source Phrase Model** $P(\mathbf{u}^{(p)}|\mathbf{s}, p)$ decides how to segment source word string into phrases. This can be based on source phrase language model with normalization.

- **Target Phrase Number Model** $P(q|\mathbf{s}, p, \mathbf{u}^{(p)})$ specifies the number of target phrases to be generated. A simple model assumes that q depends only on q: $P(q|\mathbf{s}, p, \mathbf{u}^{(p)}) = P(q|p)$

- **Markov Transition Model** $P(\mathbf{a}^{(p,q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q)$ assigns a probability mass to each valid alignment $\mathbf{a}^{(p,q)}$. We assume first order Markov

$$
\begin{aligned}
P(\mathbf{a}^{(p,q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q) &= P(a_1^q|\mathbf{s}, p, \mathbf{u}^{(p)}, q) && \text{(A.8)}\\
&= \prod_{j=1}^{q} P(a_j|a_1^{j-1}, \mathbf{s}, p, \mathbf{u}^{(p)}, q) && \text{(A.9)}\\
&= \prod_{j=1}^{q} P(a_j|a_{j-1}, \mathbf{s}, p, \mathbf{u}^{(p)}, q) && \text{(A.10)}\\
&= \prod_{j=1}^{q} a(a_j|a_{j-1}, p, q) && \text{(A.11)}
\end{aligned}
$$

The last equation ignores source phrases, which means that the underlying Markov model makes state transition without examining the phrases it generates. A generalized factorization would be:

$$P(\mathbf{a}^{(p,q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q) = \prod_{j=1}^{q} a(a_j|a_{j-1}, p, q, u_{j-1}) \quad \text{(A.12)}$$

We make the state transition depend on context. The next state is decided by the current state and the phrase it generates.

- **Phrase Translation Model** $P(\mathbf{v}^{(q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q, \mathbf{a}^{(p,q)})$ gives the probability of target phrase sequence given source phrases and alignment. We assume that target phrases are generated independently and that each target phrase is responsible for only the aligned source phrase. Therefore

$$P(\mathbf{v}^{(q)}|\mathbf{s}, p, \mathbf{u}^{(p)}, q, \mathbf{a}^{(p,q)}) = \prod_{k=1}^{q} P(v_k|u_{a_k}) \quad \text{(A.13)}$$

$$= \prod_{k=1}^{q} t(v_k|u_{a_k}) \quad \text{(A.14)}$$

where $t(v|u)$ are phrase translation lexicons, which are similar to word to word translation tables but defined over phrase pairs. Presumably, there should be a phrase inventory for both languages. This can be frequent n-grams. Typically, data spareness is a challenge for robust estimation. Clustering techniques or word linkage structures within phrase pairs can be introduced.

- **Target String Reconstruction Model** $P(\mathbf{t}|\mathbf{s}, p, \mathbf{u}^{(p)}, q, \mathbf{a}^{(p,q)}, \mathbf{v}^{(q)})$ essentially is a delta function, which is 1 if the concatenation of words in $\mathbf{v}^{(q)}$ is $\mathbf{t}$ and 0 otherwise.

Bibliography

[1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[2] J. R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296, August 2000.

[3] A. L. Berger, S. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[4] M. Berry, T. Do, and S. Varadhan. Svdpackc (version 1.0) user's guide. Tech. report cs-93-194, University of Tennessee, Knoxville, TN, 1993.

[5] M. Braschler and P. Schäuble. Multilingual information retrieval based on document alignment techniques. In *ECDL*, pages 183–197, 1998.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees*. Wadsworth International Group, 1984.

[7] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312, 1993.

[8] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), December 1992.

[9] P. F. Brown, J. C. Lai, and R. L. Mercer. Aligning sentences in parallel corpora. In *Meeting of the Association for Computational Linguistics*, pages 169–176, 1991.

[10] S. F. Chen. Aligning sentences in bilingual corpora using lexical information. In *Meeting of the Association for Computational Linguistics*, pages 9–16, 1993.

[11] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318, 1996.

[12] S. F. Chen, K. Seymore, and R. Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models. In *IEEE ICASSP*, pages 681–684. IEEE, 1998.

[13] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[14] P. Clarkson and A. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proc. ICASSP*, pages 799–802. IEEE, 1997.

[15] N. Coccaro and D. Jurafsky. Towards better integration of semantic predictors in statistical language modeling. In *Proc. ICSLP*, pages 2403–2406, Sydney, Australia, 1998.

[16] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[17] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society*, 39, 1977.

[18] Y. Deng and W. Byrne. Hmm word and phrase alignment for statistical machine translation. In *Proc. of HLT-EMNLP*, 2005.

[19] Y. Deng and S. Khudanpur. Latent semantic information in maximum entropy language models for conversational speech recognition. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 56–63, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[20] Y. Deng, S. Kumar, and W. Byrne. Bitext chunk alignment for statistical machine translation. Clsp tech report 50, Johns Hopkins University, April 2004.

[21] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of HLT 2002*, San Diego, CA. USA, 2002.

[22] S.T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

[23] M. Epstein. *Statistical Source Channel Models for Natural Language Understanding*. PhD thesis, New York University, September 1996.

[24] M. Epstein, K. Papineni, S. Roukos, T. Ward, and S. Della Pietra. Statistical natural language understanding using hidden clumpings. In *Proceedings of ICASSP*, volume 1, pages 176–179, Atlanta, GA, May 1996.

[25] M. Snover *et al.* Study of translation error rate with targeted human annotation. In *Machine Translation Workshop*, North Bethesda, MD, 2005. NIST.

[26] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184, 1991.

[27] M. Gamon, A. Aue, and M. Smets. Sentence-level mt evaluation without reference translations: Beyond language modeling. In *Proceeding of EAMT Conference*, pages 103–111, 2005.

[28] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Meeting of the Association for Computational Linguistics*, pages 228–235, 2001.

[29] M. Haruno and T. Yamazaki. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of ACL '96*, pages 131–138, 1996.

[30] E. Hovy. Towards finely differentiated evaluation metrics for machine translation. In *Proc. of the Eagles Workshop on Standards and Evaluation*, Pisa, Italy, 1999.

[31] R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. *IEEE Trans Speech and Audio Processing*, 7:30–39, 1999.

[32] J. McDaniel J. Godfrey, E. Holliman. Switchboard: telephone speech corpus for research and development. In *Proc. ICASSP*, pages 517–520, San Francisco, CA, 1992.

[33] F. Jelinek. *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, USA, 1997.

[34] M. D. Kernighan, K. W. Church, and W. A. Gale. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics*, pages 205–210, Morristown, NJ, USA, 1990. Association for Computational Linguistics.

[35] S. Khudanpur and J. Wu. A maximum entropy language model integrating n-grams and topic dependencies for conversational speech reconition. In *Proc. of ICASSP*, Phoenix, AZ, 1999.

[36] W. Kim and S. Khudanpur. Cross-lingual lexical triggers in statistical language modeling. In *Proc. of EMNLP*, pages 17–24, Sapporo, Japan, 2003.

[37] R. Kneser and H. Ney. Improved backing-o# for m-gram language modeling. In *Proceedings of ICASSP*, volume 1, pages 181–184, 1995.

[38] K. Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, 1999.

[39] K. Knight and D. Marcu. Machine translation in the year 2004. In *Proc. of ICASSP*, 2005.

[40] P. Koehn, F. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, 2003.

[41] O. Kolak, W. J. Byrne, and P. Resnik. A generative probabilistic ocr model for nlp applications. In *HLT-NAACL*, 2003.

[42] S. Kumar, Y. Deng, and W. Byrne. A weighted finite state transducer translation template model for statistical machine translation. *Journal of Natural Language Engineering*, 11(3), 2005.

[43] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semanctic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–140, 1997.

[44] LDC. *Buckwalter Arabic Morphological Analyzer Version 1.0*, 2002. LDC Catalog Number LDC2002L49.

[45] LDC. *Chinese Segmenter*, 2002. http://www.ldc.upenn.edu/Projects/Chinese.

[46] LDC. *Sinorama Chinese-English Parallel Text*, 2002. LDC Catalog Number LDC2002E58.

[47] LDC. *Xinhua Chinese-English Parallel News Text Version 1.0 beta 2*, 2002. LDC Catalog Number LDC2002E18.

[48] LDC. *Chinese Treebank English Parallel Corpus*, 2003. LDC Catalog Number LDC2003E07.

[49] LDC. *FBIS Chinese-English Parallel Corpus*, 2003. LDC Catalog Number LDC2003E14.

[50] LDC. *Hong Kong Hansard Parallel Text*, 2003. LDC Catalog Number LDC2004E09.

[51] LDC. *Hong Kong News Parallel Text*, 2003. LDC Catalog Number LDC2003E25.

[52] LDC. *UN Chinese-English Parallel Text Version 2*, 2004. LDC Catalog Number LDC2004E12.

[53] D. Lin. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324, 1999.

[54] Y. Linde, A. Buzo, and R.Gray. An algorithm for vector quantizer design. *IEEE Transaction on Communications*, 28(1):84–94, 1980.

[55] X. Ma, C. Cieri, and D. Miller. Corpora & tools for machine translation. In *Machine Translation Evaluation Workshop*, Alexandria, VA, 2004. NIST.

[56] G. Mann and D. Yarowsky. Multipath translation lexicon induction via bridge languages. In *Proceedings of the Second Conference of the North American Association for Computational Linguistics*, Pittsburgh, PA, 2001.

[57] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP*, 2002.

[58] I. D. Melamed. A portable algorithm for mapping bitext correspondence. In *Proceedings of the 35th Annual Conference of the Association for Computational Linguistics.*, pages 305–312, 1997.

[59] I. D. Melamed. Models of translational equivalence among words. *Comput. Linguist.*, 26(2):221–249, 2000.

[60] I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In *Proceedings of the HLT-NAACL*, Edmonton, Canada, 2003.

[61] T. Miller. Essay assessment with latent semantic analysis. *Journal of Educational Computing Research*, 28(3), 2003.

[62] R. C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proceeding of 5th Conference of the Association for Machine Translation in the Americas*, pages 135–244, 2002.

[63] NIST. *The NIST Machine Translation Evaluations*, 2004. http://www.nist.gov/speech/tests/mt/.

[64] D. Oard. A survey of multilingual text retrieval. Technical Report UMIACS-TR-96-19 CS-TR-3615, University of Maryland, College Park, April 1996.

[65] F. Och. *Statistical Machine Translation: From Single Word Models to Alignment Templates*. PhD thesis, RWTH Aachen, Germany, 2002.

[66] F. Och. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, 2003.

[67] F. Och and H. Ney. Improved statistical alignment models. In *Proc. of ACL-2000*, pages 440–447, Hong Kong, China, 2000.

[68] F. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, College Park, MD, USA, 1999.

[69] F. J. Och. An efficient method for determining bilingual word classes. In *EACL*, pages 71–76, 1999.

[70] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[71] M. Ostendorf, V. Digalakis, and O. Kimball. From HMMs to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 4:360–378, 1996.

[72] 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.

[73] S. Della Pietra, M. Epstein, S. Roukos, and T. Ward. Fertility models for statistical natural language understanding. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 168–173, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[74] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, 1986.

[75] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228, 1996.

[76] M. Simard and P. Plamondon. Bilingual sentence alignment: Balancing robustness and accuracy. In *Machine Translation*, volume 13, pages 59–80, 1998.

[77] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, 2002.

[78] E. Sumita, Y. Akiba, T. Dio, A. Finch, K. Imamura, H. Okuma, M. Paul, M. Shimohata, and T. Watanabe. EBMT, SMT, Hybrid and More: ATR spoken language translation system. In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004.

[79] K. Toutanova, H. T. Ilhan, and C. Manning. Extentions to HMM-based statistical word alignment models. In *Proc. of EMNLP*, 2002.

[80] A. Venugopal, S. Vogel, and A. Waibel. Effective phrase translation extraction from alignment models. In *Proc. of ACL*, 2003.

[81] S. Vogel, H. Ney, and C. Tillmann. HMM based word alignment in statistical translation. In *Proc. of the COLING*, 1996.

[82] J. S. White and T. O'Connell. The arpa mt evaluation methodologies. In *Proc. of the AMTA Conference*, pages 193–205, Columbia, MD, USA, 1994.

[83] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. In *IEEE Trans. Inform Theory*, volume 37, pages 1085–1094, July 1991.

[84] D. Wu. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Meeting of the Association for Computational Linguistics*, pages 80–87, 1994.

[85] D. Wu. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Meeting of the Association for Computational Linguistics*, pages 244–251, 1995.

[86] D. Wu. Stochastic inversion transduction grammar and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–482, 1997.

[87] J. Wu. *Maximum entropy language modeling with nonlocal dependencies*. PhD thesis, Johns Hopkins University CS Department, Baltimore, MD, 2002.

[88] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proc. ACL*, 2001.

[89] K. Yamamoto, T. Kudo, Y. Tsuboi, and Y. Matsumoto. Learning sequence-to-sequence correspondences from parallel corpora via sequential pattern mining. In Rada Mihalcea and Ted Pedersen, editors, *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 73–80, Edmonton, Alberta, Canada, May 31 2003. Association for Computational Linguistics.

[90] Y. Yan and E. Barnard. An approach to language identification with enhanced language model. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech)*, 1995.

[91] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[92] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book, Version 3.1*, Dec. 2001.

[93] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Research and Development in Information Retrieval*, pages 334–342, 2001.

[94] Y. Zhang and S. Vogel. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the Tenth Conference of the European Association for Machine Translation*, 2005.

Vita

Yonggang Deng was born in JiangXi Province, People's Republic of China. He graduated from University of Science and Technology of China, Hefei, People's Republic of China, in 1997 with a B.S. in Computer Software (with honor). After receiving an M.S. degree in Pattern Recognition & Artificial Intelligence from Institute of Automation, Chinese Academy of Sciences in 2000, he started the Ph.D. program in Department of Electrical and Computer Engineering, the Johns Hopkins University, where he received an M.S.E. degree in Electrical and Computer Engineering in 2002. Since then, he has been pursing 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 2001, he interned at Department of Human Language Technology, IBM T. J. Watson Research Center. He also interned at Speech Technology Group, Microsoft Research, during the summer of 2002.

His research interests include statistical modeling and machine learning techniques, and their applications in machine translation, speech recognition, information retrieval and other problems in speech and language processing.