LANGUAGE MODEL ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION AND STATISTICAL MACHINE TRANSLATION

by

Woosung Kim

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

Baltimore, Maryland

October 2004

Abstract

Language modeling is critical and indispensable for many natural language applications such as automatic speech recognition and machine translation. Due to the complexity of natural language grammars, it is almost impossible to construct language models by a set of linguistic rules; therefore statistical techniques have been dominant for language modeling over the last few decades.

All statistical modeling techniques, in principle, work under some conditions: 1) a reasonable amount of training data is available and 2) the training data comes from the same population as the test data to which we want to apply our model. Based on observations from the training data, we build statistical models and therefore, the success of a statistical model is crucially dependent on the training data. In other words, if we don't have enough data for training, or the training data is not matched with the test data, we are not able to build accurate statistical models. This thesis presents novel methods to cope with those problems in language modeling—language model adaptation.

We first tackle the data deficiency problem for languages in which extensive text collections are not available. We propose methods to take advantage of a resource-rich language such as English, utilizing cross-lingual information retrieval followed by machine translation, to adapt language models for the resource-deficient language. By exploiting a copious side-corpus of contemporaneous ar-

ticles in English to adapt the language model for a resource-deficient language, significant improvements in speech recognition accuracy are achieved.

We next experiment with language model adaptation in English, which is resource-rich, in a different application: statistical machine translation. Regardless of its size, having training data that is not matched with the test data, which is of main interest, does not necessarily lead to accurate statistical models. Rather, we select small but effective texts by information retrieval, and use them for adaptation. Experimental results show that our adaptation techniques are effective for statistical machine translation as well.

Advisor: Professor Sanjeev Khudanpur

Readers: Professor Frederick Jelinek, Professor David Yarowsky and Professor Sanjeev Khudanpur

Acknowledgments

First of all, I wish to express my deepest appreciation to my advisor, Professor Sanjeev Khudanpur. I would say it was a huge amount of luck in my life that I had a chance to work with him. His guidance, encouragement, inspiration and excellent advice throughout my study at Hopkins finally led me to successfully complete my thesis.

Next, I would like to show my sincere gratitude to Professor David Yarowsky. From the time when I entered Hopkins, he has taught me everything from the beginning with great care and warm encouragement. He is one of the most gentle people I have ever worked with and I really appreciate his willingness to read my dissertation in spite of his busy schedule.

I would also like to thank Professor William Byrne. He was kind enough to allow me to use his invaluable resources for my automatic speech recognition and statistical machine translation experiments. Besides, he gave me valuable comments and feedbacks during the machine translation evaluations.

My special thanks are due to Professors Frederick Jelinek and Jason Eisner, Dr. Eric Brill, and all staff members of Center for Language and Speech Processing (CLSP). All the facilities and research environments of CLSP have been perfect during my Ph.D. study and I really appreciate their support and help.

All of my work would not have been possible without others' help. Especially, I

would like to thank Shankar Kumar for providing the N-best list for the automatic speech recognition experiments. He and Yonggang Deng helped me in generating the first pass N-best list for my statistical machine translation experiments. Paola Virga helped me build the baseline language models and provided the ranked list of information retrieval for the machine translation experiments.

Many thanks to the numerous colleagues in CLSP including Jia Cui, Sourin Das, Vlasios Doumpiotis, Elliott Drabek, Ahmad Emami, Arnab Ghoshal, Lambert Mathias, Srividya Mohan, Srihari Reddy, Charles Schafer, Stavros Tsakalidis, Veera Venkataramani, Peng Xu, Ali Yazgan, and Jun Wu. All the discussions, cooperations, and the moments I spent with them have been valuable as well as pleasing, and I will never forget the times I enjoyed with them.

I am deeply indebted to my family in Korea: my parents, parents-in-law, brothers, a brother-in-law and all relatives (especially my uncle). Whenever I had a hard time during my study, they encouraged me from far away and supported me with patience. I only wish to have a chance to pay them in return.

Finally and most of all, I would like to thank my wife: Soomyung and my two sons: Daehyun and Dohyun. Without their tireless sacrifice, support and encouragement, none of these would have been possible. After a long wait, their endurance now pays off. They are the ones who deserve the degree.

Woosung Kim

To Soomyung
and
in loving memory of my mom,

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Speech and natural language are the most *natural* means for communicating between humans and it would be ideal if we can use them for communicating with a computer. Speech recognition, for example, is a task of converting human's speech into text which enables us to *talk* to a computer. Potentially, therefore, it can be applied to *almost all* computer applications. Optical character recognition is another task which can free us from the burden of typing on a keyboard. Machine translation—which is yet another example in which a computer can be effectively used even for human to human communications via natural languages—has long been a research issue. There are many problems, though, that hinder us from using speech and natural languages for operating a computer or developing natural language-based applications, and people still need to type on a keyboard and move a mouse to use a computer—instead of *talking* or *writing*.

Among many roadblocks hindering us from developing speech and natural language-based applications, the most significant is the complexity of natural lan-

guage grammars. Even a child spends years to learn its first language, and it would take longer after one gets older when one's learning ability drops significantly. Evidently, the complexity of our natural languages is very high and it is extremely difficult to *teach* a natural language grammar to a computer.

Most approaches to teach the natural language grammar can be classified into two categories: rule-based approaches and stochastic or statistical approaches. The rule-based approach is to specify the natural language grammar as a set of rules which is accurate, but difficult to acquire or learn automatically. More specifically, all the rules need to be chosen carefully by some experts such as linguists and therefore it needs to be done manually. The statistical approach, on the other hand, models the language grammar via a set of parameters which can be learned or acquired automatically from a reasonable-size training data. Unlike the rule-based approach, the statistical approach has the advantage that extensive human knowledge or manual work is not needed.

There have been dramatic improvements in the capability and performance of speech and natural language processing systems based on the statistical approach over the last few decades. This progress may be largely attributed to advances in statistical modeling techniques and procedures for automatic learning from large speech and text corpora. The construction of increasingly accurate and complex stochastic models, in particular, is crucially dependent on the availability of large corpora of transcribed speech and annotated text specific to the language and the application domain. Much of these advances, therefore, have been in languages such as English, French, German and Japanese, and in domains such as air travel information and broadcast news transcription, for which such linguistic resources have been created at considerable cost.

Construction of accurate stochastic models for processing *resource-deficient*

languages has recently started receiving attention. Certainly, it is hard to expect that there is an extensive amount of training data for resource-deficient languages. A limited amount of linguistic resources, however, can almost always be produced with moderate effort in a language and domain of interest, and we use the term *resource-deficiency* to imply the lack of hundreds of hours of orthographically transcribed speech, hundreds of millions of words of in-domain text, hundreds of thousands of manually parsed sentences, *etc.*

Given a limited amount of training data, methods have been proposed to bootstrap acoustic models for automatic speech recognition (ASR) in resource-deficient languages by reusing acoustic models from resource-rich languages.

- The notion of a universal phone-set has been used, *e.*g., by Schultz and Waibel (1998), to jointly train acoustic models in multiple languages.

- Acoustic-phonetic models in the target language have been synthesized by Byrne et al. (2000) by matching well-trained models from resource-rich languages to a limited amount of transcribed speech in the target language.

Morphological analyzers, noun-phrase chunkers, part-of-speech taggers *etc.*, have been developed for resource-deficient languages by exploiting translated texts.

- Statistical models have been used by Yarowsky et al. (2001) to align words in a sentence in the target language with words in, say, the English translation of the sentence; the English side is automatically annotated for the necessary categories (POS tags, NP brackets), and the annotation is projected to the target language via the alignment, producing a "labeled" corpus in the resource-deficient language, from which necessary statistical models are then estimated.

This dissertation first proposes novel techniques for estimating a language model (LM) which can be used for ASR, machine translation (MT), and other natural language applications when a large amount of training data is not available, which is typically the case in many (resource-deficient) languages. When an ASR system needs to be engineered for a specific domain in a new language (e.g., Arabic news broadcasts), a modest amount of domain specific LM training text is usually made available, from which a word-list and a small N-gram LM may be derived. Additional target language text from an unrelated domain (e.g., Arabic web pages) may sometimes be available, and its use to improve performance in the target language and domain has been investigated elsewhere (e.g., Berger and Miller, 1998; Scheytt et al., 1998). Abundant domain-specific text in *other* languages (e.g., English news broadcasts) is also often available. Furthermore, for several languages with a sub-par electronic presence, the amount of English text in the domain of interest is likely to far outweigh the amount of in-language text from all domains. In this dissertation we investigate methods to use such cross-language in-domain data to improve the LM via language model adaptation.

Furthermore, we apply our cross-lingual language model adaptation methods even to a resource-rich language, English, where an extensive amount of data for training is available. Regardless of the size of training data, having training data that is not close or similar to the test data which is of interest does not necessarily lead to accurate statistical models. For example, an LM trained with weather forecasts would not work well predicting test data about financial news—no matter how much weather forecast texts are available. By carefully selecting effective or relevant texts to the test data and building adaptive LMs, we show that a significant gain can be achieved in the statistical machine translation application.

Table 1.1: The TDT-4 corpus covers news in 3 languages (LDC, 2002)

	Arabic	English	Mandarin
Newswire	An-Nahar	New York Times	Zaobao
	Al-Hayat	Associated Press Wire	Xinhua
	Agence France Press		
Radio	VOA Arabic	PRI The World	VOA Mandarin
		VOA English	CNR
Television	Nile TV	CNN Headline News	CCTV
		ABC World News Tonight	CTS
		NBC Nightly News	CBS-Taiwan
		MSNBC News with Brian Williams	

1.2 Related Work: Topic Detection and Tracking

The topic detection and tracking (TDT) task (Christopher et al., 2000) is a concrete example of a large publicly funded technology demonstration program which motivates the research described in this dissertation. The original TDT corpus contains news broadcasts from 4 audio sources and 2 text sources in English as well as 1 audio source and 2 text sources in Mandarin. The broadcasts were collected concurrently over a 9 month period in 1998. Arabic language sources have since been added to the TDT collection, as indicated in Table 1.1. The goal of the TDT program is to demonstrate a system which can automatically track the reporting of a specified event or set of events across all the news sources, to detect new events as soon as they are reported in any one of the sources, *etc.* The audio sources are transcribed by language-specific ASR systems and the rest of the processing does not explicitly distinguish between speech and text sources. It has been noted in TDT literature that ASR errors, particularly those of named entities and infrequently occurring "content words," degrade fine-grained event

detection and information extraction (Allan et al., 1999).

It has been demonstrated on the other hand that even with mediocre ASR, existing cross-language information retrieval (CLIR) techniques can be effectively employed to identify concurrent documents in the English newswire which are on the same topic as an audio story in a target, resource-deficient language. In this dissertation we study methods to exploit such contemporaneous English documents, with or without MT, to sharpen the language model for each individual audio news story in the target language, an exercise we call *story-specific language modeling*, as illustrated in Figure 1.1. A second-pass transcription of the audio with such sharper models may be employed to improve the ASR accuracy—from being barely adequate for CLIR to possibly being usable for summarization or information extraction.

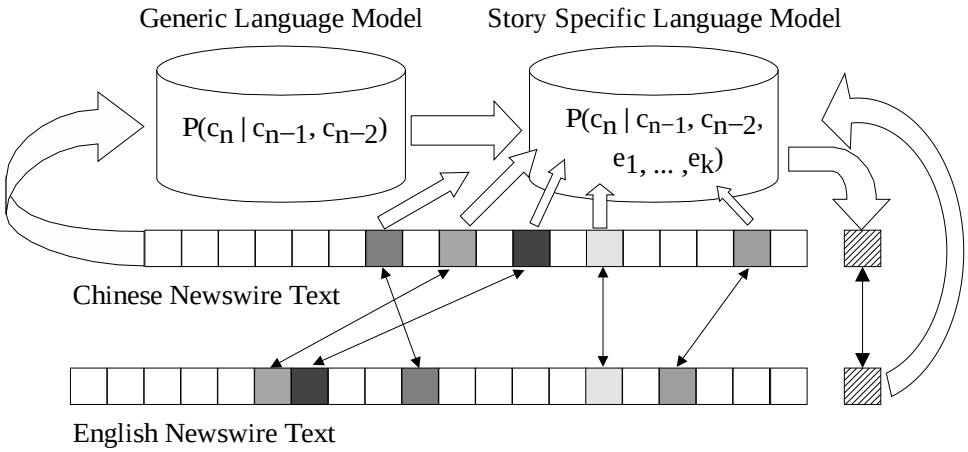

Figure 1.1: Illustration of cross-lingual story-specific language models.

A natural question to ask, in pursuing the strategy depicted in Figure 1.1, is whether sufficient resources for effective CLIR and MT are easier to obtain than additional electronic text for LM training. We argue that, in this setting, the implicit demands on the performance of a CLIR or MT system are far lower than in a setting where CLIR or MT are being evaluated as stand-alone technologies.

For instance, if a CLIR system retrieved only a tenth of the relevant documents in response to a particular query, it may rate poorly (*e.g.* offer modest precision only at unacceptably low recall levels) in a CLIR evaluation. However, if the few documents thus retrieved cover most of the proper names, places and other topic-related entities, which the ASR system would otherwise have difficulties transcribing, then the side-information provided by this CLIR system may be almost as useful as that from a near-perfect CLIR system. Similarly, the word-order of the output of a MT system is of considerable import in its performance evaluation, but may be of little consequence in language modeling: it has been shown by Khudanpur and Wu (1999) that maximum entropy models are as effective at exploiting topic-specific *unigram* frequencies as N-gram models are at exploiting topic-specific *trigram* frequencies. Therefore, the resources needed to put together a CLIR or MT system toward story-specific language modeling may indeed be far lower than for a state-of-the-art CLIR or MT system for stand-alone application. It may, *e.g.*, be possible to obtain a translation lexicon via optical character recognition from a printed bilingual dictionary (cf. Doermann et al., 2002), while additional text in electronic form may simply not be available.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows. We begin in Chapter 2 with the basics of language modeling and continue in Chapter 3 with language model adaptation.

Next, we describe general techniques for exploiting text documents in resource-rich languages for improving a story-specific language model in resource-deficient languages in Chapter 4. Before directly applying our techniques to final applica-

tions, we first carry on preliminary experiments on measuring the perplexity which is a standard metric for a language model in Chapter 5. We then move on to ASR experiments with various situations in Chapter 6. First, we report the results for the case when a fairly large Chinese-English parallel text corpus is available for estimating good statistical models for CLIR and MT, and a reasonable-size language model training text corpus is available. We then study the more compelling situation when little in-domain, in-language text is available for LM training.

One of the demands of the approach is that it requires translation lexicons (dictionaries) for both of CLIR and MT, and such lexicons may not be easily available, especially in resource-deficient languages. In Chapter 7 and Chapter 8, we propose two methods to obtain the translation lexicons automatically from a document-aligned corpus.

Next, we investigate the effectiveness of our approach in a resource-rich language, English, in a different application, Chinese to English statistical machine translation, in Chapter 9.

Finally, we conclude this dissertation in Chapter 10 along with discussions and future directions.

The reader should note that while Chinese plays the role of a resource-deficient language in all our experiments, our techniques are language-independent and applicable to most other languages with little or no modifications. Our primary motivation in choosing Chinese is, indeed, the availability of large Chinese text resources, which enables us to make comparative studies. In practice, these techniques will be of benefit for a language in which LM text is truly not plentiful.

Chapter 2

Basics of Language Modeling

Language modeling is a task assigning a probability to a given sequence of words. It is crucial and indispensable for many speech and natural language applications such as automatic speech recognition (ASR), statistical machine translation (SMT) and optical character recognition (OCR), and it helps the applications especially in two ways. First, it reduces the search space of the problems. Most speech and natural language processing problems can be regarded as finding the most likely answer (word strings) given an input data (test set), and the main difficulty lies in that innumerable candidate answers are possible—which implies that it is impossible to search all possible candidate answers—in most practical problems. Provided by some probabilities estimated from a language model (LM), the search space can be effectively reduced by ignoring unlikely candidates, and thus the search problem becomes feasible. Second, an LM actually improves an application's performance by providing contextual information. In OCR for instance, it is sometimes difficult to distinguish the letter `'l'` and the number `'1'`. In ASR also, it is practically impossible to distinguish cent, sent, and scent from one another unless some contextual information is given, even for humans—

these are, in fact, *homonyms*, meaning that their pronunciations are exactly same. Given some contextual information provided by an LM, however, it becomes much clear to distinguish them most of the time.

2.1 N-gram Language Models

To illustrate how the LM assigns probability to given input word strings, we begin with the most popular LM, N-gram LM. Although the N-gram LM can be applied many applications, here we take an example of the N-gram LM applied to the ASR problem (Jelinek, 1997).

The ASR problem is to find the most likely word string W from the given acoustic evidence (input data) A as

$$\hat{W} \quad = \quad \arg\max_{W} P(W|A) \quad . \tag{2.1}$$

By applying Bayes' formula of probability theory, equation (2.1) can be rewritten as

$$P(W|A) = \frac{P(W)P(A|W)}{P(A)}, \tag{2.2}$$

and since A is fixed—acoustic evidence is already given and doesn't change over the recognition process—our ASR problem can be decomposed into two parts, the acoustic modeling problem, and the language modeling problem.

$$\hat{W} \quad = \quad \arg\max_{W} \quad \underbrace{P(W)}_{\text{Language Model}} \quad \underbrace{P(A|W)}_{\text{Acoustic Model}} \quad . \tag{2.3}$$

In this dissertation, we only address the language modeling problem, $P(W)$.

For given a sequence of word string, $W = w_1, w_2, \cdots, w_n$, the LM estimation using the chain rule and order-2 Markov assumption leads to

$$\begin{aligned} P(w_1, \cdots, w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\cdots P(w_n|w_1, \cdots, w_{n-1}) \quad (2.4) \\ &\approx P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\cdots P(w_n|w_{n-2}, w_{n-1}) \quad (2.5) \\ &= P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P(w_i|w_{i-2}, w_{i-1}) \ . \quad (2.6) \end{aligned}$$

In the last equation (2.6), the first two terms are called a *unigram* and a *bigram*, respectively, and the last term is called a *trigram* because three words are used, w_i, w_{i-2}, and w_{i-1}. Since unigram, bigram and trigram models are same except for how many previous words are used for conditioning, they are called *N-grams*, which are the bases for N-gram language models[1]. Simple maximum likelihood estimates from the training set are used

$$\begin{aligned} P(w_i|w_{i-2}, w_{i-1}) &= \frac{N(w_{i-2}, w_{i-1}, w_i)}{N(w_{i-2}, w_{i-1})}, \\ P(w_i|w_{i-1}) &= \frac{N(w_{i-1}, w_i)}{N(w_{i-1})}, \\ P(w_i) &= \frac{N(w_i)}{\sum_{w_i \in V} N(w_i)}, \end{aligned} \quad (2.7)$$

where $N(w_{i-2}, w_{i-1}, w_i)$ denotes the number of times entry (w_{i-2}, w_{i-1}, w_i) appears in the training data and V is the vocabulary.

[1]Of course, higher order N-grams are possible such as 4-grams, 5-grams, etc.

2.2 Language Model Smoothing

N-gram LMs are easy to build as all we need is simple counts of the word N-gram events from the training set. But a problem arises when a word N-gram is encountered during the model use (testing) that was not seen in the training set. In other words, $N(w_{i-2}, w_{i-1}, w_i)$ may be zero in the training set, whereas it is non-zero in the test set. That is, we need to estimate the probability of events *unseen* in the training set. As maximum likelihood estimates are based on the observations during the training, according to equation (2.7), such a model will give 0 probability for such word N-grams. Furthermore, it will give the probability of the entire sentence to be 0 in the left hand side of equation (2.4) meaning that the sentence is not possible at all. Therefore, every sentence which contains N-grams which are not seen in the training set will be impossible. Due to data sparseness, this happens frequently in reality, and it is fatal in many practical applications. The act of modifying the model of equation (2.7) so that no word sequence gets zero probability is called *smoothing* (Jelinek, 1997).

Several smoothing techniques have been proposed, and there are several which work fairly well for ASR or other applications. The fundamental idea of smoothing techniques is to reserve (subtract) some small probability mass from the relative frequency estimates (2.7) of the probabilities of seen events, and to redistribute this probability to unseen events. Smoothing methods differ according to how much is subtracted out (discounting) and how it is redistributed (back-off). Here, we briefly review the state-of-the-art smoothing method, Kneser-Ney methods which was first introduce by Kneser and Ney (1995), and more details may be found in Chen and Goodman (1998); Kim et al. (2001).

The Kneser-Ney Methods: Upon encountering unseen events, a detailed

model backs-off to less specific models for probability calculation because less specific ones have smaller parameter spaces, and thus are less likely to have zero probability events than more specific models. E.g., N-gram models back-off to (N-1)-gram models. In the Kneser-Ney model, one backs-off to a probability distribution other than an (N-1)-gram probability. If one uses absolute discounting to compute the reserved probability, the Kneser-Ney modified back-off model may be written as

$$P(w|u,v) = \begin{cases} \frac{N(u,v,w)-d}{N(u,v)} & \text{if } N(u,v,w) > 0 \\ \underbrace{\frac{d \cdot n_{>0}(u,v,\cdot)}{N(u,v)}}_{\text{discounted mass}} \cdot \beta(w|u,v) & \text{otherwise,} \end{cases} \tag{2.8}$$

where d is a small constant, and the new back-off distribution is

$$\beta(w|u,v) = \frac{n_{>0}(\cdot,v,w)}{\sum_{\tilde{w}:N(u,v,\tilde{w})=0} n_{>0}(\cdot,v,\tilde{w})}, \tag{2.9}$$

where $n_{>0}(\cdot,v,w) = \sum_{\tilde{u}:N(\tilde{u},v,w)>0} 1$ and $n_{>0}(u,v,\cdot) = \sum_{\tilde{w}:N(u,v,\tilde{w})>0} 1$. As noted in (2.8), the back-off occurs only if $N(u,v,w) = 0$, and the backed-off probability is the product of the total probability discounted from the relative frequency estimates and a new back-off distribution β. The distribution β is based on the number of unique words (or types) w which occur in a lower-order context, while the conventional (N-1)-gram is based on the actual counts (or tokens) of w in the lower order context.

The new back-off distribution β may also be applied in a non-linear interpolation (NI) model. *i.e.*

$$P(w|u,v) = \frac{\max\{N(u,v,w) - d, 0\}}{N(u,v)} + \quad \frac{n_{>0}(u,v,\cdot) \cdot d}{N(u,v)} \cdot \hat{\beta}(w|u,v) \tag{2.10}$$

where, $\hat{\beta}(w|u,v) = \frac{n_{>0}(\cdot,v,w)}{\sum_{\tilde{w}} n_{>0}(\cdot,v,\tilde{w})}$.

Note the difference between β and $\hat{\beta}$: interpolation ($\hat{\beta}$) applies to every w in the vocabulary, while back-off (β) applies only to words w that were not seen in the context (u,v) in training.

2.3 Evaluation of Language Models

Another important issue in language modeling is how to evaluate an LM. The most intuitive direct measure for the ASR task is the word error rate (WER)[2]: the measure how accurately an ASR system recognizes speech utterances. Simply, the WER is the ratio of the number of incorrectly recognized words (errors) in the ASR output (hypothesis) to the total number of words in the reference (correct answer).

Here is an example showing a reference, a hypothesis, and each word's error types (Chelba, 2000).

```
REF:    UP UPSTATE NEW YORK SOMEWHERE UH     OVER OVER HUGE AREAS
HYP:       UPSTATE NEW YORK SOMEWHERE UH ALL ALL  THE  HUGE AREAS
ERR:    D                                I   S    S
```

where `D, I,` and `S` stand for errors due to deletions, insertions and substitutions, respectively.

Given a reference transcription and a recognized hypothesis, we can define three kinds errors (mismatches): deletions, insertions, and substitutions. The WER is then defined as follows,

$$\text{WER} = \frac{\text{Deletions} + \text{Insertions} + \text{Substitutions}}{\text{Total No. of Words in Reference}} \times 100\,. \tag{2.11}$$

[2]More accurately, the WER is a measure for an ASR system, nevertheless, it is often used for an LM because ASR system's performance depends on the LM.

In the example above, the WER is

$$\text{WER} = \frac{1+1+2}{10} \times 100 = 40\%. \tag{2.12}$$

It is, however, computationally expensive to run the ASR to measure the performance of an LM whenever we build a new LM, as the ASR system is comprised of many components in addition to the LM. It is therefore preferable to measure the quality of the LM without actually using the ASR system.

As an alternative to the WER, a much simpler measure called perplexity (PPL) has been introduced and widely used (Jelinek, 1997). It is based on the geometric mean of the test set probabilities and measures how well an LM, M, predicts the test data of size k. Of course, the test data has neither been used nor seen at training (building) the LM.

$$\text{PPL}(M) = \exp\left(-\frac{1}{k}\sum_{i=1}^{k} \ln P_M(w_i|w_{i-2}, w_{i-1})\right) \quad . \tag{2.13}$$

The perplexity is often interpreted as the branching factor: the average number of word choices for each word position. For instance, if an LM gives the perplexity of 1,000, it means that there are 1,000 possible choices on average for every word prediction position. Naturally therefore, a language model which gives the lower perplexity is regarded as a better model.

There is one problem, though, with the perplexity. While the estimation of the perplexity can be easily done—as it does not require any acoustic model in the ASR problem—it does not take into account the acoustic countability at all. Suppose there are two words clearly different in terms of acoustics, say `A` and `B`. And it is clear that word `A` is spoken according to the acoustic model score. It is possible that word `B` has the higher score in terms of the language model

(lower perplexity) score than word `A`, and therefore the language model tries to boost the score of word `B`. Consequently, word `B` may be selected as the finally recognized word even though word `A` has been actually spoken. Of course, this is one extreme and one says nothing about the truth; however, notice that there have been numerous reports which show worse ASR WER results even though the LM has the lower (better) perplexity. Evidently, this suggests that lowering the perplexity of an LM does *not always* result in improvements in the ASR performance. Typically, we use the perplexity criterion to find or build the better language model, and confirm it in terms of the WER in the end.

2.4 Significance Test

The quality of an LM, as mentioned above, can be measured by the perplexity, and it is closely related to the performance in one of our main application: e.g., WER in ASR. In principle, a better LM gives a lower (better) WER; it is, however, possible that there is not a substantial WER difference between the recognition performance under a new LM when compared to a baseline LM. In particular, we would like to ensure that the performance improvement is not caused by chance. For this purpose, we will use a statistical significance test (Rice, 1995).

A statistical significance test provides a mechanism for making quantitative decisions about a process or processes. The intent is to determine whether there is enough evidence to *reject* a conjecture or hypothesis about the process. This hypothesis that we eventually want to reject is called the *null hypothesis* (H_0). There is another hypothesis, *alternative hypothesis* (H_a), which is complementary to the null hypothesis; therefore, only one of the two hypotheses can be true or *accepted*. For example, if our goal is to see if a new language model performs

significantly better than a baseline language model, the null hypothesis would be both language model performs similar—as we want to find a strong clue which says two language models perform differently.

The test procedure is then constructed so that the risk or the probability of rejecting the null hypothesis, when it is in fact true, is small. This risk, α, is often referred to as the *significance level*[3] of the test. By designing a test with a small value[4] of α, we ensure that rejecting the null hypothesis is meaningful. Next, based on a sample statistic, we estimate the probability of obtaining the test statistic greater than or equal to the observed sample statistic under the null hypothesis: p-value. Finally, if the probability, p-value, is smaller than the significance level α, it means that it is extremely unlikely that the sample statistic happens under the null hypothesis, and thus, we reject the null hypothesis; otherwise, as there is no strong evidence that it does not happen, we accept the null hypothesis.

Here, we give an example of the significance test: a matched pairs sentence-segment word error (MAPSSWE) test, which can be performed by a NIST (National Institute of Standards and Technology) ASR evaluation tool (Pallett et al., 1990). Since we are interested in whether one system performs significantly better than the other, our hypotheses would be given by:

H_0 : the mean of error differences between two systems is zero,

H_a : the mean of error differences between two systems is not zero.

The practical meaning of the null hypothesis is that two systems are basically same or not significantly different.

[3]A statistical significance test is often referred to a hypothesis test. This is because the selection and use of the significance level is fundamental to the hypothesis testing procedure.

[4]Typical α values used for significance tests are 0.05 or 0.01.

One thing to note is that this is a *matched pairs test*: when we run the test, the sample observations come from paired samples according to some characteristics of samples. To illustrate the matched pairs test, suppose we want to test the hypotheses above, with system A and system B, and there is no pairing in our samples. That is, we collect the system A's samples and system B's samples from a pool of samples. Let's call system A's samples and system B's samples as SA and SB respectively. It is possible then that the sample SA turns out to be much more difficult to be correctly recognized than the sample SB since the samples have been selected by random. If this happens by any chance, we will probably end up with concluding that system A performs significantly worse than system B—even if both system performs similarly, in fact. That happened solely because the difficulty of the two test sets is completely ignored in the sample selection process. In order to make a fair comparison, we need to divide the samples equally in terms of the difficulty level for system A and system B. One way to achieve this is to first pair the whole test set according to the difficulty of each test set sentence, and then assign one sample to one system and the other sample to the other system for each pair.

In ASR tests, since we can use the same test set twice—once for system A's outputs, and once for system B's outputs—we test both systems with the exactly same test set. To proceed, we count the numbers of errors in sentences that are specific to the output of the two systems being compared.

$$X_i = N_A^i - N_B^i\,, \qquad i = 1, 2, \cdots, n\,, \tag{2.14}$$

where N_A^i and N_B^i are the numbers of errors in the i^{th} sentence from system A and B, respectively, and n is the total number of sentences. Assuming that the

number of sentences is large enough, the central limit theorem (Rice, 1995) allows us to make an assumption that the numbers of errors are *normally distributed*. As the variance (σ) is unknown, we can do the t-test (Rice, 1995) for estimating the mean difference of normal distributions. Let

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n} \tag{2.15}$$

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2 \tag{2.16}$$

and let our t-test statistic to be

$$t = \frac{\bar{X}}{(S/\sqrt{n})} \,. \tag{2.17}$$

The null hypothesis asserts that the distribution of error differences has mean zero (two-tailed) or has mean no larger than zero (one-tailed, with System B a possible improvement on System A). The null hypothesis is then rejected if the measured value t of T is such that

$$P(T \geq t) \leq \alpha \quad \text{(one-tailed), or} \tag{2.18}$$

$$P(|T| \geq |t|) = 2P(T \geq |t|) \leq \alpha \quad \text{(two-tailed).} \tag{2.19}$$

Figure 2.1 shows an example of one tailed t-curve when the t-value estimated from a sample is 1.5. The colored area corresponds to $P(T \geq 1.5)$ and suppose that the probability (p-value) is 0.0668. The practical interpretation of this is that under the assumption the null hypothesis—two systems perform similarly—is true, the probability of the sample statistic t is 1.5 or bigger is 0.0668. Indeed, it is unlikely to happen, but the probability under the null hypothesis is greater

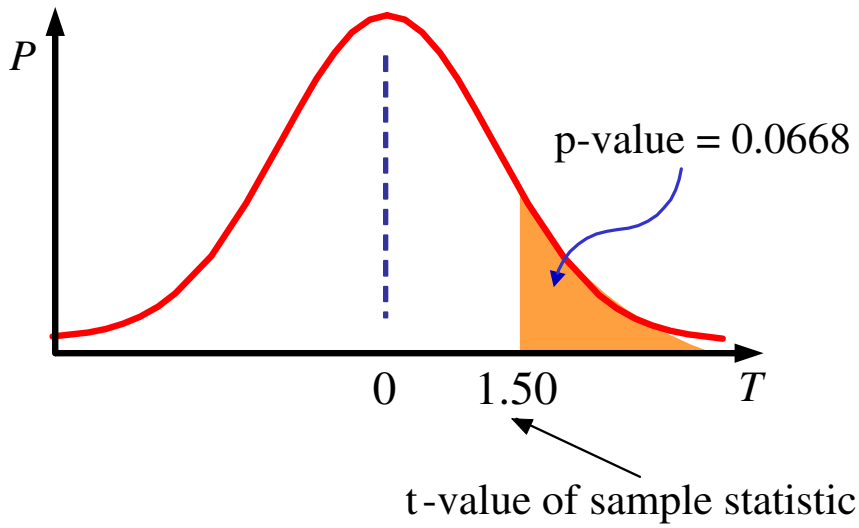

Figure 2.1: An example of the significance test (one tailed t-test)

than α (which is typically 0.05). Hence, there is no strong evidence to reject the null hypothesis and we accept the null hypothesis—the two systems are not significantly different. On the other hand, if the p-value is smaller than α, we would have strong evidence to reject the null hypothesis and say that one system is significantly better than the other.

Chapter 3

Language Model Adaptation

3.1 Motivation

N-gram LMs have been popular because their estimates are reliable and robust as well as simple to measure. Based on the Markov assumption—which regards all history contexts ending with same word sequences as equivalent—the N-gram LM builds equivalent history classes by simply looking at the previous (N-1) words, and it provides robust estimates for the next possible words even if the events are not seen during training. For example, the 4-gram phrase fifty miles per hour can belong to the same equivalence class as the phrase sixty miles per hour or any other 4-gram phrases ending with miles per hour. In other words, as long as there is a phrase ending with miles per hour in the training set, the trigram LM can give us an estimate of $P(\text{hour}|\text{fifty miles per})$ even if it has not been seen in the training set.

There are some cases, however, where N-gram LMs are not good enough to predict the next word. Suppose we have the following sentence (taken from the Switchboard corpus (Godfrey et al., 1992)).

You know I want to throw some charcoal on the grill and and throw a steak on there and some baked potatoes and stuff like that.

Suppose we are predicting the word baked. We note that the N-gram LM predicts this word based on the previous two words—and some. As almost any adjective or noun can follow this context, most of the words—including the word baked—which can follow this context will have very small probabilities. In other words, it is extremely difficult to correctly predict the word baked given the context of and some. However, if we take into account longer contexts such as the whole sentence, it becomes clear that the sentence is about outdoor cooking. Consequently, if we take the topic into account, the probability of the word baked will be much higher than if the topic is ignored, i.e.,

$$P_{general}(\text{baked}|\text{and some}) \ll P_{cooking}(\text{baked}|\text{and some}) \quad \text{or} \quad (3.1)$$

$$P_{general}(\text{baked}|\text{and some}) \ll P_{history \ni \{charcoal, grill, steak\}}(\text{baked}|\text{and some})\,. \quad (3.2)$$

It is intuitive that the N-gram LM is good at predicting function words (e.g., I, to, the, of, etc.) because the usage of function words is clearly constrained by our natural language grammar and conversely, it can be easily anticipated by simply looking at a few immediate previous words. On the other hand, the N-gram LM is not so certain about the prediction of content words especially following the function words. Indeed, it is not trivial to predict content words even for humans unless we are given a strong clue.

The obvious question is then to determine the factors in determining the content words in the natural language. It is not difficult to see that the usage of content words is usually limited by some topic, genre, domain, or style. For ex-

ample, if a document is about outdoor cooking, there are certain words which are related to outdoor cooking—such as charcoal, grill, steak, baked, etc.—and these words would appear more frequently here than in other documents. In other words, the probability of outdoor cooking-related words should be boosted only when the document is about outdoor cooking.

Another problem in the N-gram language model is that N-grams are *static*. That is, regardless of topic, genre, domain, or style, the probability of events (N-gram word tuples) remains identical as long as the N-gram tuples are same. However, this is not the case: thinking of actual texts or conversations between humans, the topic, subject, style, etc. changes often, and the word or phrase usage changes accordingly. Especially if an N-gram word tuple contains any content word, the usage of the N-gram word tuple is different according the topic, genre, domain or style, and therefore it should be changed *dynamically*. The usage of function words, on the other hand, usually remains consistent regardless of any document, topic, domain or genre. Since these static features are common to all texts in the training set, they are salient in terms of N-gram counts and therefore the static features are easily captured by N-gram models. However, dynamic features appear only under some conditions of the training set leading their overall N-gram counts not being salient.

LM adaptation is an approach to cope with the difficulty of predicting content words. It is also an attempt to capture long distance dependencies. As shown in the outdoor cooking example, the dependencies between the cue words such as charcoal, grill, steak and the predicted word baked are completely ignored in N-gram language modeling. This has been shown to be effective when the main task comprises many topics or documents such as broadcast news transcriptions (Chen et al., 2003; Federico, 1999). The basic idea is to first cluster the whole training

data according to dynamic features such as topics, build separate topic-specific language models for each cluster, select the most similar cluster for a given test data, and finally adapt it. LM adaptation differs in two ways: how to build or derive adaptive LMs and how to combine adaptive LMs with the static LM. For building dynamic LMs, there are cache-based approaches (Kuhn and Mori, 1990), trigger-based approaches which we will discuss in Chapter 7, and topic-based approaches which are going to be described in Section 3.2. For combining dynamic LMs with the static LM, one obvious way is to build mixture LMs using interpolation techniques (Gotoh and Renals, 1999; Iyer and Ostendorf, 1999). In addition, minimum discrimination information based models (Federico and Bertoldi, 2001) or maximum entropy models have been used (Khudanpur and Wu, 1999; Rosenfeld, 1996).

3.2 Topic Based Language Model

The topic based language model is a representative example of LM adaptation: the adaptation by topic (Chen et al., 1998; Florian and Yarowsky, 1999; Gotoh and Renals, 1999; Iyer and Ostendorf, 1999; Seymore and Rosenfeld, 1997). It has been popular and proven to be effective because the adaptation unit, topic, is specific enough to capture the dynamic feature of a document and, on the other hand, general enough so as not to *severely* suffer from data fragmentation. The main idea of topic based language modeling is based on the assumption that the usage of content words typically[1] depends on the topic of the text.

In our example above, words frequently used for outdoor cooking would be top-

[1]Of course there are other factors that affect the usage of the words in a text such as style, genre, domain, etc. and we can construct many adaptive LMs such as style based LMs, genre based LMs, etc.

Word	Freq in Clothing	Freq in whole Corpus	log difference
APPEARANCE	4.5181e-04	1.5789e-05	1.4566
ATTIRE	4.5181e-04	3.9471e-06	2.0586
ATTORNEYS	6.0241e-04	2.7630e-05	1.3385
AVON	3.0121e-04	7.8943e-06	1.5815
BACKLESS	3.0121e-04	2.6314e-06	2.0587
BAKERY	6.0241e-04	1.0526e-05	1.7576
BLOUSE	6.0241e-04	1.0526e-05	1.7576
BLOUSES	9.0361e-04	1.0526e-05	1.9337
BOOTS	9.0361e-04	1.9736e-05	1.6607
BAGGY	3.0121e-04	3.9471e-06	1.8847

Table 3.1: Sensitive words for the topic "CLOTHES" (Wu, 2002)

ically similar words which means that most of them are about outdoor cooking. If a test document uses the outdoor cooking-related words frequently, chances are that the test document is about outdoor cooking, and therefore, the probability of all those outdoor cooking-related words should be boosted. Table 3.1 shows a more concrete example from the Switchboard corpus (Godfrey et al., 1992). It is a conversational speech corpus on 70 previously determined topics such as education, sports, automobile. In particular, Table 3.1 compares the usage (unigram frequency) of some topic-sensitive words—where the topic is "CLOTHES"—when the topic of the document is "CLOTHES" and the usage in the whole training corpus. Evidently, those topic-sensitive words appear more frequently when the document is *in topic* than the general case.

The natural question then is how to identify the topic of a document. This is closely related to the IR problem which retrieves the similar documents to a given query; in fact, many of topic based language modeling approaches use simple IR techniques such as vector space models. In other words, they first extract a *pseudo* document from the test document, build a bag of words[2] from

[2]A bag of words can be regarded as a set of words with counts for each word. It is simply

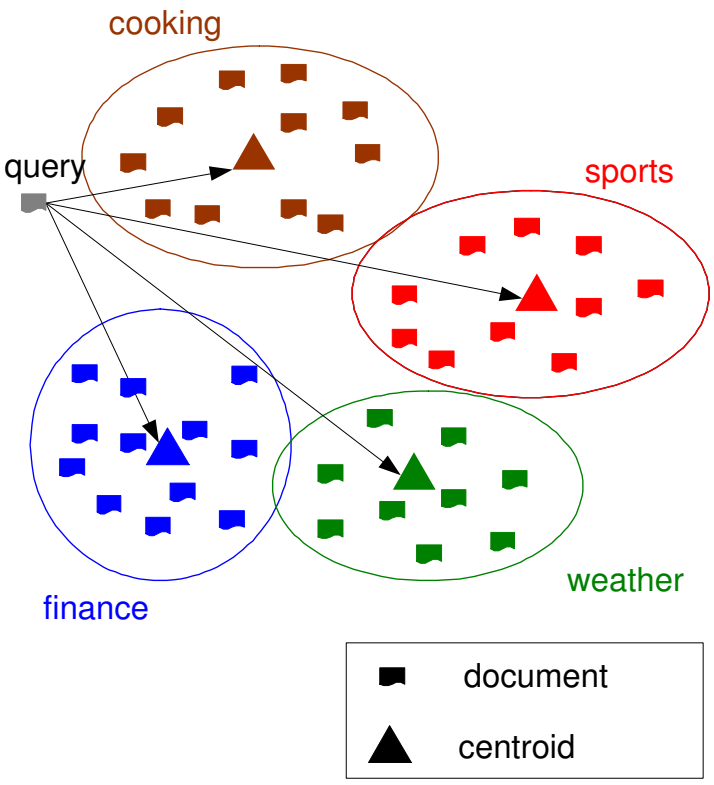

Figure 3.1: Topic based language modeling as vector space IR

the document, use the bag of words as a query, and find the closest topic based on some similarity criterion such as the cosine similarity. Figure 3.1 shows an example of topic based language modeling. Out of pre-defined topic clusters, the most similar topic cluster to the given query document is identified, and then the corresponding topic LM is used for adaptation. One noticeable different approach is Bellegarda (1998) which is based on latent semantic analysis.

Also, the topic itself could be a mixture of many topics—as opposed to be a single topic—as the topic identification is based on a stochastic decision (Gildea and Hofmann, 1999). In that case, the topic specific LM probabilities are repre-

constructed by taking each word out of the document sequentially, then putting the word into a bag with recording each word count. Therefore, word order is completely ignored.

sented as

$$p(w|h) = \sum_t p(w|t)p(t|h) \ , \tag{3.3}$$

where t is a latent class variable for different topics, h is a history equivalence class. Also, $p(w|t)$ distributions are topic-specific unigram word probabilities and $p(t|h)$ are mixing weights for each topic. Notice that Eq. (3.3) does not capture the local structure of the language as it is a unigram probability, and therefore it must be combined with the standard N-gram LM in some way (Martin et al., 1997).

The remaining question is how to classify training documents to build topic clusters. This can also be done by IR techniques and some automatic clustering technique such as K-means clustering (Duda and Hart, 1974). In short, an initial topic (class) out of previously determined K topics is assigned to each training document. Then, based on that initial topic assignment, we can build the topic centroid for each topic. In the next iteration, for each document, we can find the most similar topic based on the distance to each topic centroid and assign the topic to the document. Based on this topic assignment, each topic centroid is updated, and this iteration continues until topic centroids converge. More details about the topic clustering in language modeling can be found in Wu (2002, chap. 4). Another variation is to use manual topic assignment if it is available. It may be more accurate than the method based on the automatic clustering; nonetheless, the latter is preferred as it does not require any manual assignment.

In summary, topic based language modeling is a language model adaptation technique where the adaptation unit is the topic. Starting from some initial topic assignment, we can automatically induce K topic clusters from the training set documents. From the K topic clusters, we build topic-specific language models.

Given a test set document, then, we identify the most similar topic using some IR techniques. Finally, we adapt the corresponding topic-specific language model with the static or background N-gram language model.

3.3 Adaptation Method: Maximum Entropy Model

There has been considerable research for combining topic related information with N-gram models (Bellegarda, 1998; Clarkson and Robinson, 1997; Iyer and Ostendorf, 1999; Kneser et al., 1997). The basic idea of these approaches is to exploit the differences of word N-gram distributions across topics. That is, first the whole training data is separated into several topic-specific clusters, and then topic-specific LMs are built using the topic-specific data. One problem of this approach is data fragmentation, which results in the data sparseness problem. In order to remedy the data sparseness problem, linear interpolation (or LM mixture) has been applied.

Rather than using the linear interpolation, the maximum entropy (ME) model provides an elegant way to combine adaptive language models with the static[3] language model. Unlike linear interpolation, maximum entropy does not need the component language models. Instead, it requires knowledge or information from various sources which is represented by some *constraints* or *features* $f_i(h, w)$ which are typically, but not necessarily binary functions: i.e. given an event (h, w), it returns either 0 or 1. Given a sample event space, these constraints, in fact,

[3]Often, this language model is called *background language model* as opposed to adaptive language models in some references.

specify a subset of the sample space,

$$E_{p(h,w)}[f_i(h,w)] = \sum_{h,w} p(h,w) f_i(h,w) = K_i \tag{3.4}$$

where h is the word history, w is the word to be predicted and $E_{p(h,w)}$ denotes the expectation over the probability distribution $p(h,w)$. The most common choice for K_i are the empirical counts of the constraint $f_i(h,w)$. Our goal is then to find the probability distribution

$$p^*(h,w) = \arg \max_{p(h,w)\in\mathcal{P}} - \sum_{h,w} p(h,w) \log p(h,w)\,, \tag{3.5}$$

which maximizes the entropy (Cover and Thomas, 1991) where $\mathcal{P}$ is a linear family of probability distributions. The basic goal of the maximum entropy model is to find the solution for equation (3.5). It can be interpreted as finding the distribution which satisfies all the constraints, but no other assumption is made. Therefore, in view of the language model smoothing, it is same as finding the smoothest language model (distribution) among all distributions which satisfy the constraints.

Under the assumption that the the constraints are consistent—which means there is at least one probability distribution which satisfies all constraints—the solution of equation (3.5) is unique and has an exponential form:

$$p^*(h,w) = \frac{1}{z} \exp\left(\sum_{i=1}^{K} \lambda_i f_i(h,w)\right) \tag{3.6}$$

where z is the normalization term. There is no closed form solution for equation (3.6); however, there are some iterative procedures which starts with some arbitrary distribution and converges to the solution such as generalized iterative

scaling (GIS) (Csiszár, 1989; Darroch and Ratcliff, 1972) or improved iterative scaling (IIS) (Pietra et al., 1997).

Here we show the example of combining topic-based LMs using the maximum entropy model (Wu, 2002). They use the topic-dependent trigram as the sufficient statistic of history.

$$p(w_i|w_1, \cdots, w_{i-1}) \approx p(w_i|w_{i-1}, w_{i-2}, t_i) \tag{3.7}$$

where t_i stands for an i-th topic. Then they seek a model which, in addition to topic-independent N-gram constraints,

$$\sum_{t_i} p(w_i|w_{i-1}, w_{i-2}, t_i)p(w_{i-2}, w_{i-1}, t_i) = \frac{\#[w_{i-2}, w_{i-1}, w_i]}{\#[\text{training data}]}, \tag{3.8}$$

$$\sum_{t_i, w_{i-2}} p(w_i|w_{i-1}, w_{i-2}, t_i)p(w_{i-2}, w_{i-1}, t_i) = \frac{\#[w_{i-1}, w_i]}{\#[\text{training data}]}, \tag{3.9}$$

$$\sum_{t_i, w_{i-2}, w_{i-1}} p(w_i|w_{i-1}, w_{i-2}, t_i)p(w_{i-2}, w_{i-1}, t_i) = \frac{\#[w_i]}{\#[\text{training data}]}, \tag{3.10}$$

where $\#[\,]$ denotes the count operator, meets topic-dependent *marginal* constraints

$$\sum_{w_{i-1}, w_{i-2}} p(w_i|w_{i-1}, w_{i-2}, t_i)p(w_{i-1}, w_{i-2}, t_i) = \frac{\#[t_i, w_i]}{\#[\text{training data}]}. \tag{3.11}$$

We have seen that smoothing is an important issue in N-gram language modeling; it is important in maximum entropy modeling as well (Chen and Rosenfeld, 1999; Martin et al., 2000). Constraining unreliable marginal probabilities—those observed only once (*singletons*) or twice (*doubletons*)—increases the computational overhead, and therefore, those constraints are ignored. The key to success of maximum entropy language modeling lies in deriving which constraints should be imposed and how to impose them. One easiest way to achieve this goal is

to start from the important features and gradually extend the features (Pietra et al., 1997). In N-gram based maximum entropy language modeling, it has been shown that unreliable constraints such as singleton trigrams should be completely ignored from the model's constraints. Furthermore, some discounting methods such as Good-Turing discounts (Good, 1953) may be applied to the relative frequency counts of the marginal probabilities on the right-hand sides of equations (3.8)-(3.11). Finally, the ME solution has an exponential form

$$p(w_i|w_{i-1}, w_{i-2}, t_i) = \frac{e^{\lambda(w_i)} \cdot e^{\lambda(w_{i-1},w_i)} \cdot e^{\lambda(w_{i-2},w_{i-1},w_i)} \cdot e^{\lambda(t_i,w_i)}}{z(w_{i-1}, w_{i-2}, t_i)} \tag{3.12}$$

where $z(w_{i-1}, w_{i-2}, t_i)$ is a normalization constant. The first three numerator terms correspond to standard N-gram constraints, whereas the last one is a topic-unigram parameter in a particular topic.

Chapter 4

Cross-Lingual Story-Specific Language Modeling

In this chapter, we tackle the problem of the lack of enough training data in the language modeling task. To be more concrete, suppose we are building an ASR system for some language where there is little language model training text. Since our language of interest is resource-deficient, chances are that extensive amounts of LM training data would not be available. In other words, the LM training data would be too small to build a reliable language model, and we are not able to build an accurate language model. However, there are some resource-rich languages where extensive amounts of LM training texts are easily available such as English, French or German. Therefore, it would be beneficial if there were a way to use texts in resource-rich languages to build a better language model in a resource-deficient language. To this end, we try to take advantage of resource-rich languages, and try to build better language models in the resource-deficient language via language model adaptation techniques (Khudanpur and Kim, 2004; Kim and Khudanpur, 2003b).

Our aim is to sharpen (adapt) a language model in a resource-deficient language, say, Mandarin Chinese, by using data from a resource-rich language, say, English. Mandarin Chinese is, of course, no longer resource-deficient for language modeling—hundreds of millions of words are available on-line. However, we have chosen it for our experiments partly because it is sufficiently different from English to pose a real challenge, and because the availability of large text corpora in fact permits us to simulate controlled resource deficiency. Of course, any other pair of languages will serve the purpose of this exposition. Besides, our main task of interest is the broadcast news transcription. Again, there is nothing especially designed or tuned for the broadcast news transcription task; our methods may be easily applied to any other task without any significant modification.

4.1 Overview

For the sake of illustration, let $d_1^C, \ldots, d_N^C$ denote the text of N *test stories* in Mandarin news broadcast to be transcribed by an ASR system, and let $d_1^E, \ldots, d_N^E$ denote their corresponding or *aligned* English newswire articles, selected from some *contemporaneous* text corpus. Correspondence (or alignment) here does not necessarily imply that the English documents d_i^E needs to be an exact translation of the Mandarin story d_i^C. It is quite adequate, for instance, if the two stories report the same news event. Our approach is expected to be helpful even when the English document is merely on the same general topic as the Mandarin story, although the closer the content of a pair of articles the better the proposed methods are likely to work. Finding such document correspondence is a well-known problem in the IR community, and especially when the document collections and query documents are written in different languages, it is called cross-lingual in-

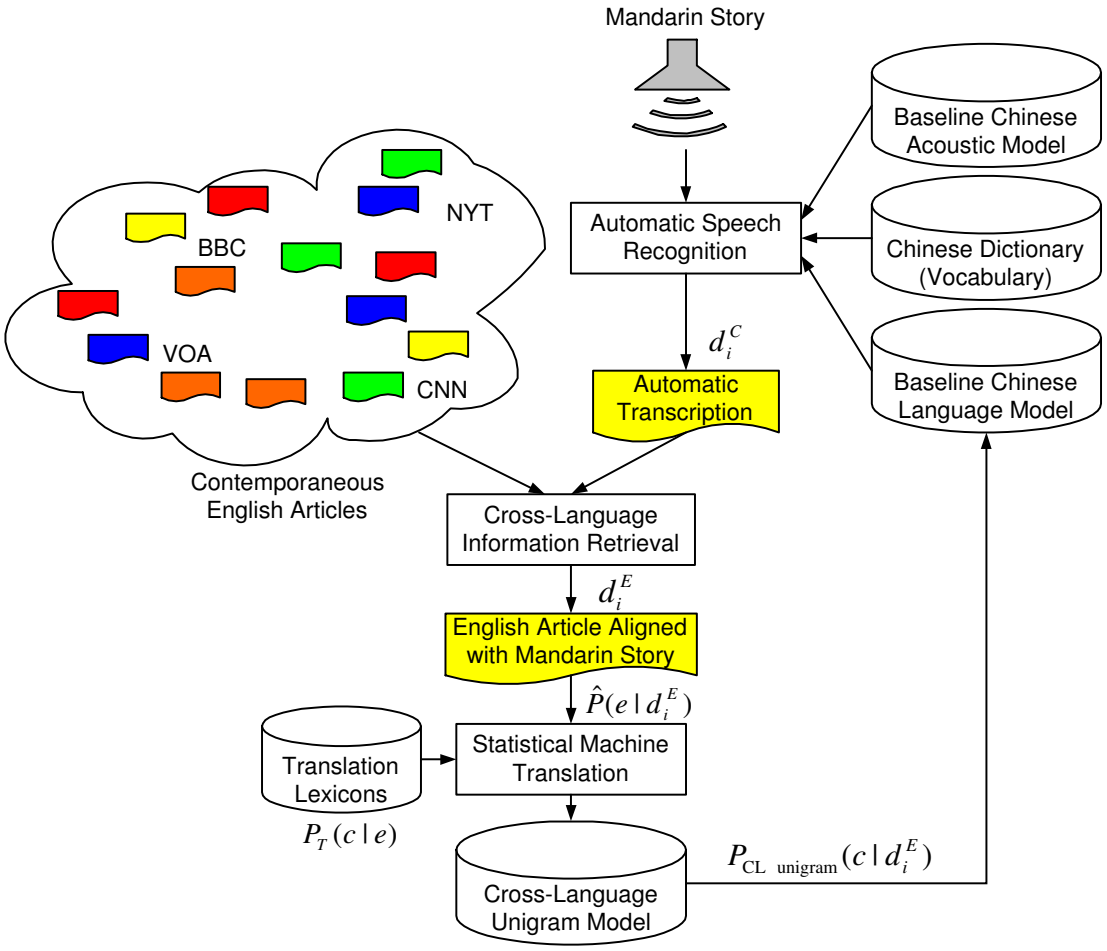

Figure 4.1: Story-specific cross-lingual adaptation of an LM.

formation retrieval (CLIR). Assume for the time being[1] that a sufficiently good Chinese-English story alignment is (somehow) given. Assume further[2] that we have at our disposal a stochastic translation lexicon—a probabilistic model of the form $P_T(c|e)$—which provides the Chinese translation $c \in \mathcal{C}$ of each English word $e \in \mathcal{E}$, where $\mathcal{C}$ and $\mathcal{E}$ respectively denote our Chinese and English vocabularies.

Given the document correspondence and translation lexicons, we are ready to show how to build cross-lingual adaptive language models. Figure 4.1 shows

[1]Details of getting document correspondence will be discussed in Section 4.2.
[2]This will also be described in Section 4.3.

the data flow in this cross-lingual LM adaptation approach. From the first pass output of an ASR system which is an automatic transcription, we build pseudo documents[3] and these documents, in turn, are used as queries for CLIR—to find the most similar or relevant document(s) d_i^E from an English text collection. As English is resource-rich, we assume that there are an extensive amount of English data available. The retrieved English document(s) as a result of CLIR, presumably, would be relevant to the given Chinese query document, d_i^C; even though they are written in English, not in Chinese which is our language of interest. Nonetheless, some useful information or statistics necessary to language model adaption can be effectively extracted and *translated* into Chinese by crude machine translation techniques[4]. Section 4.4 will describe how to translate this English into Chinese.

This translated Chinese unigram distribution, $P_{\text{CL-unigram}}(c_k|d_i^E)$, can be viewed as a story-specific cross-lingual adaptive LM. That is, for each of the Chinese test document, d_i^C, we build the separate LM which is specific to the document. Notice the difference with the topic based LM adaptation mentioned in Section 3.2; unlike topic based LM adaptation that the most similar topic LM is activated as in Figure 3.1, there is no topic cluster in the story-specific language modeling approach. Instead, it retrieves the relevant documents not being limited by topic, and hence, it can effectively identify the relevant documents even when the test document is a composite of multiple topics as in Figure 4.2. Furthermore, our story-specific adaptation unit is finer than the one in topic based language modeling in terms of the granularity. The clustering approach used for topic base

[3]Notice that these documents are different from the reference documents, as they are based on the first pass outputs. However, we believe those are good enough to be used as queries for CLIR to pick out relevant documents.

[4]From the viewpoint of MT, as English should be translated into Chinese, English is the source language and Chinese is the target language.

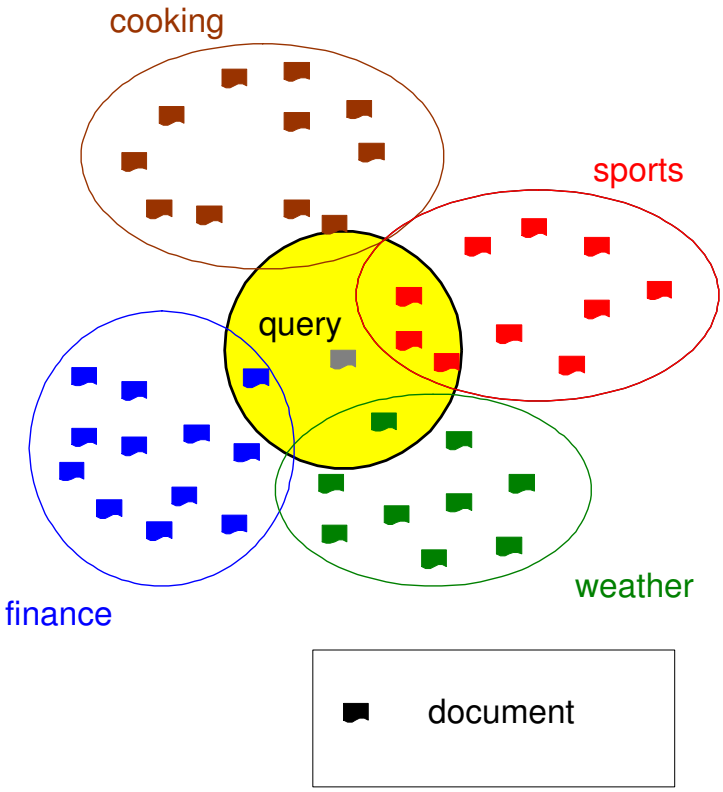

Figure 4.2: Story-specific language model in a vector space

language modeling is inevitably dependent on some factors such as the number of topics and topic distributions over the training set, which should be determined in advance. By constructing adaptive language models for each test document, we try to minimize the unnecessary effect imposed by the training set. Furthermore, we can dynamically adjust the amount of adaptation based on the characteristics of the test document.

Finally the ASR system uses the LM in a second pass decoding. In the following sections, we describe more details of each step of our cross-lingual language model adaptation approach.

4.2 Obtaining the Matching English Document(s) d_i^E

The first assumption made in our approach is that the document correspondence between English and Chinese documents is somehow given. In this section, we describe a method to obtain the correspondence automatically. This is indeed a classical task of CLIR, which selects similar or relevant documents for a given query where the documents and the query are written in different languages and it has been long and widely studied (Davis and Ogden, 1997; Grefenstette and Grefenstette, 1998; Oard, 1997). In our approach, we don't necessarily use the state-of-the-art IR method; rather, we use a simple and crude IR method, vector space model, and we try to show that our cross-lingual language model adaptation is effective even with the crude IR method.

To illustrate how one may obtain the English document(s) d_i^E to match the Mandarin story d_i^C, let us assume that we also have a stochastic reverse-translation lexicon $P_T(e|c)$ which is a Chinese to English dictionary. One obtains from the first pass ASR output, cf. Figure 4.1, the relative frequency estimate $\hat{P}(c|d_i^C)$ of Chinese words c in d_i^C, $c \in \mathcal{C}$, and uses the translation lexicon $P_T(e|c)$ to compute

$$P_{\text{CL-unigram}}(e|d_i^C) = \sum_{c \in \mathcal{C}} P_T(e|c)\hat{P}(c|d_i^C)\,, \quad \forall\, e \in \mathcal{E}, \tag{4.1}$$

an English bag-of-words representation of the Mandarin story d_i^C as used in standard vector-based information retrieval (Baeza-Yates et al., 1999; Salton and McGill, 1986).

Once we have obtained this English bag-of-words representation from a Chinese query document, our next step is to measure the similarity between this query

and documents in our English monolingual collection. Taking the bag-of-words representation as a vector, the cosine similarity is used to measure the distance between a query and a document.

$$\text{sim}(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \cdot |V_2|} = \frac{\sum_t V_{1,t} \cdot V_{2,t}}{\sqrt{\sum_t V_{1,t}^2} \cdot \sqrt{\sum_t V_{2,t}^2}}\,. \tag{4.2}$$

where the vectors are represented as $V_1 =< V_{1,1}, V_{1,2}, \cdots, V_{1,n} >$ and $V_2 =< V_{2,1}, V_{2,2}, \cdots, V_{2,m} >$.

This distance calculation is based on the frequency of terms (words) common to the two vectors, and as two vectors share more common terms—which means they are similar—the cosine similarity score will be higher, yielding a score close to 1. Conversely, if two vectors are *orthogonal* (independent), the score will be 0.

One important point to be considered in this vector based model is that it does not take into account the distinction between function words and content words. Obviously, sharing of many function words such as the, I, and we should not lead to the conclusion that two documents are similar. Therefore, it stands to reason that each word should be weighted based on whether it is a function word or not before computing cosine similarity—so that content words have higher weights than function words. To achieve this goal, some weighting methods such as TF-IDF weighting, or entropy-based weighting have been used. TF-IDF is based on the product of the raw term frequency (TF) and the inverse document frequency (IDF) which is defined as follows.

$$w_{ij} = \underbrace{tf_{ij}}_{\text{TF}} \underbrace{\log_2 \frac{N}{n_j}}_{\text{IDF}} \tag{4.3}$$

where w_{ij} the weight of term j in document i, tf_{ij} the frequency of term j in

document i, N is the total number of documents in the collection, and n_j is the number of documents where term j appears. The TF part emphasizes the term if the term appears frequently in the document; however, if the term appears in many documents—which is usually the case for the function words—the IDF part will have a smaller value to give a smaller weight for the word. Finally, the document d_i^E with the highest TF-IDF weighted cosine-similarity to d_i^C is then selected.

$$d_i^E = \arg\max_{d_j^E} \ \mathrm{sim}(P_{\text{CL-unigram}}(e|d_i^C), \hat{P}(e|d_j^E)) \tag{4.4}$$

Readers familiar with information retrieval literature will recognize our approach described in this secion to be the standard *query-translation* approach (Baeza-Yates et al., 1999) to CLIR. Notice that this is one simple and standard way to do CLIR, and there are many other alternatives. We will, indeed, describe another method for CLIR without resorting to the translation lexicon, cross-lingual latent semantic analysis, in Chapter 8.

4.3 Obtaining Stochastic Translation Lexicons $P_T(c|e)$ and $P_T(e|c)$

The last underlying assumption in our approach is that stochastic translation dictionaries, $P_T(c|e)$ and $P_T(e|c)$, are available. Notice that the reverse-translation lexicon $P_T(e|c)$ is needed for CLIR as explained in Section 4.2 and $P_T(c|e)$ is used for the translation of d_i^E into our target language Chinese as shown in Figure 4.1. The translation lexicons may be created out of an available electronic translation lexicon, with multiple translations of a word being treated as equally likely. Stemming and other morphological analyses may be applied to increase

the vocabulary-coverage of the translation lexicons.

In this section, we describe an alternate way to obtain the translation dictionaries automatically from the sentence-level aligned parallel corpus[5] and statistical machine translation techniques, such as the publicly available GIZA++ tools (Och and Ney, 2000) which are based on the IBM models (Brown et al., 1990, 1993). These tools use several iterations of the EM algorithm (Dempster et al., 1977) on increasingly complex word-alignment models to infer, among other translation model parameters, the conditional probabilities $P_T(c|e)$ and $P_T(e|c)$ of words c and e being mutual translations. Unlike standard MT systems, however, we will apply the translation models to entire articles, one word at a time, to get a *bag of translated words*. A sentence-aligned corpus is therefore not necessary for our purposes and a document-aligned corpus ought, in theory, to suffice for obtaining $P_T(c|e)$ and $P_T(e|c)$. In fact, in Chapter 7 and Chapter 8, respectively, we propose two alternate methods to extract translation lexicons—which are based on cross-lingual lexical triggers and cross-lingual latent semantic analysis—from a document-aligned corpus.

Finally, for truly resource-deficient languages, one may obtain a translation lexicon via optical character recognition from a printed bilingual dictionary (cf. Doermann et al. (2002)). This task is arguably easier than obtaining a large LM training corpus.

4.4 Cross-Lingual Unigram Distribution

As we already know the English documents which are relevant to a given Chinese test document, d_i^C, from the document correspondence, we seek a way to convert

[5]Here the sentence-level aligned parallel corpus means that for each *sentence* of one language, the translated *sentence* in the other language is available.

some useful statistics in English into Chinese. This may be done in two steps: first, extract the statistics from the English documents—here we propose using the unigram statistics for the cross-lingual LM adaptation—then translate it into Chinese.

Let $\hat{P}(e|d_i^E)$ denote the relative frequency of a word e in the document d_i^E, $e \in \mathcal{E}$, $1 \leq i \leq N$. It seems plausible that

$$P_{\text{CL-unigram}}(c|d_i^E) = \sum_{e \in \mathcal{E}} P_T(c|e)\hat{P}(e|d_i^E)\,, \quad \forall\, c \in \mathcal{C}, \tag{4.5}$$

would be a good unigram model for the i-th Mandarin story d_i^C. Even though higher order N-gram statistics such as bigrams or trigrams may be preferable, notice that the unigram statistic already conveys useful information such as topic, domain or genre for language model adaptation.

We propose using this cross-lingual unigram statistic to sharpen a static Chinese LM used for processing the test story d_i^C. One way to do this is via linear interpolation

$$P_{\text{CL-interpolated}}(c_k|c_{k-1}, c_{k-2}, d_i^E) = \lambda P_{\text{CL-unigram}}(c_k|d_i^E) + (1-\lambda)P(c_k|c_{k-1}, c_{k-2}) \tag{4.6}$$

of the cross-lingual unigram model (4.5) with a static trigram model for Chinese, where the interpolation weight λ may be chosen off-line to maximize the likelihood of some held-out Mandarin stories via the EM algorithm (Dempster et al., 1977). The improvement in (4.6) is expected from the fact that unlike the static text from which the Chinese trigram LM is estimated, d_i^E is semantically close to d_i^C and even the adjustment of unigram statistics, based on a stochastic translation model, may help. One such variation will be discussed with the experimental

results in Section 6.3.

Other variations on (4.6) are easily anticipated, such as

$$\begin{aligned}
&\tilde{P}_{\text{CL-interpolated}}(c_k|c_{k-1}, c_{k-2}, d_i^E) \\
&\quad = \frac{\lambda_{c_k} P_{\text{CL-unigram}}(c_k|d_i^E) + (1-\lambda_{c_k}) P(c_k|c_{k-1}, c_{k-2})}{\sum_{c\in\mathbf{C}} \lambda_c P_{\text{CL-unigram}}(c|d_i^E) + (1-\lambda_c) P(c|c_{k-1}, c_{k-2})},
\end{aligned} \tag{4.7}$$

where the interpolation weight may be chosen to let content-bearing words be influenced more by the cross-lingual cues than function words by making λ_{c_k} proportional to the *inverse document frequency* of c_k in the Chinese LM training text (cf. e.g. Coccaro and Jurafsky (1998)), log-linear interpolation with a global or word-dependent λ's, bucketing the λ's based on Chinese N-gram counts, etc. However, this approach may cause a data fragmentation problem: in other words, there are as many λ_c's as the number of vocabulary, and we will not have enough data to reliably determine the optimal λ_c's. Furthermore, determining λ_c's based on each word as opposed to each document completely ignores the adjustment of unigram statistics for each document.

Chapter 5

Language Model Perplexity Experiments

Before directly applying our cross-lingual language model adaptation techniques described in Chapter 4 to applications such as ASR or SMT, we should assure that our new approaches perform better than a state-of-the-art baseline approach. To this end, we perform a pilot study (Khudanpur and Kim, 2002). In other words, we investigate the effectiveness of our approach in an ideal setup: the perplexity experiments in a parallel text[1]. As described so far, our cross-lingual language model adaptation can be regarded as an efficient combination of two techniques: cross-lingual information retrieval and a simple form of machine translation. Since our test set documents come from a parallel text, the document correspondence—which is essential in our approach—is already given[2], and we don't have to worry about the CLIR step described in Section 4.2.

[1]By *parallel text*, we mean that it is comprised of texts written in two languages and for each document in one language, there is a translation of the document in the other language.

[2]We know this is not a realistic assumption; ordinarily, we cannot guarantee that there is a translation in the other language for any document in one language. Nevertheless, this is an experimental setup for a preliminary or pilot study to assure that our approach works in this ideal setup. Experimental results on more realistic situation will follow later.

Language	Chinese			English		
Corpus Partition	Train	Dev	Eval	Train	Dev	Eval
Number of Documents	16010	750	682	16010	750	682
Number of Word Tokens	4.2M	255K	177K	4.3M	263K	182K
Number of Characters	6.2M	376K	260K	–	–	–
Word-Vocabulary Size	41K		–	39K		–
Out-of-Vocabulary Rate	–	–	0.4%	–	–	0.4%

Table 5.1: Partition of the Hong Kong News corpus into training (Train), cross-validation and development (Dev) and evaluation (Eval) sets.

5.1 Parallel Corpus: Hong Kong News

We use the Hong Kong News parallel text corpus (Ma, 2000) for all the preliminary experiments reported in this chapter. The corpus contains 18,147 aligned translation-equivalent Chinese-English article pairs, dating from July 1997 to April 2000, released by the Information Services Department of Hong Kong Special Administrative Region of the People's Republic of China, through the Linguistic Data Consortium (LDC). After removing a few articles containing nonstandard Chinese characters, we divide the corpus, by random selection, into article-pairs for training, cross-validation, and evaluation. All the Chinese articles, training, development and evaluation sets included, have been automatically segmented into words (cf. Radev et al. (2001)), and the resulting corpus statistics are noted in Table 5.1.

Note that the statistics for the English portion of the corpus are in harmony with those for the Chinese portion due to the fact that the article-pairs are indeed translations of each other. We use the obvious notation C-train, C-dev and C-eval, and E-train, E-dev and E-eval to denote the six corpus partitions described in Table 5.1.

Only perplexity and out-of-vocabulary rate measurements are performed on

the evaluation portion of the corpus; no parameters are tuned on it, nor any iterative diagnostics performed.

5.2 Perplexity Experiments on the Hong Kong News Parallel Corpus

5.2.1 Baseline Chinese Language Model Estimation

We estimate a standard trigram LM, using Good-Turing discounting and Katz back-off, from C-train as noted in Section 5.1. Its perplexity on C-dev and C-eval is reported in Section 5.2.3. We considered basing all our Chinese LMs on character N-grams instead of words, but went with a word-based LM primarily because we believe that the cross-language cues will be directly beneficial to a word-based model.

Chinese LM discussions, particularly for ASR, frequently report *character perplexity* (instead of word perplexity) and character error rates, mainly to facilitate comparison across approaches that use different word-segmentations. We too report character perplexity: while calculating the average perplexity of a set of sentences, we simply divide the total log-probability of a sentence by the number of characters in the sentence rather than the number of segmented words. All our models, however, assign probability to entire words.

5.2.2 Estimating Statistical Translation Lexicons

The Hong Kong News corpus has been automatically aligned at the sentence level (cf. Radev et al. (2001)). We use GIZA++, a statistical machine translation training tool (Och and Ney, 2000), to train an IBM-model translation system

from the 16,000 article pairs from our training set. We extract the translation tables from GIZA++ and use them as translation lexicons $P_T(c|e)$ and $P_T(e|c)$. Note that since we apply these translation models word-by-word to entire English documents to get the statistic of (4.5), or to entire Chinese documents in (4.1) for CLIR, a sentence aligned corpus is not crucial.

5.2.3 Language Model Perplexities on Test Data

We first assume that the *true* alignment of a Chinese test document d_i^C with its English counterpart d_i^E is given—notice that this is an English-Chinese parallel text corpus, therefore for every Chinese document, there is a corresponding English document—and we compute the language model of (4.6), henceforth called the *CL-interpolated LM*. The interpolation weight λ is chosen to minimize the perplexity of the C-dev data, and then reused blindly on the C-eval data. We report the average perplexity[3] for C-dev and C-eval in Table 5.2; both word- and character-perplexity are reported for completeness.

Next, we relax the assumption that the true story-alignment is given; in other words, we pretend we don't know which English document corresponds to which Chinese document and vice versa. For each Chinese article d_i^C, we use the reverse translation model $P_T(e|c)$ described earlier to create an English bag-of-words representation of (4.1), and use it to find the English document with the highest cosine similarity as described in (4.4)—this document then plays the role of d_i^E. Again, the interpolation weight λ is chosen to minimize the perplexity of the

[3]Note that in results reported here, the entire article d_i^C is used to determine d_i^E, which in turn conditions the probability assigned to words in d_i^C. Strictly speaking, this is inappropriate conditioning of the probabilistic model. However, the theoretically correct version of conditioning the LM for each word c_k only on $c_1, \ldots, c_{k-1}$, is known from many other cases to produce nearly identical results—due to the robust determination of d_i^E even with small values of k—so we proceed with this somewhat tainted but defensible investigation.

Language model (# Words)	C-dev Perplexity		C-eval Perplexity	
	Word	Character	Word	Character
Baseline trigram (4.2M)	106	23.7	62.5	16.7
CL-interpolated with d_i^E from CLIR	90.1	21.2	51.3	14.6
CL-interpolated with true d_i^E	89.7	21.1	51.2	14.6

Table 5.2: Performance of story-specific language models with cross-lingual cues.

C-dev data, and reused blindly on the C-eval. Comparison with the true story-alignment in Table 5.2 indicates that the CL-interpolated LM is quite robust to CLIR errors[4]

As an aside, the correct English document is retrieved from E-dev for 92% of the articles in C-dev, and from E-eval for 89% of the articles in C-eval. The E-dev and E-eval sets are small in size relative to document collections used for benchmarking information retrieval systems. If one were looking at English newswire feed on a given day for an article to match a Chinese story, however, it ought to be feasible to narrow the search down to a few hundred candidate articles.

5.2.4 Contrast with Topic-Specific Language Models

The linear interpolation of the story-dependent unigram model (4.5) with a story-independent trigram model, as described in (4.6), is very reminiscent of monolingual topic-dependent language models (cf. (Clarkson and Robinson, 1997; Iyer and Ostendorf, 1999; Seymore and Rosenfeld, 1997)). This motivates us to construct topic-dependent LMs and contrast their performance with the models in Table 5.2. We proceed as follows.

[4]There is a concern that the results reported here are obtained when there *exists* the aligned English document for every Chinese document, which does not reflect the real situation. More challenging situations with no aligned document available will be investigated later.

The 16,000 articles in C-train are each represented by a bag-of-words vector $\hat{P}(c|d_i^C)$. These 16,000 vectors are then clustered into 100 classes using a standard K-means clustering algorithm. Random initialization is used to seed the algorithm, and standard TF-IDF weighted cosine-similarity is used as the "metric" for clustering. Five iterations of the K-means algorithm are performed, and the resulting 100 clusters are deemed to represent different *topics*. A bag-of-words *centroid* created from all the articles in a cluster is used to represent each topic. Topic-dependent unigram, and trigram LMs, denoted $P_t(c)$ and $P_t(c_k|c_{k-1}, c_{k-2})$ respectively, are also computed for each topic t exclusively from the articles in its cluster.

For each article d_i^C in C-dev or C-eval, a bag-of-words vector is generated in the same manner as was done for C-train, and the topic-centroid having the highest cosine-similarity to it is chosen as the topic t_i of d_i^C. Topic-dependent LMs are then constructed for each article d_i^C as

$$P_{\text{Topic-unigram}}(c_k|c_{k-1}, c_{k-2}, t_i) = \lambda P_{t_i}(c_k) + (1-\lambda)P(c_k|c_{k-1}, c_{k-2})\,, \qquad (5.1)$$

and

$$P_{\text{Topic-trigram}}(c_k|c_{k-1}, c_{k-2}, t_i) = \lambda P_{t_i}(c_k|c_{k-1}, c_{k-2}) + (1-\lambda)P(c_k|c_{k-1}, c_{k-2})\,. \qquad (5.2)$$

The development set C-dev is used to estimate the global interpolation weight λ. The perplexity of the resulting topic-dependent models is reported in Table 5.3.

We conclude from a comparison of Tables 5.2 and 5.3 that contemporaneous side-information, even if cross-lingual, is more accurate than the static topic-unigram statistics of (5.1), but (4.6) lacks the contextual awareness of the topic-trigram statistics of (5.2).

Language model (# Words)	C-dev perplexity		C-eval perplexity	
	Word	Char	Word	Char
Baseline trigram (4.2M)	106	23.7	62.5	16.7
Topic-unigram	94.6	21.9	57.4	15.8
Topic-trigram	84.4	20.3	49.3	14.2
Topic-trigram + CL-interpolated	80.1	19.6	44.6	13.3

Table 5.3: Performance of topic-specific language models, and their interpolation with story-specific models that use cross-lingual cues.

An obvious experiment is to interpolate the cross-language unigram and the topic trigram models with the baseline trigram model, which we do and report in Table 5.3. The further reduction in perplexity suggests that the topic-wide and article-specific cues obtained from the two models are considerably complementary.

5.3 Summary

In this chapter, we have reported the pilot study results of our language model adaptation approach. We have demonstrated 15–18% reduction in perplexity by applying cross-lingual language model adaptation to the state-of-the-art trigram language model. We have achieved further gains (25–29% reductions in perplexity) by combining cross-lingual language model adaptation with the topic-based trigram language models. These results indicate that our approach is effective itself, and it works complementary to the topic based language models—even though this experimental setup may not be realistic. The next question is whether our approach will work in the real situation where the translated (aligned) documents in the other language may not be available, which we are going to investigate in Chapter 6.

Chapter 6

Automatic Speech Recognition Experiments

We investigate the use of the cross-lingual language modeling techniques described so far for improving ASR performance on Mandarin news broadcasts using English newswire texts. We have chosen the experimental ASR setup created in the 2000 Johns Hopkins Summer Workshop to study Mandarin pronunciation modeling, extensive details about which are available in Fung et al. (2000). The acoustic training data (∼10 hours) for their ASR system was obtained from the 1997 Mandarin Broadcast News distribution, and context-dependent state-clustered models were estimated using initials and finals as subword units.

In addition to the acoustic model training data, text-only corpora—which are written in different languages—may be exploited for our approaches. In essence, our approaches can be summarized as an efficient combination of CLIR and MT for language model adaption, to which translation lexicons, $P_T(c|e)$ and $P_T(e|c)$ are needed. We have proposed a method to obtain the translation lexicons using a publicly available GIZA++ toolkit (Och and Ney, 2000) which is based

on the IBM MT models (Brown et al., 1990, 1993). We first build the translation lexicons automatically from an English-Chinese parallel corpus and use the same translation tables, acquired from the Hong Kong News parallel text, as described in Section 5.1. Besides, an extensive amount of English monolingual text is available—as English is a resource-rich language for sure—which is used for CLIR. Notice that, unlike the experiments on the parallel text described in Chapter 5, we do not guarantee that the corresponding English document is available for a given Chinese test document because this is not a parallel corpus. Certainly, this is a more challenging problem; nevertheless it is a more realistic situation which motivates us to proceed. Finally, Chinese monolingual texts are used for building static baseline language models.

6.1 Monolingual corpora

Two Chinese text corpora and an English corpus are used to estimate LMs in our experiments. A vocabulary $\mathcal{C}$ of 51K Chinese words, used in the ASR system, is used to segment the training text into words. This vocabulary gives an out--of-vocabulary (OOV) rate of 5% on the manually word-segmentation of the test data.

XINHUA: We use the Xinhua News corpus (Fung et al., 2000) of about 13 million words to represent the scenario when the amount of available LM training text borders on adequate, and we estimate a baseline trigram LM from it for one set of experiments.

HUB-4NE: We also estimate a trigram model *only* from the 96K words in the transcriptions used for training acoustic models in our ASR system. This corpus represents the scenario when little or no additional text is available to train LMs.

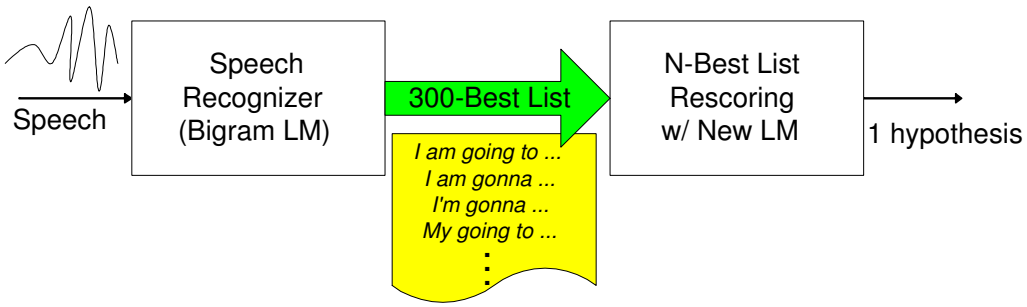

Figure 6.1: N-best (N=300) list rescoring of ASR

NAB-TDT: English text contemporaneous with the test data is often easily available. For our test set, described below, we select (from the North American News Text corpus) articles published in 1997 in The Los Angeles Times and The Washington Post, and articles from 1998 in the New York Times and the Associated Press news service (from TDT-2 corpus). This amounts to a collection of roughly 45,000 articles containing about 30-million words of English text; a modest collection by CLIR standards.

Our ASR test set is a subset (Fung et al., 2000) of the NIST 1997 and 1998 HUB-4NE benchmark tests, containing Mandarin news broadcasts from three sources for a total of about 9800 words.

We proceed to ASR experiments by rescoring the N-best lists as shown in Figure 6.1. We establish two baseline systems to simulate the different language classes in the world. There are some languages where hundreds of millions of word texts are already available such as English, German, and French. For those languages we can apply the state-of-the-art language model techniques to build accurate language models. On the other hand, there are some other languages only about tens of millions of word texts are available such as Arabic. Our methods may be useful for those class languages, and we build one baseline LM with the XINHUA 13M words corpus to simulate those languages. Finally, there are the

other languages even the tens of millions of word corpus is not easily available. For those languages, we establish another baseline, namely HUB-4NE with 96K words. We generate two sets of lattices using the baseline acoustic models and *bigram* LMs estimated from XINHUA and HUB-4NE. All our LMs are evaluated by rescoring 300-best lists extracted from these two sets of lattices. The 300-best lists from the XINHUA bigram LM are used in all XINHUA experiments, and those from the HUB-4NE bigram LM in all HUB-4NE experiments. We report both word error rates (WER) and character error rates (CER), the latter being independent of any difference in segmentation of the ASR output and reference transcriptions. The oracle best (worst) WERs for 300-best list are 34.4% (94.4%) for the XINHUA LM and 39.7% (95.5%) for the HUB-4NE LM.

6.2 Baseline ASR Performance of Cross-Lingual LMs

Given the 300-best lists for each of XINHUA and HUB-4NE baselines, we first rescore the lists with the trigram LMs. Again, each trigram LM is estimated using standard Good-Turing discounting and Katz back-off. The baseline ASR WER and CER are 49.9% and 28.8% for XINHUA, and 60.1% and 44.1% for HUB-4NE, as shown in Table 6.1.

For each test story d_i^C, next, we perform CLIR using the first pass ASR output to choose the most similar English document d_i^E from NAB-TDT. Then we create the cross-lingual unigram model of (4.5) and finally interpolate it with the trigram model as in (4.6). Here, the optimal interpolation weight λ can be found via the EM algorithm (Dempster et al., 1977) so as to maximize the likelihood of some held-out data. In our experiments, we use the 1-best hypotheses of all test

Language model	Perp	WER	CER	p-value
XINHUA trigram	426	49.9%	28.8%	–
CL-interpolated	375	49.5%	28.7%	0.208
HUB-4NE trigram	1195	60.1%	44.1%	–
CL-interpolated	750	59.3%	43.7%	< 0.001

Table 6.1: Word perplexity and ASR WER of LMs based on single English document and global λ.

utterances from the first pass ASR output as our held-out data. In other words, all free parameters are optimized to minimize the perplexity of 1-best hypotheses. All translation tables, $P_T(e|c)$ for CLIR and $P_T(c|e)$ for back translation into Chinese come from GIZA++ translation tables, as explained in Section 5.2.2. Table 6.1 shows the CL-interpolated models' perplexity and WER for XINHUA and HUB-4NE.

All p-values reported in this paper are based on the standard NIST MAPSSWE test (Pallett et al., 1990), and indicate the statistical significance of a WER improvement over the corresponding trigram baseline, unless otherwise specified.

We have achieved a small improvement in WER, 0.4% absolute, by using CL-interpolated LM when the baseline LM (XINHUA) has a reasonable size of LM training texts (13M words). Nonetheless, the improvement brought by CL-interpolated LM is not statistically significant on XINHUA as the high (> 0.05) p-value implies. On HUB-4NE however, where Chinese LM text is scarce, the CL-interpolated LM delivers considerable benefits via the large English corpus.

6.3 Likelihood-Based Story-Specific Selection of Interpolation Weights and the Number of English Documents per Mandarin Story

The experiments above naïvely used the one most similar English document for each Mandarin story, and a global λ in (4.6)—no matter how similar the best matching English document is to a given Mandarin news story. One possible argument against this approach is that there may not always exist an English document which is relevant to the Chinese test document. Conversely, there may be more than one document which is relevant; if this is the case, some useful information may be lost. Remember that the CLIR results are a ranked list of English documents along with relevance or similarity scores. Rather than choosing the one most similar English document from NAB-TDT, it stands to reason that choosing more than one English document may be helpful if many have a high similarity score, and perhaps not using even the best matching document may be fruitful if the match is sufficiently poor. It may also help to have a greater interpolation weight $\lambda_{d_i^E}$ for stories with good matches, and a smaller $\lambda_{d_i^E}$ for others. For experiments in this section, we select a different $\lambda_{d_i^E}$ for each test story, again based on maximizing the likelihood of the 1-best output given a CL-Unigram model. The other issue then is the choice and the number of English documents to translate.

N-best documents: One could choose a predetermined number N of the best matching English documents for each Mandarin story. Table 6.2 shows the test set perplexity when top N documents are used for building cross-lingual language models (Xinhua baseline case). As the table shows, the case with $N = 30$ gave us

Top N	Perp
1	375
10	364
30	359
50	361
100	362
500	365

Table 6.2: Word perplexity with N-best documents (Xinhua baseline)

the best LM performance.

All documents above a similarity threshold: The argument against always taking a predetermined number of the best matching documents may be that it ignores the goodness of the match. An alternative is to take all English documents whose similarity to a Mandarin story exceeds a certain predetermined global threshold. As this threshold is lowered, starting from a high value, the *order* in which English documents are selected for a particular Mandarin story is the same as the order when choosing the N-best documents, but the number of documents selected now varies from story to story. We experimented with various thresholds, and Table 6.3 shows the perplexity results again with the Xinhua baseline case. We found that while a threshold of 0.12 gives us the lowest perplexity on the test set. However, this method does not take into account of dynamic range of similarities for each test document. In other words, it is possible that for some stories, even the best matching English document falls below the threshold at which other stories have found more than one good match. This points to the need for a story-specific strategy for choosing the number of English documents, instead of a global threshold.

Likelihood-based selection of the number of English documents: Figure 6.2 shows the perplexity of the reference transcriptions of one typical test story

Threshold N	Perp
0.10	362
0.12	361
0.14	366
0.16	366
0.18	366
0.20	365

Table 6.3: Word perplexity with different similarity thresholds (Xinhua baseline)

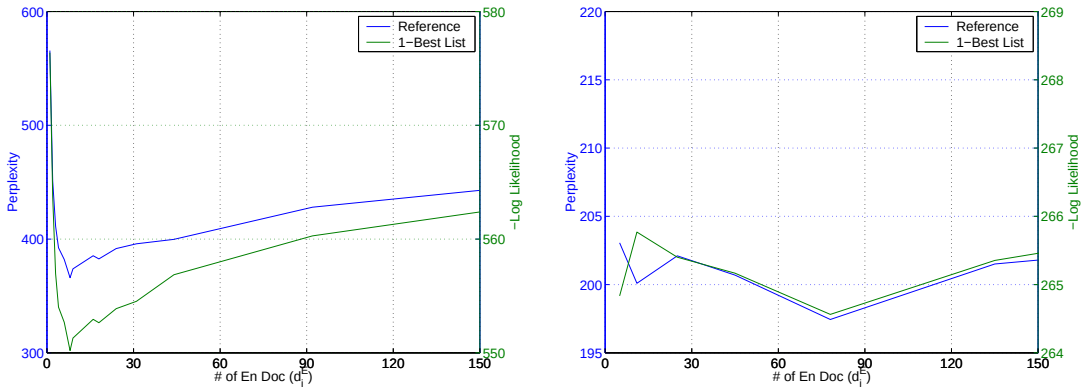

Figure 6.2: Perplexity of the reference transcription and the likelihood of the ASR output v/s number of d_i^E for typical test stories.

under the LM (4.6) as a function of the number of English documents chosen for creating (4.5). For each choice of the number of English documents, the interpolation weight $\lambda_{d_i^E}$ in (4.6) is chosen to maximize the likelihood (also shown) of the first pass output. This suggests that choosing the number of English documents to maximize the likelihood of the first pass ASR output is a good strategy.

For each Mandarin test story, we choose the 1000-best-matching English documents and divide the *dynamic range* of their similarity scores evenly into 10 intervals. Next, we choose the documents in the top $\frac{1}{10}$-th of the *range of similarity scores*, not necessarily the top-100 documents, compute $P_{\text{CL}-\text{unigram}}(c|d_i^E)$, determine the $\lambda_{d_i^E}$ in (4.6) that maximizes the likelihood of the first pass output of only the utterances in that story, and record this likelihood. We repeat this with

Language model	Perp	WER	CER	p-value
XINHUA trigram	426	49.9%	28.8%	–
CL-interpolated	346	48.8%	28.4%	< 0.001
Topic-trigram	381	49.1%	28.4%	0.003
Topic + CL-interpolated	326	48.5%	28.2%	< 0.001
HUB-4NE trigram	1195	60.1%	44.1%	–
CL-interpolated	630	58.8%	43.1%	< 0.001
Topic-trigram	1122	60.0%	44.1%	0.660
Topic + CL-interpolated	631	59.0%	43.3%	< 0.001

Table 6.4: Word perplexity and ASR WER of LMs with a likelihood-based story-specific selection of the number of English documents d_i^E's and interpolation weight $\lambda_{d_i^E}$ for each Mandarin story.

documents in the top $\frac{2}{10}$-th of the range of similarity scores, the top $\frac{3}{10}$-th, etc., and obtain the likelihood as a function of the similarity threshold. We choose the threshold that maximizes the likelihood of the first pass output. Thus the number of English documents d_i^E in (4.5), as well as the interpolation weight $\lambda_{d_i^E}$ in (4.6), are chosen dynamically for each Mandarin story to maximize the likelihood of the ASR output. Table 6.4 shows ASR results for this *likelihood-based story-specific adaptation* scheme.

Note that significant WER improvements are obtained from the CL-interpolated LM using likelihood-based story-specific adaptation even for the case of the XINHUA LM. Furthermore, the performance of the CL-interpolated LM is even better than the topic-dependent LM. This is remarkable, since the CL-interpolated LM is based on unigram statistics from English documents, while the topic-trigram LM is based on trigram statistics. We believe that the contemporaneous and story-specific nature of the English document leads to its relatively higher effectiveness. Our conjecture, that the *contemporaneous* cross-lingual statistics and *static* topic-trigram statistics are complementary, is supported by the significant additional improvement in WER obtained by the interpolation of the two LMs,

as shown on the last line for XINHUA.

The significant gain in ASR performance in the resource deficient HUB-4NE case are obvious. The small size of the HUB-4NE corpus makes topic-models ineffective.

6.4 Comparison of Stochastic Translation with Manually Created Dictionaries

While the notion of inducing stochastic translation lexicons from aligned bilingual text is appealing, it is also worth investigating the scenario in which a modest sized, manually created, machine readable dictionary (MRD) is available. We used a widely available Chinese-English translation lexicon[1] for this purpose. In order to make meaningful comparisons, we used the very same procedure for CLIR and CL-unigram construction as done for the models reported in Table 8.2, but we used the MRD in place of a stochastic translation lexicon. In other words, instead of using the translation probabilities derived using GIZA++ for the CL-interpolated LM, cross-lingual triggers for the Trig-interpolated LM and CL-LSA for the LSA-interpolated LM respectively, we used 18K English-to-Chinese entries and 24K Chinese-to-English entries from the LDC translation lexicon. It is clear from the results reported in Table 6.5 that while the MRD leads to a reduction in perplexity, no reduction in WER is obtained.

This *should not* lead the reader to conclude that an MRD can be completely dispensed-with for cross-lingual applications. An MRD is often crucial for obtaining bilingual text-alignment of acceptable quality. Recall that our techniques are predicated on the capability to obtain such bilingual text.

[1]See `http://www.ldc.upenn.edu/Projects/Chinese/LDC_ch.htm`

Language Model	Perp	WER	CER
XINHUA Trigram	426	49.9%	28.8%
MRD-interpolated	387	49.9%	29.0%
HUB-4NE Trigram	1195	60.1%	44.1%
MRD-interpolated	770	60.1%	44.1%

Table 6.5: Word perplexity and ASR WER comparisons using a machine readable dictionary

We instead conclude that the best application of an MRD in a resource-deficient language is in the process of obtaining either sentence- or document-aligned bilingual text from available text repositories. A stochastic translation lexicon derived from such text should then be used in the actual cross-lingual application.

6.5 Summary

We have proposed using CLIR and MT to improve a statistical language model in a resource-deficient language by exploiting copious amounts of text available in resource-rich languages. When transcribing a news story in a resource-deficient language, the core idea is to use the first pass output of a rudimentary ASR system as a query for CLIR, identify contemporaneous English documents on that news topic, followed by MT to provide a rough translation which, even if not fluent, is adequate to update estimates of word frequencies and the LM vocabulary. We have shown that a query-translation approach can be used for CLIR and also covered how to obtain translation lexicons automatically from a sentence-aligned corpus (Khudanpur and Kim, 2004).

In spite of significant improvements using our approach, there are some shortcomings in this method. Specifically, stochastic translation lexicons estimated using the IBM method (Brown et al., 1993) from a fairly large *sentence-aligned*

Chinese-English parallel corpus are used—a considerable demand especially for a resource-deficient language. As suggested above, an easier-to-obtain *document-aligned* comparable corpus may suffice. We propose in Chapter 7 and Chapter 8 two alternatives which do not require an expensive sentence-aligned corpus.

Chapter 7

Cross-Lingual Lexical Triggers

It seems plausible that most of the information one gets from the cross-lingual unigram LM of (4.5) is in the form of the altered statistics of topic-specific Chinese words conveyed by the statistics of content-bearing English words in the matching story. The translation lexicon used for obtaining the information, however, is an expensive resource. Yet, if one were only interested in the conditional distribution of Chinese words given some English words, there is no reason to require translation as an intermediate step. In a monolingual setting, the *average mutual information* between lexical pairs co-occurring anywhere within a long "window" of each-other has been used to capture statistical dependencies not covered by N-gram LMs (Lau, 1994; Rosenfeld, 1994, 1996; Tillmann and Ney, 1997). We note that even though no distinction is made between content-bearing and function words in the process of selecting trigger pairs, a vast majority of trigger-pairs turn out to be content-bearing words. We use this inspiration to propose the following notion of cross-lingual lexical triggers (Kim and Khudanpur, 2003a). In order to contrast our cross-lingual trigger approach, we begin with the review of the monolingual trigger approach which was proposed by Rosenfeld (1994, 1996).

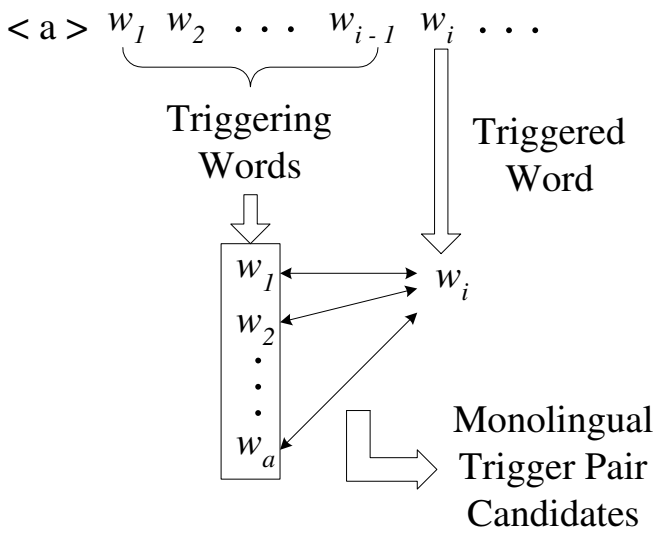

Figure 7.1: Monolingual trigger pair candidates

7.1 Previous Work: Monolingual Triggers

In a monolingual setting, a pair of words (a, b) is considered a trigger pair if, given a word position in a sentence, the occurrence of a in any of the preceding word positions significantly alters the (conditional) probability that the word in the given position is b: a is said to *trigger* b. E.g., the occurrence of either significantly increases the probability of or subsequently in the sentence. The set of preceding word positions is variably defined to include all words from the beginning of the sentence, paragraph or document, or is limited to a fixed number of preceding words, limited of course by the beginning of the sentence, paragraph or document.

The obvious questions are then 1) how to automatically select these trigger pairs from texts and 2) how to actually use the selected trigger pairs for language modeling. Selecting trigger pairs can further be divided into two steps: selecting trigger pair *candidates* and selecting final trigger pairs.

Figure 7.1 shows a snapshot of selecting trigger pair candidates in a monolingual setting. Suppose we are at the point of processing the i^{th} word, w_i in a training set. All words in the history window which spans from the beginning of the document (<a>) to the $(i-1)^{th}$ word—$w_1, w_2, \cdots w_{i-1}$—are considered as possible *triggering* words[1]. Taking the i^{th} word as the *triggered* word, the possible trigger pair candidates[2] are (w_1, w_i), (w_2, w_i), $\cdots$ (w_a, w_i). As the triggered word position is advanced to the next word w_{i+1}, the triggering word window moves forward correspondingly to include w_i, and this procedure is repeated for the whole training text. Notice that the triggering word window will be cleared when a document boundary marker appears; i.e., when we encounter a new document. Each occurrence of these pairs along the training document gives us one co-occurrence of the pair, and the co-occurrence will be accumulated. Finally, these counts will become the bases for measuring the average mutual information statistic. In other words, out of these candidates, the pairs that have high average mutual information values will be chosen as trigger pairs.

One consideration before computing average mutual information of a candidate pair is that $|V|^2$ trigger pair candidates are possible where V is the vocabulary. Not only is it computationally infeasible for practical ASR, but also most of them would not be selected as trigger pairs. Hence, Rosenfeld (1994, 1996) first filtered out some infrequent word pairs that appeared less than a certain number of times and then considered only the remaining frequent pairs.

Given the trigger pair candidates, the average mutual information can be easily computed. Of course, there are many possible ways to select trigger pairs based

[1]Notice, in the figure, that triggering words are $w_1, w_2, \cdots, w_a$ which implies they are in fact a bag of words ignoring all repetitions.

[2]In fact, Rosenfeld (1994, 1996) excluded the last two context words from the triggering word window as they are the context words for the trigram model, and he used the trigger model in addition to the trigram model.

on the average mutual information. In Rosenfeld's approach, the pairs having an average mutual information score lower than a certain threshold are first removed. Finally, from the remaining candidates, three or six triggering words that have the highest average mutual information for each triggered word are selected.

One of the findings in his monolingual trigger experiments is that many of the trigger pairs are in fact the *self triggers*: the appearance of a word triggers the appearance of itself in the near future. E.g., if the word Havana appeared in the previous history, the chances are the word Havana will appear again. Also, the semantically related words such as Cuba, Castro, and Miami, or common root words may trigger the word Havana (Rosenfeld, 1994, 1996).

Once the trigger pairs are selected, Rosenfeld (1994, 1996) continued with building maximum entropy models using trigger pairs as a set of constraints as well as conventional N-gram constraints resulting in the ME trigger model. Linear interpolation of the ME trigger model and a trigram language model was tested as well as the ME trigger model alone. He achieved about 23%–25% perplexity reductions over the trigram backoff language model in the 5M word Wall Street Journal corpus experiments with the interpolated model.

7.2 Cross-Lingual Trigger Pair Candidates

This section proposes using the trigger idea from a monolingual setup for our cross-lingual approach. That is, in the cross-lingual setting, we consider a pair of words (e, c), $e \in \mathcal{E}$ and $c \in \mathcal{C}$, to be a trigger pair if, given an English-Chinese pair of aligned documents, the occurrence of e in the English document significantly alters the (conditional) probability that the word c appears in the Chinese document: e is said to trigger c. Extending the idea that most monolingual

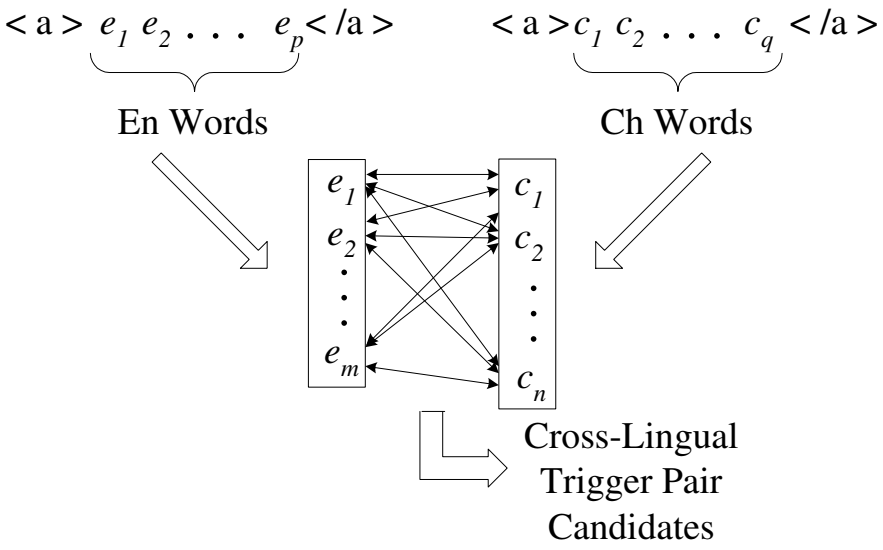

Figure 7.2: Cross-lingual trigger pair candidates

trigger pairs are self triggers or semantically related pairs, it is not difficult to see that cross-lingual trigger pairs would be direct translation pairs—as the triggering word is in one language and the triggered word is in the other language. Notice, however, it is not necessary for a cross-lingual trigger pair to be a translation pair. E.g., the occurrence of Belgrade in the English document may trigger the Chinese transliterations of Serbia and Kosovo, and possibly the translations of China, embassy and bomb! By inferring trigger pairs from a document-aligned corpus of Chinese-English articles, we expect to be able to discover semantically- or topically-related pairs in addition to translation equivalences.

Figure 7.2 shows the trigger pair candidates in a cross-lingual setting. Notice that this time the text is a document-aligned text. In the figure, <a> and are document boundary markers, not sentence boundaries markers, and thus, there may be many sentences between these markers. We assume document level aligned texts may be used even though more refined sentence level aligned texts can be also used. Similarly to the monolingual case, all words in both of Chinese

and English are represented as a bag of words prior to be considered as trigger pair candidates.

Notice the difference to the monolingual trigger pair candidates as shown in Figure 7.1. Remember, in the monolingual setting, the triggering word window is dynamically changed as we move forward (right) to the next word, and all words in the triggering word window are considered to trigger only one triggered word. On the other hand in the cross-lingual setting, the triggering word window is fixed once we have a document pair, and it remains unchanged unless we encounter a new Chinese-English document pair. And all possible combinations of the words in both language are considered. Again, due to the computational overhead, we consider the pairs of words only appear more than a certain threshold times in the whole corpus.

7.3 Average Mutual Information

Once we have found candidate trigger pairs, the next thing to do is to select the final trigger pairs out of these candidates. In a monolingual setting, the average mutual information between the words in the candidate trigger pairs has been used (Lau, 1994; Rosenfeld, 1996). We also use the same measure to select the cross-lingual trigger pairs. As explained in the previous section, however, the way how to select trigger pair candidates is different. Here, we give a brief introduction to the average mutual information.

In information theory (Cover and Thomas, 1991), the uncertainty of a random variable X is measured as the *entropy*, $H(X)$,

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x) \,. \tag{7.1}$$

Also, the conditional entropy is a measure of the average information in X given that another random variable Y is known. In other words, it is the remaining uncertainty of X after observing Y:

$$\begin{aligned} H(X|Y) &= -\sum_{y\in\mathcal{Y}} P(y)H(X|Y=y) && (7.2)\\ &= -\sum_{y\in\mathcal{Y}} P(y)\sum_{x\in\mathcal{X}} P(x|y)\ \log P(x|y) && (7.3)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\ \log P(x|y)\,. && (7.4) \end{aligned}$$

The average mutual information $I(X;Y)$ between X and Y is defined as the average information about X gained by observing Y,

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) && (7.5)\\ &= -\sum_{x\in\mathcal{X}} P(x)\log P(x) - (-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x|y)) && (7.6)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x|y) && (7.7)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\left(\frac{P(x,y)}{P(x)P(y)}\right). && (7.8) \end{aligned}$$

The average mutual information, $I(X;Y)$, is in fact a measure how much information one random variable provides about another, or the reduction of uncertainty of X due to the knowledge of Y. In other words, if two variables X and Y are closely related each other, $I(X;Y)$ would be high. Moreover, it has the following properties.

1. If X and Y are independent, then $I(X,Y)=0$.

   If X and Y are independent, in probability theory, the joint probability of the two random variables is same as the product of the probability of each

random variable ($P(X,Y) = P(X)P(Y)$), and thus, $I(X,Y) = 0$. This indeed makes sense: if they are independent random variables, then Y can tell us nothing about X.

2. Mutual information is symmetric.

$$\begin{aligned} I(Y;X) &= H(Y) - H(Y|X) && (7.9)\\ &= -\sum_{y\in\mathcal{Y}} P(y)\log P(y) + \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P(x,y)\log P(y|x) && (7.10)\\ &= \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P(x,y)\log\left(\frac{P(x,y)}{P(y)P(x)}\right) && (7.11)\\ &= I(X;Y) && (7.12) \end{aligned}$$

That is, Y tells us as much about X as X does about Y.

3. Mutual information is nonnegative: $I(X,Y) \geq 0$.

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\left(\frac{P(x,y)}{P(x)P(y)}\right) && (7.13)\\ &\geq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\left(\frac{\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)}{\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x)P(y)}\right) && (7.14)\\ &= 0. && (7.15) \end{aligned}$$

The inequality (7.14) holds by the log sum inequality (Cover and Thomas, 1991).

7.4 Selection of Cross-Lingual Triggers

As shown in Section 7.1, the way to select (identify) trigger pairs in monolingual settings is straightforward: we simply count how many times a word appears in a triggering window, and how many times a triggered word appears, estimate the necessary probabilities from these counts, and finally select the trigger pairs based on the average mutual information estimates. Similarly, we can select the cross-lingual trigger pairs once we compute the average mutual information for every English-Chinese word pair (e, c) after collecting some necessary statistics, as follows.

Let $\{d_i^E, d_i^C\}$, $i = 1, \ldots, N$, now be a document-aligned training corpus of English-Chinese article pairs. Let $\#d(e, c)$ denote the *document frequency*, i.e., the number of aligned article-pairs, in which e occurs in the English article and c in the Chinese. Let $\#d(e, \bar{c})$ denote the number of aligned article-pairs in which e occurs in the English articles but c does not occur in the Chinese article. Let

$$P(e, c) = \frac{\#d(e, c)}{N}, \tag{7.16}$$

$$P(e, \bar{c}) = \frac{\#d(e, \bar{c})}{N}, \tag{7.17}$$

$$P(\bar{e}, c) = \frac{\#d(\bar{e}, c)}{N}, \tag{7.18}$$

$$\text{and} \quad P(\bar{e}, \bar{c}) = \frac{\#d(\bar{e}, \bar{c})}{N}. \tag{7.19}$$

Next, let $\#d(e)$ denote the number of English articles in which e occurs. Set

$$P(c) = \frac{\#d(c)}{N} \quad \text{and} \quad P(\bar{c}) = \frac{\#d(\bar{c})}{N}, \tag{7.20}$$

and

$$P(c|e) = \frac{P(e,c)}{P(e)} \quad \text{and} \quad P(c|\bar{e}) = \frac{P(\bar{e},c)}{P(\bar{e})}\,. \tag{7.21}$$

Similarly, define $P(e)$, $P(\bar{c}|e)$, etc. Finally, let

$$\begin{aligned} I(e;c) \quad &= P(e,c)\log\frac{P(c|e)}{P(c)} + P(e,\bar{c})\log\frac{P(\bar{c}|e)}{P(\bar{c})} \\ &+P(\bar{e},c)\log\frac{P(c|\bar{e})}{P(c)} + P(\bar{e},\bar{c})\log\frac{P(\bar{c}|\bar{e})}{P(\bar{c})}\,. \end{aligned} \tag{7.22}$$

We propose to select word pairs with the high average mutual information as cross-lingual lexical triggers.

There are $|\mathcal{E}|\times|\mathcal{C}|$ possible English-Chinese word pairs which may be prohibitively large to search for the pairs with the highest mutual information. Therefore, from the training set of the Hong Kong News parallel corpus in Table 5.1, we first filter out infrequent words in each language, say, words appearing less than five times, then measure $I(e;c)$ for all possible pairs from the remaining words, sort them by $I(e;c)$, and select, say, the top one million pairs.

7.5 Estimating Trigger LM Probabilities

Once we have chosen a set of trigger pairs, the next step is to estimate the probability $P_{\text{Trigger}}(c|e)$ in lieu of the translation probability $P_T(c|e)$ in (4.5).

Following the maximum likelihood approach proposed by Tillmann and Ney (1997), one could choose the trigger probability $P_{\text{Trigger}}(c|e)$ to be based on the unigram frequency of c among Chinese word tokens in that subset of aligned

documents d_i^C which have e in d_i^E, namely

$$P_{\text{Trigger}}(c|e) = \frac{\sum_{i\,:\,d_i^E \ni e} N_{d_i^C}(c)}{\sum_{c' \in \mathcal{C}} \sum_{i\,:\,d_i^E \ni e} N_{d_i^C}(c')}\,. \tag{7.23}$$

As an alternative to (7.23), we experimented with

$$P_{\text{Trigger}}(c|e) = \frac{I(e;c)}{\sum_{c' \in \mathcal{C}} I(e;c')}, \tag{7.24}$$

where we set $I(e;c) = 0$ whenever (e,c) is not a trigger-pair. This is the definition of $P_{\text{Trigger}}(c|e)$ used henceforth in this dissertation. Analogous to (4.5), we set

$$P_{\text{Trigger-unigram}}(c|d_i^E) = \sum_{e \in \mathcal{E}} P_{\text{Trigger}}(c|e)\hat{P}(e|d_i^E)\,, \tag{7.25}$$

and, again, we build the interpolated model

$$\begin{aligned} & P_{\text{Trigger-interpolated}}(c_k|c_{k-1}, c_{k-2}, d_i^E) \\ & \quad = \quad \lambda_{d_i^E} P_{\text{Trigger-unigram}}(c_k|d_i^E) + (1 - \lambda_{d_i^E}) P(c_k|c_{k-1}, c_{k-2}). \end{aligned} \tag{7.26}$$

Comparison of Trigger Estimation of (7.23) and (7.24)

We compare the alternative $P_{\text{Trigger}}(\cdot|\cdot)$ definitions (7.23) and (7.24) for replacing $P_T(\cdot|\cdot)$ in (4.5). The resulting CL-interpolated LM (4.6) yields a perplexity of 370 on the XINHUA test set using (7.23), compared to 367 using (7.24). Similarly, on the HUB-4NE test set, using (7.23) yields 736, while (7.24) yields 727. Therefore, (7.24) has been used.

Language model	Perp	WER	CER	p-value
XINHUA trigram	426	49.9%	28.8%	–
CL-interpolated	346	48.8%	28.4%	< 0.001
Trigger-interpolated	367	49.1%	28.6%	0.004
HUB-4NE trigram	1195	60.1%	44.1%	–
CL-interpolated	630	58.8%	43.1%	< 0.001
Trigger-interpolated	727	58.8%	43.3%	< 0.001

Table 7.1: Perplexity and ASR performance comparisons with triggers and stochastic translation dictionaries

7.6 Comparison of Cross-Lingual Triggers with Stochastic Translation Dictionaries

Once we select the cross-lingual trigger pairs as described in Section 7.4, $P_T(c|e)$ in (4.5) is replaced by $P_{\text{Trigger}}(c|e)$ of (7.24), and $P_T(e|c)$ in (4.1) by $P_{\text{Trigger}}(e|c)$. Therefore, given a set of the cross-lingual trigger pairs, the trigger-based models are free from requiring a translation lexicon. Furthermore, a document-aligned comparable corpus—which is much easier to obtain—is all that is required to construct the set of trigger-pairs. We otherwise follow the same experimental procedures as in Section 6.3.

As Table 7.1 shows, the trigger-based model (Trigger-interpolated) performs only slightly worse than the CL-interpolated model. We have the following explanations for this slight degradation.

- The CL-interpolated model is trained from the sentence-aligned corpus while the trigger-based model is from the document-aligned corpus. There are two steps which could be affected by this difference, one being CLIR and the other being the translation of the d_i^E's into Chinese. Some errors in CLIR may however be masked by our *likelihood-based story-specific adaptation*

scheme, since it finds optimal retrieval settings, dynamically adjusting the number of English documents as well as the interpolation weight, even if CLIR performs somewhat suboptimally. Furthermore, a document-aligned corpus is much easier to build. Thus a much bigger and more reliable comparable corpus may be used, and eventually more accurate trigger-pairs will be acquired.

- The CL-interpolated model uses the IBM models (Brown et al., 1990, 1993) which are based on more detailed statistics than simple word pair occurrences in parallel texts on which the trigger model is based. In other words, the training of the IBM models uses several iterations of the EM algorithm (Dempster et al., 1977) under the assumption that there are word-to-word alignments between a pair of segments[3] in the parallel texts. In addition, there are other complicated models such as the distortion model and the fertility model which are suitable for the complete translation of segments in one language into ones in the other language. Remember that all we need is a simple form of translations from unigrams in one language to the ones in the other language. In consequence, the translation tables obtained by the IBM models contain only exact translation pairs, while the tables obtained by the cross-lingual trigger approach contain many semantically related translation pairs. Table 7.2 shows an example of dictionary entries for the same Chinese word when the IBM models were used and cross-lingual triggers were used. Clearly, cross-lingual trigger based dictionary contains many entries which are not the exact translations, but the semantically related translations.

[3]Typically, one segment corresponds to one sentence; however, one segment may contain two or more sentences.

c	e	$P_T(e\|c)$	c	e	$P_{\text{Trigger}}(e\|c)$
足球	❶ football	0.51556	足球	❶ football	0.18002
	❷ soccer	0.13127		❷ soccer	0.10248
	❸ fee	0.03851		❸ sports	0.08954
	❹ classes	0.03771		❹ pitch	0.07103
	❺ this	0.03487		❺ basketball	0.05353
	❻ hkfa	0.03072		❻ provisional	0.04551
	❼ mini-soccer	0.02165		❼ badminton	0.04154
	❽ personal	0.02011		❽ behalf	0.03829
	❾ females	0.01668		❾ competitions	0.03585
	❿ fu	0.01524		❿ tennis	0.03339
	⋮	⋮		⋮	⋮

Table 7.2: Sample dictionary entries from IBM model-based dictionary (left) and cross-lingual trigger-based dictionary (right)

The difference between the WERs of the CL-interpolated LM and the Trigger-interpolated LM in Table 7.1 are *not* statistically significant; the smallest p-value at which they would be significant is 0.4 for XINHUA and 0.7 for HUB-4NE.

7.7 Experiments with More Document-Aligned Corpora

Since a large document-aligned corpus is much easier to obtain than a sentence-aligned one, our cross-lingual trigger technique has a potential for further gains from larger bilingual training data. In order to assure that we can get a benefit with a document-aligned corpus over a sentence-aligned corpus, we experiment with a larger document-aligned corpus. Table 7.3 shows the size of two additional Chinese-English parallel corpora we used to build cross-lingual trigger pairs. Both of the foreign broadcast information service (FBIS) corpus and the Xinhua[4] corpus

[4]This is different from the one used for building a baseline LM in Section 6.1. Whereas the one used for building a baseline LM is a Chinese monolingual corpus, this is a Chinese-English

Corpus	No. of documents	No. of words	
		En	Ch
Xinhua	19K	4.1M	4.0M
FBIS	11K	10.5M	7.9M

Table 7.3: Additional document-aligned corpora

were released by linguistic data consortium (LDC) (Byrne et al., 2003). Remember that these are document-aligned corpora, which means they cannot be simply used for training of the IBM models (Brown et al., 1990, 1993). Following the steps to build cross-lingual lexical triggers, we first build the lexical triggers from each of Hong Kong news, Xinhua, and FBIS data, resulting in three different sets of trigger pairs. Then, we simply merge these trigger pair sets together to build the extended trigger pairs by using additional corpora.

As Table 7.4 shows, clearly, the trigger model with the additional corpora outperforms the trigger model only with the Hong Kong news corpus in both of the baselines. Furthermore, the trigger model with additional data has demonstrated better performance even than the case with the stochastic dictionaries noted as CL-interpolated—as we have expected. The difference in terms of WER from the case with the stochastic dictionaries is statistically significant in the HUB-4NE baseline, as with a p-value of 0.04. The difference in the XINHUA baseline are less significant with a p-value of 0.27, but obviously, we would be able to achieve statistically significant improvements by adding more data.

parallel corpus.

Language model	Perp	WER	CER	p-value
XINHUA trigram	426	49.9%	28.8%	–
CL-interpolated	346	48.8%	28.4%	< 0.001
Trigger-interpolated	367	49.1%	28.6%	0.004
w/ additional data	350	48.6%	28.5%	< 0.001
HUB-4NE trigram	1195	60.1%	44.1%	–
CL-interpolated	630	58.8%	43.1%	< 0.001
Trigger-interpolated	727	58.8%	43.3%	< 0.001
w/ additional data	657	58.3%	42.8%	< 0.001

Table 7.4: Perplexity and ASR performance of cross-lingual triggers from larger corpora

7.8 Summary

We note with some satisfaction that even simple trigger-pairs selected on the basis of mutual information are able to achieve perplexity and WER reductions comparable to a stochastic translation lexicon. Furthermore, by adding 14 million word document-aligned corpora to the cross-lingual trigger model, we have achieved better results than the stochastic lexicon-based (or IBM model-based) results. This suggests that the cross-lingual trigger model can be an alternative when there is little or no sentence-aligned corpus available, which is likely to be the case in resource-deficient languages. Finally, the potential benefit of the cross-lingual lexical trigger approach is obvious: it will lead us to build better translation lexicons by allowing us to use more data which is easy to obtain.

Chapter 8

Cross-Lingual Latent Semantic Analysis

Latent semantic analysis (LSA) is a standard automatic technique to extract corpus-based relations between words or documents. It was originally proposed to improve the effectiveness of (monolingual) information retrieval (IR) by performing comparisons of words in a reduced *semantic* space (or LSA space)—as opposed to performing exact word (term) matching as most traditional IR approaches do. By doing so, LSA has the advantage of avoiding the *synonymy* problem, in which different words can be used to describe the same semantic concept.

Furthermore, through the singular value decomposition (SVD) which will be explained in Section 8.2, LSA finds the optimal projection of the original word document frequency matrix into a low-dimensional space (Berry et al., 1995; Husbands et al., 2000). As a consequence, all terms or documents semantically related will remain salient in the projected LSA space, which leads us to efficiently find similar words, similar documents, or similar documents to given a word or vice

versa.

CLIR is a special case of the IR problem in which queries and documents are written in different languages. Since there is (almost[1]) no common word between queries and documents, word matching-based methods cannot be directly used for the CLIR problem. To tackle this problem in general, one may first translate words in one language to the other using a translation dictionary and then perform exact word matching in the same language space. Unlike the exact word matching-based methods, however, cross-lingual LSA doesn't require any explicit translation dictionary or knowledge since the comparisons are made in the reduced semantic space, and therefore, it provides us a way to do CLIR without resorting to any explicit translation dictionary. This inspires us to use LSA for CLIR in our cross-lingual language modeling approach. In addition, LSA allows us to compare the similarity of each word based on their usage across the whole document collection. Therefore, we can build a translation dictionary automatically without using a sentence-aligned parallel corpus. Thus, we employ LSA for translating English documents resulting from CLIR into Chinese which is our main language of interest (Kim and Khudanpur, 2004a,b). For completeness, we begin with a brief description of LSA.

8.1 Motivation

The trigger-based approach tries to establish relationships between words from training data, as shown in Chapter 7. There are, however, some limitations in the trigger-based approach. First, the words in a trigger pair should appear at least a certain number of times in the triggering word window and triggered word

[1]If the two language use same encodings and are similar to each other, some named entity words may be written as exactly same in the two languages, resulting in common words.

position setup—in order to be considered as a candidate. This is solely due to the practical reason: in other words, there would be too many candidates to be computed, otherwise. Typically, any pair that appears less than 5 or 10 times during the training is completely ignored from trigger pair candidates. Second, the trigger-based approach cannot capture transitivity relationships between trigger pairs. Suppose, for instance, we have a trigger pair (doctor $\rightarrow$ nurse[2]) and another pair (nurse $\rightarrow$ care). Then, it is not difficult to see that the pair (doctor $\rightarrow$ care) would be a good trigger pair; however, as long as the pair (doctor $\rightarrow$ care) didn't appear a certain number of times in the training set, they cannot be considered as a trigger pair. On the other hand, LSA ignores all structural information of each word position; instead, it takes a bag-of-words representation of documents. Consequently, any complex word pair selection procedure is not needed at all. Rather, the whole bag-of-words representation of documents adjusts the probabilities of the words in the predicted position.

A different motivation of LSA originates from an IR perspective. Most traditional IR models such as vector based models and Boolean models perform comparisons of queries and documents based on exact word matching. If a query and a document share many common words[3], it is likely that these two are topically or semantically relevant. Therefore, all documents retrieved by IR systems usually contain some words in the query. This is true most of the time: e.g., if a query contains words such as temperature, hot and rainy, then the query is likely to look for the documents about weather forecasts, and most documents related to weather forecasts are likely to contain those terms.

[2]Here, the arrow symbol shows the trigger pair relationship; i.e., a $\rightarrow$ b means that a triggers the appearance of b.

[3]Sharing content words rather than function words should be emphasized as the latter is little meaningful in terms of relevance. Typically, this can be achieved by appropriate term weighting schemes which give higher weights to content words.

Unfortunately however, this is not always true. Suppose a document contains words temperatures or rains. According to the IR methods based on exact word matches, there is no way to capture the variations of temperature and temperatures, or ones of rain and rainy.[4] In particular, the vector space model (Salton and McGill, 1986) which is one standard of IR models represents all queries and documents as a vector of words and measures the similarity of a query and a document by the inner product of each vector. In other words, it regards all words as *orthogonal* axes in the vector space—which means all words are completely independent. As a result, a query containing temperatures is regarded as completely independent of the documents containing temperature resulting in they are not relevant to each other.

In addition, most IR systems suffer from synonymy problems: the same concept can be expressed as different words. For example, let's consider the pair of words, car and automobile. Even though they are different words, the meaning of those words is identical or similar. And there exist extensive pairs of synonyms in our natural languages. Here are some more examples of synonyms in English: (baby, infant); (sick, ill); (cat, feline); and (smart, intelligent). If a query contains one word of the pairs and a document contains the other word of the pair, the IR models will regard them as not relevant even though they represent the same concept.

LSA is an alternative approach to alleviate the problems in traditional IR models based on exact word matching. Rather than performing direct word matching, LSA tries to find some underlying or *latent* structure in the pattern of word usage

[4]There is a reasonable approach to solve this problem—based on matches between *word stems*. Namely, they stem the word from temperatures into temperatur and from rainy or rains into rain ignoring all suffix variations. However, this method causes another problem making completely different words to be same after stemming. E.g., it stems both of university and universe into univers.

across documents. Through an analysis of the associations among words and documents, LSA produces a representation in which words that are used in similar contexts will be more semantically associated. Hence, it provides a nice solution for the synonymy problem; while it does a partial solution for the polysemy problem which is another significant problem in IR. Polysemys are the words which may be used for different meanings although they look same. For example, the word bank can be used for different meanings: 1) a place to save money, 2) the slope of land adjoining a body of water, 3) the lateral inward tilting, as of an aircraft, etc. Since the meaning of a word can be conditioned on contexts, LSA helps to decide which of the possible meanings has been used by looking at the context words. However, remember that each word is represented as a single point in a vector space. Therefore, the position of each word will be represented as a weighted average of all of its possible meanings. It is therefore possible that the weighted average corresponds to a completely different position to any of its meanings (Deerwester et al., 1990).

Prior to describing further details of LSA, we give a brief introduction to the underlying technologies of LSA—QR factorization and SVD.

8.2 QR Factorization and Singular Vector Decomposition

For the sake of illustration, we continue with the description of the QR factorization which is an orthogonal triangular factorization. In other words, any M-by-N matrix A ($A \in \mathbb{R}^{M \times N}$) can be decomposed or factorized into an orthogonal matrix $Q \in \mathbb{R}^{M \times M}$ and an upper triangular matrix $R \in \mathbb{R}^{M \times N}$ such that $A = QR$. A square matrix $Q \in \mathbb{R}^{M \times M}$ is called *orthogonal* if $Q\,Q^T = Q^T\,Q = I \in \mathbb{R}^{M \times M}$.

Notice that the QR factorization is similar to the SVD in the sense that it also provides a means of rank reduction of a matrix, but simpler than the SVD. Rank reduction is useful for many practical problems such as large scale IR, data compression and image processing as it enables to approximate a large (and sparse) matrix with a minimal loss of information. See Golub and Loan (1996) for more details about how to compute Q and R.

Here we show a simple example of the QR factorization. Suppose we have a 4-by-3 matrix A as following.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}. \tag{8.1}$$

Then by the QR factorization, the matrix can be decomposed as $A = QR$ where

$$Q = \left[\begin{array}{cc|cc} -0.0776 & -0.8331 & 0.5444 & 0.0605 \\ -0.3105 & -0.4512 & -0.7709 & 0.3251 \\ -0.5433 & -0.0694 & -0.0913 & -0.8317 \\ -0.7762 & 0.3124 & 0.3178 & 0.4461 \end{array}\right], \tag{8.2}$$

$$R = \left[\begin{array}{ccc} -12.8841 & -14.5916 & -16.2992 \\ \hline 0 & -1.0413 & -2.0826 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]. \tag{8.3}$$

Notice that R is upper triangular. Furthermore, this can be partitioned as

$$A = \left[\begin{array}{c|c} Q_A & Q_{\bar{A}} \end{array}\right] \begin{bmatrix} R_A \\ - \\ 0 \end{bmatrix} \tag{8.4}$$

$$= Q_A R_A + Q_{\bar{A}} \cdot 0 = Q_A R_A \,. \tag{8.5}$$

As shown above, $Q_{\bar{A}}$ doesn't contribute to the original matrix A and therefore, the ranks of A, R, and R_A are equal. Notice the diagonal entries of R which shows that the rank of R is 2. By partitioning the matrices Q and R based on the the diagonal entries of R, we can get a compact representation of the original matrix.

Furthermore, it allows us to approximate the original matrix. That is, as the diagonal entries of the matrix R are sorted from the upper left corner to the lower right corner, we can set all entries other than top k-by-k entries to zero, and by doing so, we can get rank-k approximation of the original matrix. In our example, we can get the rank-1 approximation, A', by setting the second row of R to zero and then multiplying by Q.

$$A' = \begin{bmatrix} 1.0000 & 1.1325 & 1.2651 \\ 4.0000 & 4.5301 & 5.0603 \\ 7.0000 & 7.9277 & 8.8554 \\ 10.0000 & 11.3253 & 12.6506 \end{bmatrix} . \tag{8.6}$$

Similarly, we can apply the SVD to any matrix such that $A = USV^T$ where $A \in \mathbb{R}^{M \times N}, U \in \mathbb{R}^{M \times M}, V \in \mathbb{R}^{N \times N}$, and $S \in \mathbb{R}^{M \times N}$. Here U, V are orthogonal matrices and S is a diagonal matrix. In our example, the matrix A in (8.1) can

be decomposed as

$$U = \begin{bmatrix} -0.1409 & 0.8247 & 0.5456 & -0.0478 \\ -0.3439 & 0.4263 & -0.6919 & 0.4704 \\ -0.5470 & 0.0278 & -0.2531 & -0.7975 \\ -0.7501 & -0.3706 & 0.3994 & 0.3748 \end{bmatrix} \tag{8.7}$$

$$S = \begin{bmatrix} 25.4624 & 0 & 0 \\ 0 & 1.2907 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8.8}$$

$$V = \begin{bmatrix} -0.5045 & -0.7608 & -0.4082 \\ -0.5745 & -0.0571 & 0.8165 \\ -0.6445 & 0.6465 & -0.4082 \end{bmatrix} \tag{8.9}$$

Notice the diagonal entries in S which implies that the rank of the original matrix A is 2—same as we have seen in the QR factorization. Like the QR factorization, the SVD provides us a means for the rank-k approximation by setting all diagonal entries to zero whose columns and rows are greater than k. In our example, we can get rank-1 approximation using the SVD resulting the matrix A''.

$$A'' = \begin{bmatrix} 1.8098 & 2.0608 & 2.3119 \\ 4.4185 & 5.0314 & 5.6443 \\ 7.0273 & 8.0020 & 8.9768 \\ 9.6361 & 10.9727 & 12.3093 \end{bmatrix} \tag{8.10}$$

It is worthwhile noting the similarities and differences between the QR factorization and the SVD. Both methods reveal the rank of the original matrix: the

number of nonzero diagonal entries in the matrix R in the QR factorization and S in the SVD. In addition, both provide the way for the rank-k approximation; on the other hand, the distance between the original matrix A and the rank-k approximated matrix, in terms of the *Frobenius norm*, is minimized by using the SVD (Berry et al., 1999; Hofmann, 2001). In other words,

$$A'' = \underset{\bar{A}:rank(\bar{A})=k}{\arg\min} ||A - \bar{A}||_F \tag{8.11}$$

where the Frobenius norm ($||\cdot||_F$) is defined as follows.

$$||A||_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^2} \tag{8.12}$$

where a_{ij} refers to the i^{th} row, j^{th} column entry in $A \in \mathbb{R}^{M\times N}$.

8.3 Previous Work on Latent Semantic Analysis

LSA is a powerful tool which can be applied to many problems such as digital image processing, information retrieval, language modeling, and word/document clustering. In particular, we review previous work of LSA that has been applied to natural language processing: information retrieval and language modeling.

8.3.1 Latent Semantic Analysis for Information Retrieval

One of the earliest uses of LSA for information retrieval (IR) can be found in Deerwester et al. (1990). It contains detailed mathematical descriptions with examples of LSA. They first built a word document frequency matrix from monolingual doc-

ument collections. Next, SVD which has been described in Section 8.2 was applied resulting in three matrices, U, V, and S. Finally all queries were projected into the LSA space to perform query document comparisons[5]. They compared the performance of LSA-based IR to that of some other IR systems such as the term-matching based IR system, and the SMART IR system (Salton and Lesk, 1965) in a monolingual IR experimental setup. According to their experimental results, LSA performed better than or equal to any other methods in all of their comparisons. They also describe how LSA can be used for term-to-term comparisons, document-to-document comparisons and term-to-document comparisons.

LSA has also been successfully applied to CLIR by Dumais et al. (1997, 1996); Landauer and Littman (1990), and they showed that CLIR can be performed without using any translation in LSA. In particular, Dumais et al. (1997) conducted a comparative study of vector-based IR and cross-lingual latent semantic analysis (CL-LSA) in a CLIR task via a cross-language mate retrieval experiment. Starting from a French-English parallel corpus, they divided the parallel corpus into training and test data, used training data to build SVD matrices, U, S and V, and measured the performance of vector-based IR and CL-LSA: how well each IR system finds the corresponding mate (aligned document) in the other language for a given test document in one language. Notice here that neither of the two systems used any translation. According to their experiments, CL-LSA shows superior results to vector-based IR (98.4% vs. 48.5% on average in terms of accuracy). The accuracy of 48.5% in vector-based IR may be surprising since this is CLIR without any translation, but notice that there may be many common words such as named entities between two languages (French and English). In order to remove the possibility of common words in two language data, they

[5]Details of this step are similar to the ones which will be shown shortly when we describe cross-lingual LSA in Section 8.5.

also converted all the words so that there is no common word (or overlap), and repeated the experiment. Still, CL-LSA performed consistently well showing an accuracy of 98.9%. They also conducted some LSA experiments after translating one language into the other using a publicly available machine translation system, but they didn't directly compare the performance of vector-based IR after machine translation with LSA after machine translation.

8.3.2 Latent Semantic Analysis for Language Modeling

LSA has been successfully used for language modeling by Bellegarda (1998, 2000a,b); Coccaro and Jurafsky (1998), and this section gives a summary of Bellegarda's more comprehensive work. Language modeling is a task of assigning a probability to a word given some context; how can LSA be used for language modeling? As mentioned earlier, LSA can be used for term-to-term comparisons, document--to-document comparisons, and term-to-document comparisons. This is possible since either a row (which corresponds to a word[6]) or a column (which corresponds to a document) may be projected into a same reduced dimensional LSA space. In other words, any word or document in the original matrix can be projected into a point in the same reduced LSA space regardless of whether it is a document, or a word, which enables us to compare two words, two documents, or a document and a word.

Keeping in mind that any term (word) or document can be compared in the LSA space, what is needed for language modeling is a way to build a word document frequency matrix for LSA from language model training data[7] and to use it for testing. Of course, some weighting scheme may need to be applied prior to

[6]This will be described later in Section 8.4

[7]Here, we assume that document boundaries of the language model training data are already marked.

building the matrix. Details of the weighting scheme may be found in Bellegarda (1998, 2000a,b). Once the word document frequency matrix has been constructed, SVD decomposes the matrix and forms an LSA space. Then, in language model testing, one can build a pseudo document from a bag of words representation of the context (all the words preceding the predicted word in language modeling). This pseudo document (which was constructed from the context) can then be projected into the LSA space. Since the pseudo context document has been projected into the LSA space, one can find similar words to the context document by measuring (cosine) similarity which may be viewed as a document-to-term comparison. According to Bellegarda (1998, 2000a,b), the words close to the pseudo context document should be assigned a higher probability, which seems to be intuitively reasonable. Notice that only relevant content words will be assigned high probabilities while most function words will be assigned low probabilities, since the function words are not close to any specific contexts. This behavior, however, is exactly the opposite of the one from conventional N-gram language models, which led him to integrate the LSA based language model with an N-gram language model. His LSA based language modeling work can be summarized as follows.

1. Construct a word document frequency matrix W from training data. Local weighting and/or global weighting scheme may be applied before constructing the matrix.

2. Compute SVD of W as

$$W \approx \hat{W} = USV^T . \tag{8.13}$$

3. Estimate the language model probability of the predicted word w_n given a

context $\tilde{d}_{n-1} \equiv w_1, w_2, \cdots, w_{n-1}$

$$P_{LSA}(w_n|\tilde{d}_{n-1}) = \frac{P(w_n, \tilde{d}_{n-1})}{\sum_{w_i \in \mathcal{V}} P(w_i, \tilde{d}_{n-1})}, \tag{8.14}$$

where $\mathcal{V}$ is the vocabulary. Also, $P(w_n, \tilde{d}_{n-1})$ can be computed to reflect the similarity of the word w_n and the context $\tilde{d}_{n-1}$ in the LSA space which were constructed above. One possible way to achieve this is to take a power of the similarity of the two as in (8.15) and proposed by Coccaro and Jurafsky (1998).

$$P_{\text{LSA}}(w_n|\tilde{d}_{n-1}) = \frac{Sim(w_n, \tilde{d}_{n-1})^{\gamma}}{\sum_{w_i \in \mathcal{V}} Sim(w_i, \tilde{d}_{n-1})^{\gamma}} \tag{8.15}$$

4. Integrate the LSA based language model with N-grams (Bellegarda, 1998).

$$P(w_n|H_{n-1}) = \frac{P(w_n|w_{n-1}w_{n-2}\cdots w_{n-N+1})P_{LSA}(\tilde{d}_{n-1}|w_n)}{\sum_{w_i \in \mathcal{V}} P(w_i|w_{n-1}w_{n-2}\cdots w_{n-k+1})P_{LSA}(\tilde{d}_{n-1}|w_i)} \tag{8.16}$$

where H_{n-1} is the admissible LSA history. Notice that if $P_{LSA}(\tilde{d}_{n-1}|w_n)$ is viewed as a prior probability, then (8.16) is the Bayesian estimation of the N-gram probability using a prior distribution from LSA.

Bellegarda (1998, 2000a,b) also experimented with class based language models via word clustering since word-to-word comparisons can be easily made in the LSA space. In other words, based on the similarity scores of any pair of words, word clusters can be constructed by the K means clustering algorithm (Duda and Hart, 1974). He achieved a 25% reduction in perplexity over the standard bigram language model and a 15% reduction in average WER. By using a class-based LSA language model, he achieved a 16% reduction in average WER over the trigram language model.

8.4 Cross-Lingual Latent Semantic Analysis

In this section, we describe the details of LSA with our cross-lingual setup. We begin with a document-aligned Chinese-English corpus, which is already available. The first step for cross-lingual latent semantic analysis (CL-LSA) is to represent the corpus as a word-document *co-occurrence frequency matrix* W. That is, we extract word unigrams from each Chinese-English document pair and fill the unigram entries into one column vector of the matrix W regarding the unigrams as a column vector. By repeating this process for all document pairs, we can build a matrix W. In other words, each row represents a word in one of the two languages, and each column represents a document pair. If the size of the Chinese plus English vocabularies $|\mathcal{C} \cup \mathcal{E}|$ is M, and the corpus has N document-pairs, then W is an M-by-N matrix. Here, each element w_{ij} of W contains the frequency (count) of the i-th word in the j-th document-pair. Next, each row of W is weighted by some function which deemphasizes frequent (function) words in either language, such as the inverse of the number of documents in which the word appears. Next, SVD is performed on W and, for some $R \ll \min\{M, N\}$, we approximate W by its largest R singular values (rank-R approximation) and the corresponding singular vectors as

$$W \approx U \times S \times V^T, \tag{8.17}$$

where columns of U and V are orthogonal left- and right-singular vectors of W, and S is a diagonal matrix whose entries $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_R$ are the corresponding singular values (Berry et al., 1995). If R is same as the rank of W, the multiplication (8.17) will be exactly same as W; On the other hand, if R is less than the true rank of W, then (8.17) is the least-square approximation of W among

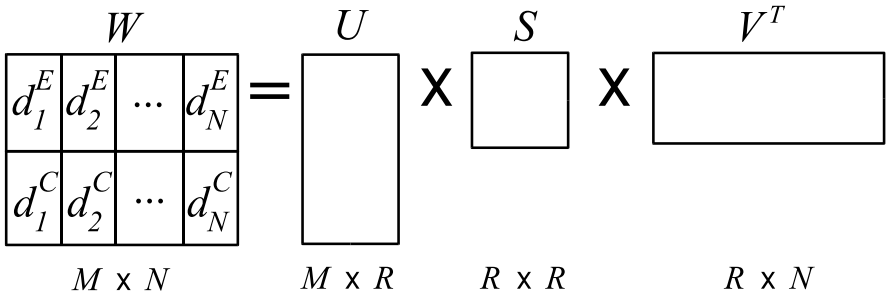

Figure 8.1: SVD of a word-document matrix for CL-LSA

all rank-R matrices. This implies that we can make any rank-R approximation to the original matrix W by adjusting the R values.

Figure 8.1 shows the matrices after the SVD. Even though the dimensions after the SVD look different from the ones shown in (8.7)–(8.9)—U, V are not square matrices and S is a square matrix this time—we can convert the latter into the former using the matrix partition as shown in Section 8.2 and in fact, both result in the same original matrix after multiplying out. If we give a small value of R, we will get a compact representation ignoring many details of the original.

In view of this rank-R approximation, the j-th column W_{*j} of W, or the document-pair d_j^E and d_j^C, is a linear combination of the columns of $U \times S$, the *weights* for the linear combination being provided by the j-th column of V^T. Equivalently the *projection*

$$V_{*j}^T \approx S^{-1} \times U^T \times W_{*j} \tag{8.18}$$

of the j-th column of W on to the *basis* formed by the column-vectors of $U \times S$ provides an R-dimensional representation of the pair d_j^E and d_j^C. Similarly, projecting the i-th row of W, which represents the distribution of the i-th word, onto the basis formed by row-vectors of $S \times V^T$ provides an R-dimensional representation

of words.

It is clear that a Chinese-English translation-pair (e, c) that has been used consistently in the paired-documents will yield two similar row vectors in W, and hence their R-dimensional representations will be very close to each other. An elegant consequence of CL-LSA is that other topically- or semantically-related words also end up having very similar R-dimensional representations, as do documents on those topics. This common R-dimensional representation of words and documents has therefore come to be called *semantic space*.

8.5 LSA for Cross-Language Information Retrieval

One nice property of LSA for our purpose is that it does not use direct word matching between queries and documents for IR. Given the input word document frequency matrix, it first transforms the matrix using SVD and then permits us to perform the comparison between queries and documents in the transformed space. Applying this idea to our cross-lingual language modeling setup, it provides us a way to measure the similarity between queries and documents even though they don't share any common words. That is, we can compare Chinese queries with English documents without using a translation lexicon $P_T(e|c)$ as required by (4.1) in Section 4.2. Here we first use LSA for CLIR; this is similar to the previous work by Dumais et al. (1997, 1996); Landauer and Littman (1990).

Assume that a modest-sized document-aligned Chinese-English corpus has been used to construct the matrices U, S and V of (8.17). Next, an additional corpus of English documents is given, a Chinese query is provided, and our task is to find a document from the English corpus which is topically related to the

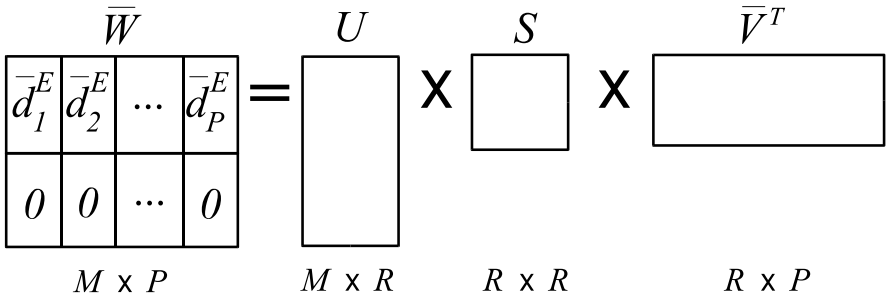

Figure 8.2: Folding-in a monolingual corpus into LSA

Chinese query.

We construct a word-document matrix $\overline{W}$ using the English corpus, much as we did earlier with the document aligned corpus. All rows corresponding to the Chinese vocabulary items have zeros in this matrix, as illustrated in Figure 8.2. Yet, we proceed to project these documents d_j^E into the semantic space as suggested by (8.18), and obtain the R-dimensional representations $\overline{V}^T$ for these documents. If we have P English documents in our collection, we will get an R-by-P dimensional matrix, $\overline{V}^T$. Given an additional Chinese query vector, d_i^C, it will have zeros in the upper half which corresponds to English words this time, since it is a Chinese query. Nevertheless, we can again project the Chinese query d_i^C into the R-dimensional space, similar to (8.18). Notice that this results in an R dimensional column vector—just as the English documents correspond to R dimensional column vectors. As both the Chinese query and English documents are now represented as R dimensional vectors, it enables us to compare them without using any translation dictionary. Finally, we use the cosine similarity (4.4) between the query- and document-representations to find the English document d_i^E which is most similar to the Chinese query d_i^C.

8.6 LSA-Derived Translation Probabilities

We also use the CL-LSA framework to construct the translation model $P_T(c|e)$ of (4.5). In the matrix W of (8.17), each word is represented as a row no matter whether it is English or Chinese. As discussed earlier, projecting these words into R-dimensional space yields rows of U, and the semantic similarity of an English word e and a Chinese word c may be measured by the cosine-similarity of their R-dimensional representation. We extend this notion and construct a word-word translation model, $\forall\, c \in \mathcal{C}$, $\forall\, e \in \mathcal{E}$, as

$$P_{\text{LSA}}(c|e) = \frac{Sim(c,e)^{\gamma}}{\sum_{c' \in \mathcal{C}} Sim(c',e)^{\gamma}} \tag{8.19}$$

where $\gamma \gg 1$ as suggested in Coccaro and Jurafsky (1998). Having estimated (8.19), we build our LSA-based LM analogous to (4.5) and (4.6).

$$P_{\text{LSA}-\text{unigram}}(c|d_i^E) = \sum_{e \in \mathcal{E}} P_{\text{LSA}}(c|e)\hat{P}(e|d_i^E) \tag{8.20}$$

$$\begin{aligned} & P_{\text{LSA}-\text{interpolated}}(c_k|c_{k-1}, c_{k-2}, d_i^E) \\ &= \lambda_{d_i^E} P_{\text{LSA}-\text{unigram}}(c_k|d_i^E) + (1 - \lambda_{d_i^E}) P(c_k|c_{k-1}, c_{k-2}). \end{aligned} \tag{8.21}$$

8.7 Cross-Lingual Mate Retrieval Experiments

Before applying LSA for our cross-lingual language modeling setup, we first replicate the CLIR experiment using LSA which was conducted by Dumais et al. (1996). Our parallel data is the Hong Kong News Chinese-English collection as

Model	Accuracy
Vector-based IR	92.4 %
CL-LSA ($R = 76$)	70.9%
CL-LSA ($R = 207$)	80.4%
CL-LSA ($R = 447$)	87.6%
CL-LSA ($R = 693$)	90.2%

Table 8.1: Cross-lingual mate retrieval results

shown in Table 5.1. After diving the data into the training set and the test set as in the table, we apply SVD to the word document matrix constructed from the training set. We then project the English test set documents into the low R-dimensional space. For each of the Chinese test document, again we project it into the R-dimensional space, find the most similar one among English test documents by comparing them in the R-dimensional space. We use cosine similarity to measure the similarity between each Chinese and English document. Finally we measure how accurately CL-LSA selects the corresponding English mate. For vector-based IR, rather than doing CLIR as is where there are almost no common words, we use statistical machine translation before applying vector-based IR. First, we get the *translated* bag-of-words English document by using the GIZA++ translation dictionary, $P_T(e|c)$ as described in Section 5.2.2, from a given Chinese test document. Once we have a *translated* bag-of-words English document (from a Chinese query document), we can apply standard vector-based IR from then on. In other words, we find the most similar documents from the English test set again by measuring cosine similarity.

Table 8.1 shows the accuracy of the cross-lingual mate retrieval experiment. Contrary to the results in Dumais et al. (1997), but not surprisingly, our vector-based IR performs reasonably well giving 92.4% of accuracy even though we are testing on much more difficult language pairs (Chinese-English) than their lan-

guage pairs (French-English). This is purely due to the use of fairly well tuned translation dictionary $P_T(e|c)$ before applying vector-based IR.

Our first CL-LSA with a small approximation rank ($R = 76$) performs far below vector-based IR. Assuming that the approximation rank of 76 is too small for our database, we test the accuracy by changing the approximation rank of SVD. As we increase the approximation rank, the accuracy increases and we get a similar performance to vector-based IR when R is 693[8]. From this experiment, we can conclude that even without using any translation dictionary, CL-LSA can achieve a performance similar to that of cross-lingual vector-based IR after machine translation.

8.8 CL-LSA Automatic Speech Recognition Experiments

Following the procedures explained in Section 8.5 for CLIR and Section 8.6 for building translation lexicons, we experiment with CL-LSA for our cross-lingual language modeling ASR experiments. We perform SVD using document-pairs from the Hong Kong News parallel corpus and retain about $R = 700$ singular values in S and the corresponding singular vectors U and V. This provides the basis for the word-to-word similarity computation of (8.19). We project the NAB-TDT corpus in Section 6.1 on to this space via (8.18), and use the resulting representations $\overline{V}$ for the English documents. This provides the basis for the CLIR step used to match a Mandarin story being transcribed by the ASR system with English documents in NAB-TDT.

[8]Due to the memory limitation, 693 was the maximum of our system and we were not able to go further.

Language Model	Perp	WER	CER	p-value
XINHUA Trigram	426	49.9%	28.8%	–
LSA-interpolated	364	49.3%	28.9%	0.043
Trig+LSA-intpl	351	49.0%	28.7%	0.002
CL-interpolated	346	48.8%	28.4%	< 0.001
HUB-4NE Trigram	1195	60.1%	44.1%	–
LSA-interpolated	695	58.6%	43.1%	$<$0.001
Trig+LSA-intpl	686	58.7%	43.2%	$<$0.001
CL-interpolated	630	58.8%	43.1%	< 0.001

Table 8.2: Word perplexity and ASR WER comparisons

We follow the same procedures of 300-best list rescoring as in Section 6.3. For each test story d_i^C, we then perform CLIR using the first pass ASR output to choose the most similar English documents d_i^E's from NAB-TDT. Then we create the cross-lingual unigram of (8.20). We next find the interpolation weight $\lambda_{d_i^E}$ in (8.21) that maximizes the likelihood of the 1-best hypotheses of all test utterances in a story obtained from the first ASR pass. We finally rescore the 300-best lists using the *LSA-interpolated* LM in (8.21), and report results[9] in Table 8.2.

For comparison, we also show the *CL-interpolated* results of Section 6.3 which use a superior translation lexicon derived from a sentence-aligned corpus, both for CLIR to find d_i^E and instead of $P_{\text{LSA}}(c|e)$ in (8.20). Finally, we note that the technique of cross-lingual lexical triggers reported in Section 7.6 also assumes only a document-aligned corpus as done here, and an interpolation of their model with ours does not require any additional resources. We perform this interpolation and report the results as *Trig+LSA-intpl* in Table 8.2.

As Table 8.2 shows, the LSA-interpolated model shows a significant reduction in both perplexity (15-42%) and WER (0.6-1.5% absolute) over the baseline trigram model both when a moderate amount of LM training text is available (XIN-

[9]All p-values are based on the comparisons over the trigram baseline in Section 6.2.

HUA) and when it is really scarce (HUB-4NE). It performs only slightly worse than the CL-interpolated model, which requires the more expensive sentence-aligned corpus. Finally, the interpolation of our LSA-based model and the trigger-based model which is noted as Trig+LSA-intpl brings further gains, removing the remaining gap from the CL-interpolated model: the p-values of the differences between CL-interpolated and Trig+LSA-intpl models are 0.58 for XINHUA and 0.79 for HUB-4NE.

8.9 Summary

We have demonstrated that latent semantic analysis can be successfully incorporated into our cross-lingual language model adaptation. Given a comparable corpus which is document-level aligned, we can perform CLIR and build the translation lexicons via latent semantic analysis. We have demonstrated a significant reduction in perplexity and WER over a trigram model. Performance statistically indistinguishable from a CL-interpolated model predicated on good MT capabilities can be achieved by our methods.

Chapter 9

Statistical Machine Translation Experiments

Statistical modeling approaches rely heavily on the nature of training data—as it provides the only way to build (train) statistical models. In other words, regardless of its amount, there is no way to build an accurate statistical model if training data is not similar[1] to the test data in which we are interested. For example, having an extensive amount of data from newswire texts doesn't help much to recognize conversational speech.

In this chapter, we turn our attention to the data mismatch problem which is prevalent even in resource-rich languages. Especially, if a majority of training data significantly differ from the test data, the data mismatch problem becomes more serious and leads us to build suboptimal language models for the test data. Therefore, even if an extensive amount of training data is available, we need to make sure that our training data matches the test data. Since not all of training data would match the test data, a better approach is to first select only the

[1]This happens frequently in reality, and we call it the *data mismatch* problem.

data that is matched with the test data, and then use the selected data for language model adaptation. This chapter investigates monolingual language model adaptation—where the source of adaptation comes from the same language as the language of interest, English—to solve the data mismatch problem. Besides, monolingual language model adaptation is applied to a different application, statistical machine translation. For the sake of completeness, we begin this chapter with a brief introduction to statistical machine translation which has been intensively studied since the original work by Brown et al. (1990, 1993), often called the *IBM models*. Another good introduction to statistical machine translation is available from Knight (1999).

9.1 Statistical Machine Translation: IBM models

The goal of statistical machine translation (SMT) is to find the most likely word sequence $e_1^I = e_1 e_2 \cdots e_I$ in a target language, say English, given a source language word sequence $c_1^J = c_1 c_2 \cdots c_J$, say in Chinese.

$$\hat{e}_1^I = \arg\max_{e_1^I} P(e_1^I | c_1^J) \quad . \tag{9.1}$$

By applying Bayes' rule, (9.1) may be rewritten as

$$P(e_1^I | c_1^J) = \frac{P(e_1^I) P(c_1^J | e_1^I)}{P(c_1^J)}, \tag{9.2}$$

and since c_1^J is fixed—the source language word sequence is given—our problem can be decomposed into two parts, the translation modeling problem, and the

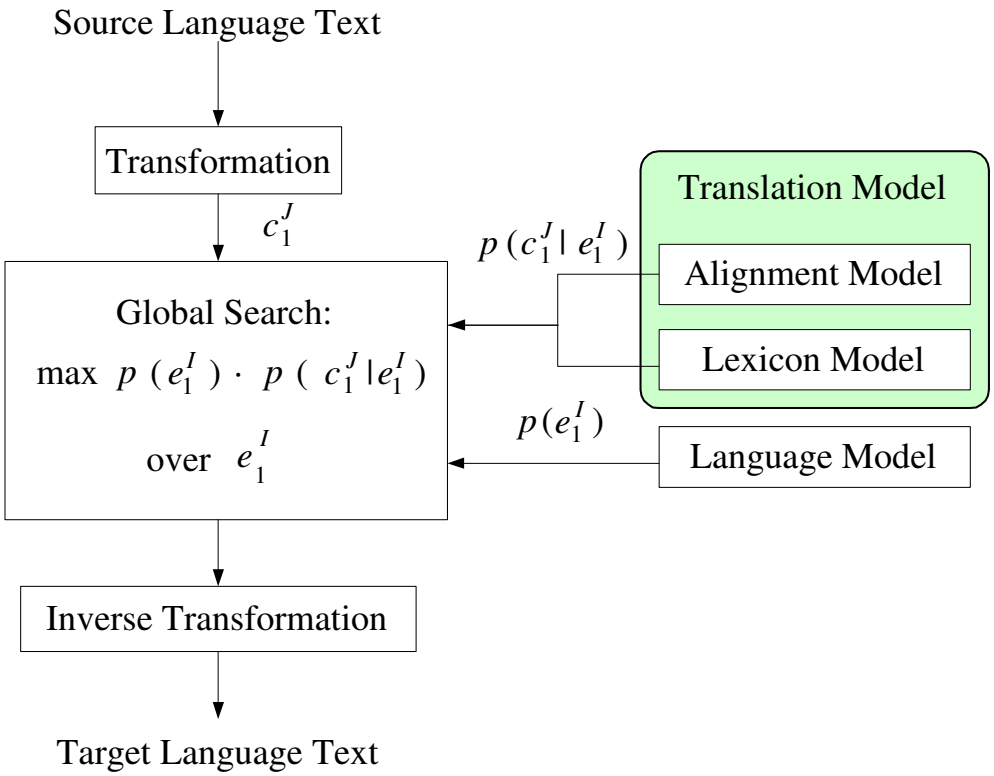

Figure 9.1: Statistical machine translation architecture (Vogel et al., 2000)

language modeling problem.

$$\hat{e}_1^I = \arg\max_{e_1^I} \underbrace{P(e_1^I)}_{\text{Language Model}} \times \underbrace{P(c_1^J|e_1^I)}_{\text{Translation Model}} . \tag{9.3}$$

Just as the language model plays an important role in ASR, it is equally important in SMT: 1) it reduces the search space in the target language by suppressing unlikely hypotheses in the target language and 2) it resolves lexical ambiguities by providing contextual information. In other words, a language model in SMT takes care of the grammaticality of segments (sentences) in the target language so that the translation model doesn't have to.

Figure 9.1 shows the overall architecture of the conventional SMT approach.

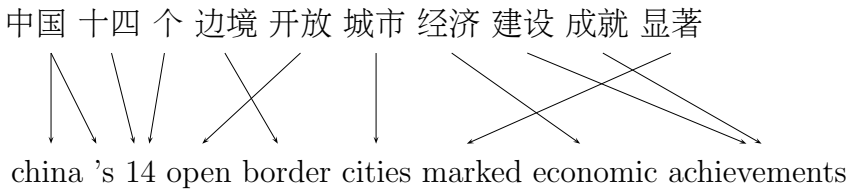

Figure 9.2: An alignment example between Chinese and English (Och et al., 2003)

Notice that the translation model consists of an alignment model and a lexicon model. Recall that the language model probability of (9.1) can be easily decomposed into a product of N-gram probabilities using the Markov assumption in Chapter 2. Similarly, we may be able to decompose a translation model probability into several parts. In order to make this decomposition possible, however, we need more detailed information such as a word-to-word correspondence between the segments in two languages. To this end, a new hidden variable, alignment, has been introduced.

Figure 9.2 shows an example of the alignment between an English sentence and a Chinese sentence. An alignment consists of a number of word-to-word links (arrows) and each link specifies which word in one language corresponds (or is translated) to which word in the other language. Notice that the alignment is not necessarily a one-to-one mapping. One word in the source language may correspond to several words in the target language, or it may not have any corresponding word in the target language. Notice also that the alignment represents the relationship of word positions, not the relationship of words themselves. In other words, it only says the English word in position i corresponds to the Chinese word in position j (and vice versa), not which word does to which word. The alignment is represented as a set of mapping: $j \rightarrow i = a_j$, which means that the word c_j in position j corresponds to the word e_i in position $i\,(= a_j)$. Following

the notation above, we can represent any alignment sequence as $a_1^J = a_1 a_2 \cdots a_J$. Once an alignment is determined, its probability should be estimated given the word sequence in the source language, $P(a_1^J|e_1^I)$.

Next, the remaining probability in the translation model is the probability of actual word correspondence between the words in two languages—lexicon model probability. Since this probability estimation requires an alignment, it is represented as $P(c_1^J|a_1^J, e_1^I)$.

Finally, the two probabilities—the alignment probability and the lexicon model probability—should be multiplied together to generate the total probability of the word sequence in the target language given the one in the source language. Many alignments, however, are possible for the given word sequences in two languages, and we don't know which alignment is correct. This leads us to consider all possible alignments between the sentence pair in two languages. In other words, the translation model probability becomes

$$P(c_1^J|e_1^I) = \sum_{a_1^J} \underbrace{P(c_1^J|a_1^J, e_1^I)}_{\text{Lexicon Model}} \times \underbrace{P(a_1^J|e_1^I)}_{\text{Alignment Model}} . \tag{9.4}$$

The estimation of each model's probability can be made by the EM algorithm (Dempster et al., 1977) where further details can be found in Brown et al. (1990, 1993); Knight (1999). Basically, the algorithm begins with the assumption that the lexicon model has the uniform distribution, initially. Then, for each possible alignment from the given segment pair, it estimates the alignment probabilities based on the lexicon model. In other words, there exists a pair of words in each language which corresponds to one link (arrow) in an alignment, and the probability of an alignment can be estimated based on the translation model probability for the word pair. Next, since the alignment probability has been

estimated, the lexicon model probability can be obtained by collecting counts—which are fractional counts weighted by the alignment probabilities—of a pair of words in each language. Finally, this process is repeated until the lexicon model probability and the alignment model probability converge.

9.2 Translation Template Model

One weakness of the IBM models is that they are based on word translations and movements; therefore these models are poor for use in actual translation. To alleviate some of the problems caused by word-based translations, Och et al. (1999) proposed the alignment template translation model which instead is based on phrase-to-phrase translations. Another challenge in the IBM models is that it is not trivial to find out the best hypothesis during the machine translation decoding process due to the complexity of the underlying models, namely IBM model 1–5 (Brown et al., 1990, 1993). Recently, Kumar et al. (2004a) proposed the *translation template model* (TTM) which is a generative source-channel formulation for phrase-based translation constructed so that each step in the generative process can be implemented within a weighted finite state transducer (WFST) framework (Mohri, 1997; Mohri et al., 2000). We summarize this technique here for completeness, and interested readers are referred to Kumar et al. (2004a) for details.

The TTM defines a joint probability distribution over all steps needed to translate a source language segment into a target language segment. Figure 9.3 shows an example of the generative translation process in TTM. Following the notations from Kumar et al. (2004a), the naming of the source language and the target language is interchanged only in Section 9.2. Notice that Figure 9.3 is a

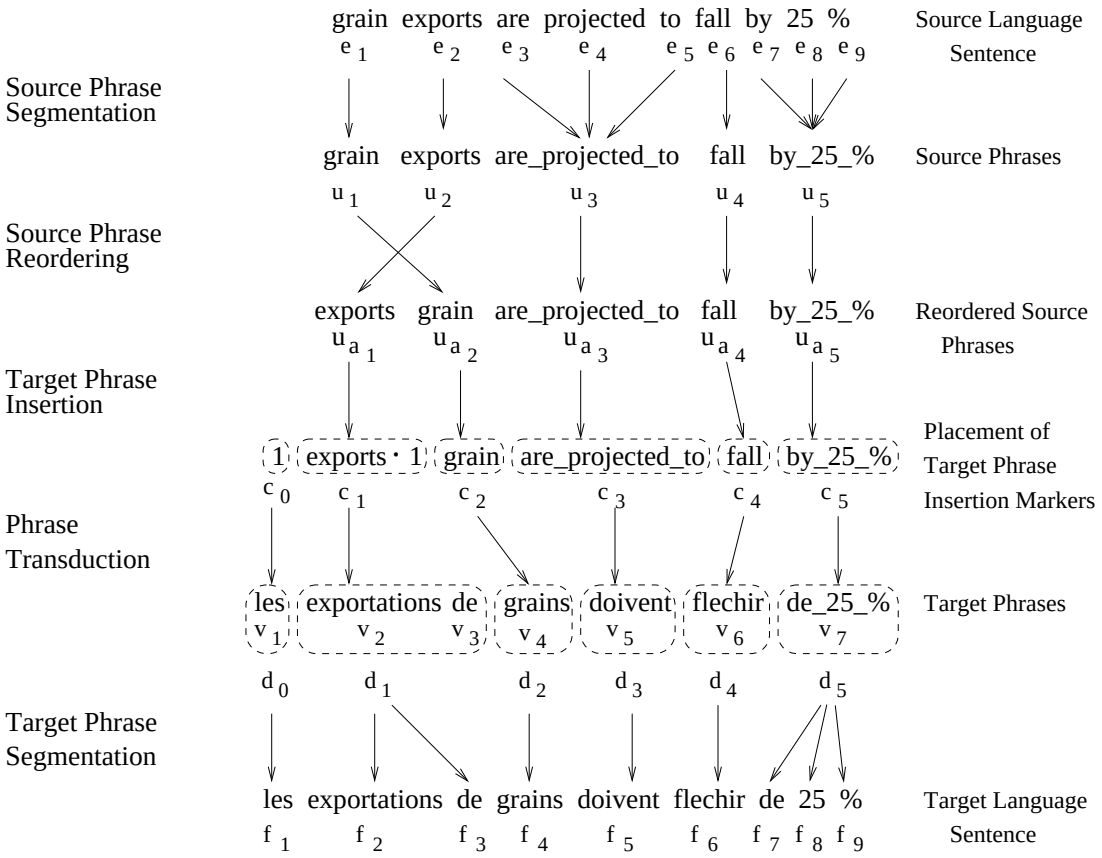

Figure 9.3: An example of the generative process underlying the TTM (Kumar et al., 2004a)

generative process according to the source-channel model, and the goal of MT is to find out the best hypothesis in the source language to the given target language segment. Here we briefly describe each component of the TTM.

Source Language Model Since an LM simply assigns a probability to any given word strings, this can be easily implemented by a weighted finite state acceptor (WFSA) (Mohri, 1997; Mohri et al., 2000).

Source Phrase Segmentation Model Given a sequence of source words, the goal of the source phrase segmentation model is to generate a sequence of source phrases. This model is based on the source phrase inventory from

the training set (see Kumar et al., 2004a).

Phrase Order Model This model reorders the source phrase sequence into the target language phrase sequence. This can be done in two steps: 1) permute the input phrase so that all reorderings are possible 2) assign alignment probabilities to each reordering (see Kumar and Byrne, 2003).

Target Phrase Insertion Model It is well-known that the number of words in the source language may not be identical to the number of words in the target language. Likewise, there is no guarantee that the number of phrases in the two languages are same; therefore, the target phrase insertion model is introduced to allow insertion of target language phrases, and avoid the necessity that the target language should contain the same number of phrases as the source language phrases (see Kumar et al., 2004a).

Phrase Transduction Model This model actually *translates* the source language phrases into the target language phrases and is the main component of the TTM. In a way similar to building the word lexicon model in the IBM models, it estimates probabilities from the phrase alignments of parallel texts.

Target Phrase Segmentation Model This is to enforce the constraint that words in the target language agree with the phrase sequence obtained by composing the models above.

Given all the machinery described so far, the final goal of machine translation is to find out the best hypothesis in the source language to the given a target

language sentence f_1^J—via *Maximum A Posteriori* (MAP) decoding as:

$$\{\hat{e}_1^{\hat{I}}, \hat{K}, \hat{u}_1^{\hat{K}}, \hat{a}_1^{\hat{K}}, \hat{c}_0^{\hat{K}}, \hat{d}_0^{\hat{K}}, \hat{v}_1^{\hat{R}}\} = \underset{e_1^I, K, u_1^K, a_1^K, c_0^K, d_0^K, v_1^R}{\operatorname{argmax}} P(K, u_1^K, a_1^K, c_0^k, d_0^K, v_1^R | f_1^J). \quad (9.5)$$

where $\hat{e}_1^I$ is the translation of f_1^J, and all other notations follow from Figure 9.3. In the WFST-TTM framework, this is done by finding the best path with the highest score in a translation lattice which, in turn, is obtained by composing the component WFSTs of the TTM (Kumar et al., 2004a).

Remark: In this chapter, we focus our attention on the first model component, namely the source language model, and all other model components are provided by Kumar et al. (2004b).

9.3 Statistical Machine Translation Evaluations

ASR performance is measured, as shown in Section 2.3, by WER with respect to the reference transcription. Unlike in ASR however, the evaluation of SMT is a difficult problem. Of course, we can rely on human judgments to evaluate SMT outputs, but human judgments are expensive and they are difficult to quantify. Therefore, the automatic evaluation of SMT outputs has been an important issue, and still there is no single standard measure (Akiba et al., 2001; Lin and Och, 2004; Melamed et al., 2003; NIST, 2002; Papineni et al., 2002). The main difficulty in the automatic evaluation of SMT lies in the fact that there is no single ground truth. In other words, there may be many correct translations for a source language input segment depending on word choice and word order.

Recently, Papineni et al. (2002) proposed the automatic SMT measure, BLEU (BiLingual Evaluation Understudy), aiming to approximate the human judgment.

Reference 1: the cat is on the mat
Reference 2: there is a cat on the mat
Candidate : the the the the the

Figure 9.4: Example of machine translation references and a candidate

Basically, the calculation of BLEU is based on N-gram precision of the SMT output with respect to the human reference translations of the input segment. Since it is almost always the case that there are many correct translations, more than one human reference translations are used. Given M human reference translations (typically $M = 4$), BLEU simply measures precision of the N-grams in the candidate translation output. For example, unigram precision is simply $\frac{C}{L}$ where L is the number of unigrams (words) in the candidate, and C is the number of unigrams in the candidate which are also seen in at least one of the reference translations. However, there is one problem with simple precision as shown in the example of Figure 9.4.

In the example, the candidate translation is definitely a bad translation regardless of the source language segment; however, simple unigram precision is $5/5 = 1.0$, which means it is a good translation. Therefore, BLEU modifies precision so that the C cannot be bigger than the maximum number of times an N-gram appears in any single reference translation. In the example, therefore $C = 2$ and the *modified* precision becomes $2/5 = 0.4$. Similarly, we can estimate modified higher-order N-gram precision[2].

After computing all modified N-gram precisions (N= $1, \cdots, 4$), the remaining question is how to combine them to obtain a single measure. Simple weighted

[2]Notice that *recall*, which is another good measure for IR, is not used for MT evaluation. Indeed, simply improving recall without considering precision in SMT generates worse translations. For instance, the cat is on the mat there is a is not a good translation even though recall is high.

linear average may not be a good solution as, usually, low-order N-gram precision is much higher than high-order one. In fact, as the order N increases, precision decays exponentially, and a better averaging scheme is to take an exponential decay into account; a weighted average of the logarithm of modified precisions, which is equivalent to using the geometric mean of modified N-gram precisions, is used.

Finally, there may be differences in a sentence length: i.e., the candidate translation should not be too long or too short compared to the reference translations. In fact, very long candidate translations are already penalized by precision as the long candidates with redundant N-grams will have lower precision. In other words, there is no need to penalize the long candidates. Therefore, BLEU penalizes short candidates by another mechanism: the brevity penalty (BP).

$$\mathrm{BP} = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} , \tag{9.6}$$

where c is the length of the candidate translation, and r is the effective reference corpus length. Finally, the BLEU score is defined as

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{K} w_n \log p_n\right) , \tag{9.7}$$

and typically $K = 4$ and $w_n = 1/K$ are used.

9.3.1 Bootstrap Resampling

Once we have obtained MT evaluation scores, say BLEU, from two or more MT systems, we can directly compare them based on their BLEU scores and say which

system works better than the others. This may, however, not be enough to say how confident we are about that statement. As described in Section 2.4, a statistical significance test may be needed to *confidently* say which is better as it gives us an evidence that the difference or improvement is *statistically significant*. Unlike in ASR, current MT evaluation is typically measured on a broad level such as a set of segments (sentences) rather than a sentence level, and the significance test measured on the sentence level cannot be used for MT. Therefore, another well known technique called *bootstrap resampling* has been applied to measure confidence intervals for MT[3] (Zhang et al., 2004).

The basic idea of bootstrap resampling works as follows. Suppose there are 919 segments in a test set, and we obtained a 28.5 BLEU score from the output of a baseline MT system. From the 919 hypothesis segments of the MT system, we build a new set of the hypotheses by sampling with replacement. Notice that a same segment may appear many times in the new set since this sampling is done with replacement. We repeat this process many times, say 1000, and obtain 1000 new sets. Whenever we select a hypothesis segment from the test set, we also select the corresponding reference segments[4] to build the set of references. Given the 1000 sets of the hypotheses and the references, we can measure 1000 BLEU scores. The median of the 1000 BLEU scores should be close to the original BLEU score, 28.5. Then, we can sort the 1000 BLEU scores of the sets, and take 25th and 975th scores from the sorted 1000 scores. Next, these two scores comprise a confidence interval at the level of 95%. Finally, if another MT system scores beyond that confidence interval, we can say that the system performs significantly different from the baseline system.

[3]The idea to use bootstrap resampling for measuring confidence intervals for MT scores was originally proposed by Franz Och. This was in fact adapted from the method to measure confidence intervals for ASR (Bisani and Ney, 2004).

[4]Notice that multiple references are typically available for one segment.

Remark: This bootstrap estimate has been implemented by Och et al. (2003) and we use this implementation in our analysis below.

9.4 Language Model Adaptation for Statistical Machine Translation

Our task is to build a Chinese to English translation system as shown in (9.3). Notice that the language model resides in the target language, English, which has extensive amounts of data. Hence, unlike the ASR experiments in Chapter 6, the LM doesn't seriously suffer from the data sparseness problem in this case; nevertheless, the data mismatch problem is prevalent since *not* all of training data would match the test data. In order to alleviate the data mismatch problem, we experiment with monolingual language model adaptation. Following our approach for ASR in Figure 4.1, we build a Chinese to English machine translation system as in Figure 9.5, which is based on monolingual language model adaptation. Notice the differences from Figure 4.1: 1) monolingual IR is used instead of CLIR, 2) translations from the retrieved English documents are no longer needed as our target language is also English, and consequently 3) higher order story-specific models are constructed instead of cross-lingual unigram language models. Story-specific likelihood-based optimization, however, is still used.

To begin with, we first find the best matching English documents out of an English text collection for given a Chinese test document. Finding topically matching documents is an information retrieval problem. Following our previous approach used for cross-lingual language modeling, we take the first pass translation outputs as the seed (query) for finding best matching documents through IR.

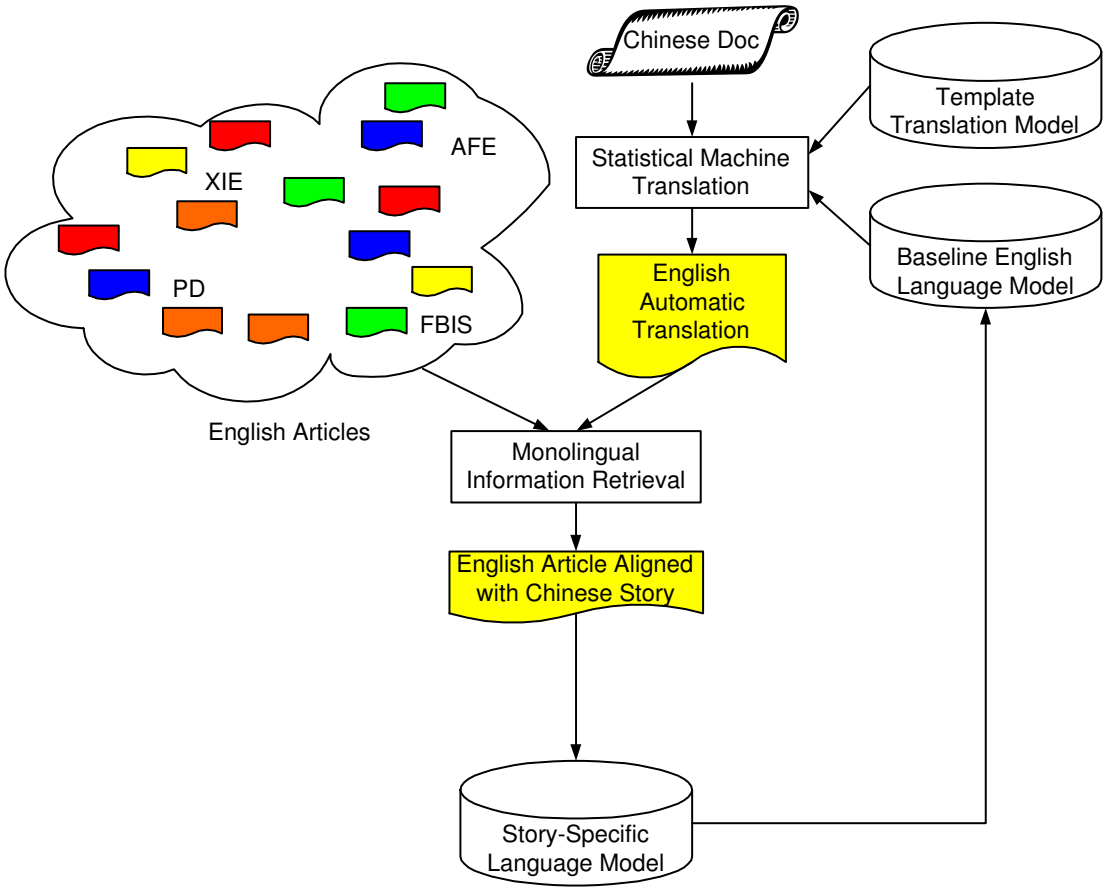

Figure 9.5: Story-specific monolingual LM adaptation for SMT

Here we use a state-of-the-art IR toolkit, LEMUR[5], which is publicly available. Given an English query document, which is built from the 1-best lists of the first-pass SMT outputs of all the segments in a Chinese document, LEMUR returns a ranked list of relevant documents, sorted by their relevance scores. These relevant documents are used for constructing adaptive LMs and this step is carried out for each Chinese test document; thus, we build story-specific adaptive LMs. Finally, we interpolate the story-specific LMs with a general (static) N-gram LM. The number of English documents used, as well as the interpolation weights are

[5]Details and download are available at http://www.cs.cmu.edu/~lemur

again chosen so as to minimize the perplexity of the first pass output of the SMT system.

9.5 Experimental Results

In these experiments, we use the Johns Hopkins University Chinese-English machine translation setup for the NIST 2004 MT evaluation (Kumar et al., 2004a,b). Our results are obtained by performing LM rescoring on the N-best lists generated under the TTM that were supplied to us by Kumar et al. (2004a). The baseline LM training texts are mostly obtained from the LDC English Gigaword corpus[6].

- **AFP**: 562K documents, 201M words
- **XIE**: 611K documents, 156M words
- **FBIS**: 12K documents, 11M words
- **People's Daily**: 55K documents, 16M words

These documents are also used for IR; in other words, for each test story, LEMUR finds the most similar documents from the four English text collections. We build a baseline 3-gram LM for the first pass N-best (N = 1000) list generation, mixing four LMs, each of which is built using one of the four corpora, with equal weights. In the second pass, we experiment with N-best list rescoring using different LMs. We have used three evaluation sets, which are provided by NIST.

Table 9.1 shows the experimental results on the MT evaluation sets used in the NIST 2001, 2002, and 2003 evaluations; we will refer to each set as eval01, eval02, and eval03 respectively. The first row shows the BLEU scores using the

[6]Details of the Gigaword corpus may be found in the LDC web site which is available at http://wave.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05.

Model	Eval01		Eval02		Eval03	
	3gram	4gram	3gram	4gram	3gram	4gram
Baseline N-gram LM	28.7	29.7	27.7	28.3	27.1	27.4
Adaptive LM (Top 1000)	29.7	30.0	28.7	28.6	27.5	27.8
Adaptive LM (Opt. K)	29.7	30.2	28.6	28.8	27.6	27.9

Table 9.1: BLEU scores for NIST MT eval01, eval02, and eval03 sets

baseline static 3gram LM and a 4gram LM. The second row shows the results using the adaptive LMs constructed from the one thousand most similar retrieved documents from the ranked-list. Here, the interpolation weights between the adaptive LMs and the general LM are chosen to maximize the likelihood of the 1-best lists by the EM algorithm. Evidently, top 1000 document-based adaptive LMs show better results over the baseline LM results. Interestingly, the gains brought by the top 1000 document-based adaptive LMs are bigger than or equal to the ones obtained by moving from the 3gram LM to the 4gram LM. In other words, if we cannot afford to build the higher order N-gram LM, e.g., for memory usage constraints LM adaptation may be a better solution.

So far, we simply select the 1000-best-matching documents regardless of their similarity scores and adaptive LMs are built from the 1000 documents. A better solution may be to select the variable top K documents based on the similarity scores relative to some threshold. This approach, however, introduces another problem of finding the optimal threshold. It stands to reason, therefore, that choosing the optimal number of documents should be based on some optimization criterion. Here we use perplexity, which is simple to measure, as our optimization criterion or objective function. In other words, we find the optimal number of documents K by measuring the perplexity of the 1-best first-pass outputs—as we are not allowed to measure the perplexity of the reference translations—trying

Model	Eval01–03	
	3gram	4gram
Baseline static N-gram LM	27.8	28.5
Adaptive LM (Top 1000)	28.6	28.8
Adaptive LM (Opt. top K)	28.7	29.0

Table 9.2: BLEU scores for NIST MT evaluation 01–03 set

$K = 100, 500, 1000, 2000, 4000,$ and 8000. As the last row of Table 9.1 shows, we achieve further gains over the fixed top 1000 document-based cases, though they are insignificant. This result is different from our observation in ASR experiments, which leads us to analyze the results.

Next, we pool the eval01, eval02, and eval03 sets together and build a new set (eval01–03 set) in order to see the overall improvements in the whole test set. As Table 9.2 shows, the improvements in the BLEU score are evident. Unlike the ASR test, however, no standard statistical significance test program or script has been made publicly available yet. Therefore, we have measured the confidence intervals with the level of 95% confidence using bootstrap resampling (Bisani and Ney, 2004; Zhang et al., 2004) as described in Section 9.3.1.

The 95% confidence intervals are (27.3–28.4) for the 3gram and (27.9–29.0) for the 4gram baseline, respectively. Since our BLEU scores (when optimal top K were used) fall beyond the intervals for both of the 3gram and 4gram case, we conclude that our improvements are statistically significant. Indirect evidence that the adaptation is beneficial is also provided by the fact that the improvements are consistent across all three evaluation sets, as well as on a fourth 2004 evaluation set, which is described in Kumar et al. (2004b).

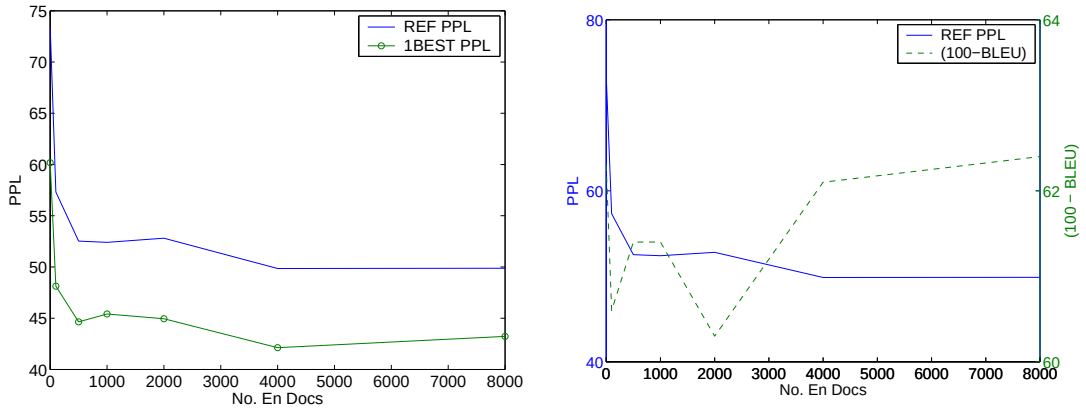

Figure 9.6: Perplexity of 1-best list and reference (left) and perplexity of reference and (100 − BLEU) scores (right) for one sample story according to different the number of English IR documents

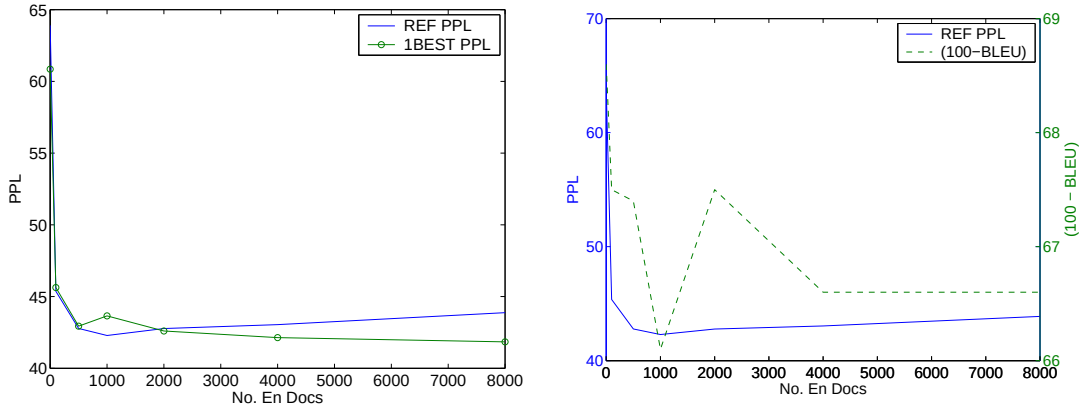

Figure 9.7: Perplexity of 1-best list and reference (left) and perplexity of reference and (100 − BLEU) scores (right) for another sample story according to different the number of English IR documents

9.6 Analysis of the Results

9.6.1 Perplexity and BLEU score for Different Number of English Documents

In this section, we continue with the analysis of the experimental results. Figure 9.6 shows the perplexity of the 1-best lists and the reference translations for one sample story when different numbers of English IR documents were used for

building story-specific adaptive language models (left plot). It also shows the corresponding (100 - BLEU) scores[7] of the best hypothesis after rescoring with the second pass decoding (right plot). Figure 9.7 is the same plots for another sample story.

The first thing to notice is that the perplexity curve drops rapidly when the number of English documents is first increased, and then the drop becomes insignificant (flat or increased for some points) in the left plot of the two stories. And this happens for both of the 1-best list and the reference translations. For the story in Figure 9.6, the two perplexity curves between the 1-best list and the reference translation show the perfect match, while they don't match for the other story in Figure 9.7. However, the differences of the perplexity in Y-axis for the mismatched case are not significant, which means that coarsely optimizing the perplexity of the 1-best list still seems to reduce the perplexity of the reference translation. Similarly, the BLEU score also improves initially and then oscillations around the higher value are observed.

Secondly, although the best number of English documents (in X-axis) match for both of the 1-best list and the reference translation in the one story (Figure 9.6), this is not always true for all stories as shown in Figure 9.7. Indeed, about a half of the randomly selected 20 sample stories show that the best numbers match, but the other half show slightly different numbers. Unlike the ASR case which has only one reference transcription, many (typically four) reference translations are used in MT and this may be the reason why there are many mismatches in the best number of documents. This suggests that the perplexity of the 1-best output may not be an appropriate criterion for fine-tuning the adaptation parameters to minimize the perplexity of all 4 reference translations.

[7]Remember that unlike perplexity and WER, the higher is the better in BLEU scores.

Furthermore, notice that the BLEU score curves[8] are seldom matched to the perplexity curves (right plots). Since the BLEU score is based on the precisions of N-grams in a hypothesis translation to the given reference translations, the behavior of the BLEU curve may not completely match the perplexity curves. This suggests that fine-tuning for perplexity reduction may not be necessary. Instead, directly maximizing an estimate of the BLEU score may be more beneficial.

Also, notice that there has been only small gains (or no gain) of BLEU scores between the top 1000 document-based adaptive LMs and optimal top K document-based ones in Table 9.1 and Table 9.2. Our explanation for this small gain is therefore the different behaviors of perplexity and BLEU curves. In other words, even though we have used perplexity as an objective function for optimization, we may not be able to find truly optimal language models in terms of the BLEU scores.

9.6.2 Analysis of N-Best List Rescoring

We analyze our experimental results in a different perspective. That is, we first measure the log probability scores of the 1000-best list hypotheses for one sample segment (sentence), and Figure 9.8 shows the plot. The curve denoted as "Original" shows the log probability scores of the original first pass N-best list hypotheses translations. Since the rank has been determined based on this score, the curve is monotonic decreased of course. After rescoring the N-best list hypotheses with the second pass language models, the new scores are then changed, as the dots labeled "Rescored" show. Notice that the rank of original hypotheses has been held same while plotting the dots. Finally, the hypotheses are next

[8]Notice that this curve is a (100 - BLEU) curve since the higher is the better in BLEU. Therefore, the lowest point of the curve is optimal in each of the curves.

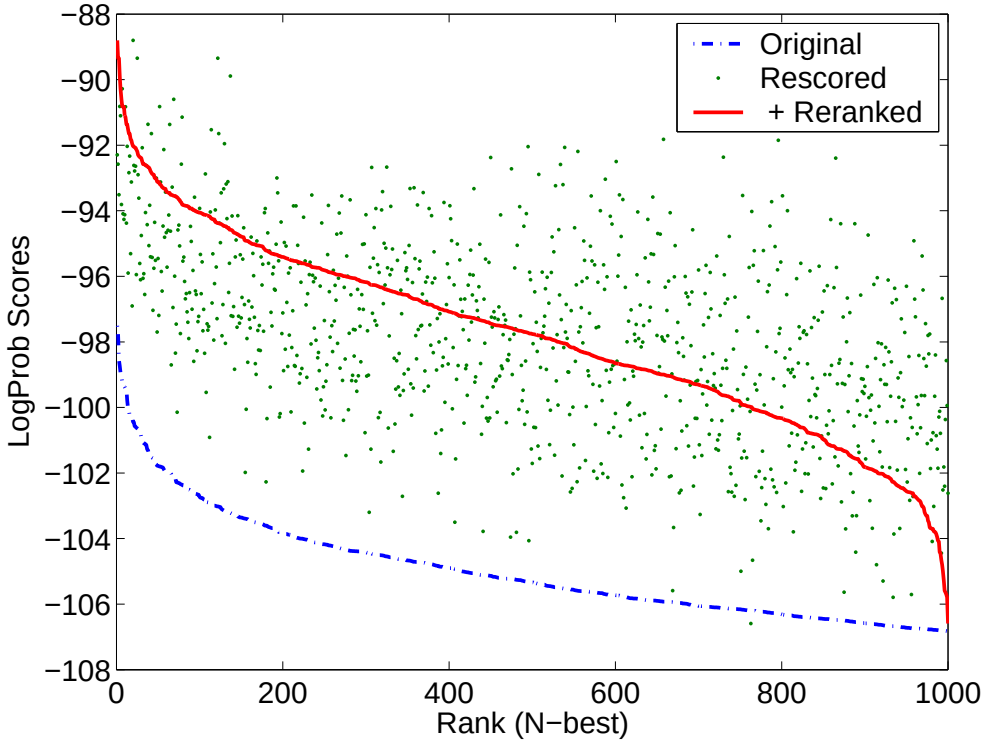

Figure 9.8: Log probability scores of the 1000-best list hypotheses for one sample segment

reranked according to these new scores, and the curve denoted as "+ Reranked" shows the plot. Notice that the all scores after rescoring with the second pass language model have improved, which demonstrates the language model is closer to the test data. Also, the top 1-best hypothesis in the first pass is no longer the best hypothesis after rescoring. Therefore, even though our second pass language model has been optimized for the 1-best hypothesis, overfitting to the 1-best hypothesis is not a concern.

The plot in Figure 9.8 is only from one sample segment, which means it may not reflect the overall tendency of the whole test data. Hence, we continue with plotting the curve of the average log probability scores for one test set, namely eval03 set. Since there are 919 segments for the eval03 test set, we first compute

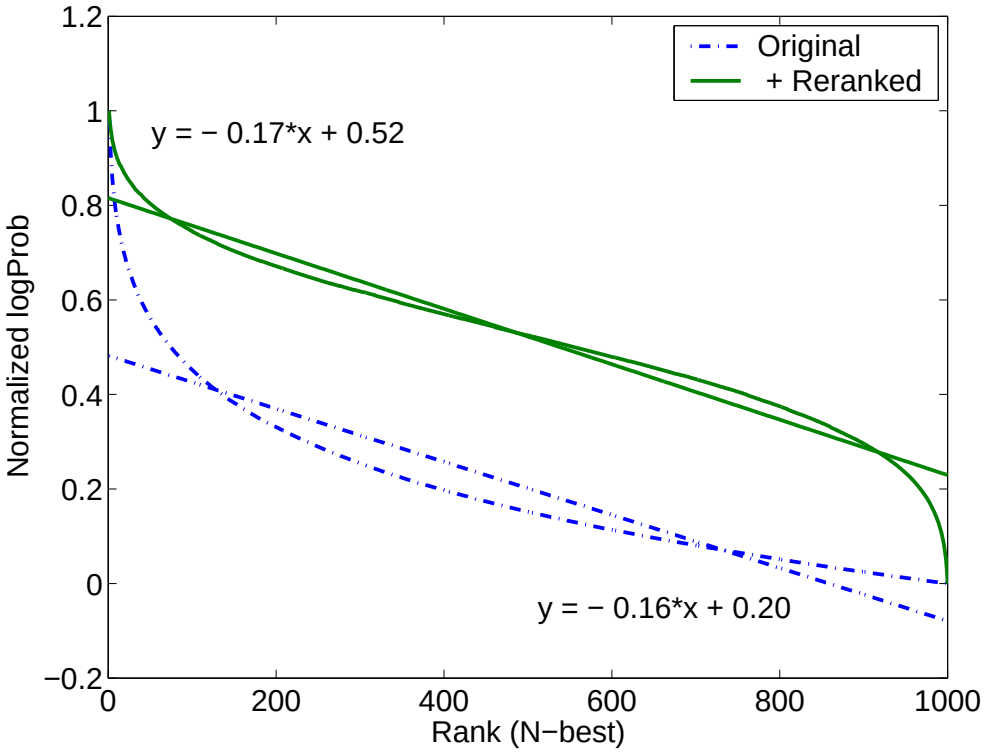

Figure 9.9: Average log probability scores of the 1000-best list hypotheses of the eval03 set

the log probability scores for 919 segments. Since the log probability scores of the 919 segments along the Y axis are distributed in a different scale, we first normalize the log probability scores so that they are distributed between 0 and 1, and then average them. Figure 9.9 shows the average log probability scores for the eval03 test set. We also find out the best line fit of the two curves (original N-best list scores and the scores after rescoring and reranking). As the linear fit equations show, the slope of the curve has not been significantly changed. Notice that this is an overall tendency for the eval03 test set since these curves are plotted from the average scores.

Finally, we take the exponential of the log probability scores to convert the scores into probabilities and normalized the probabilities to sum to one over the

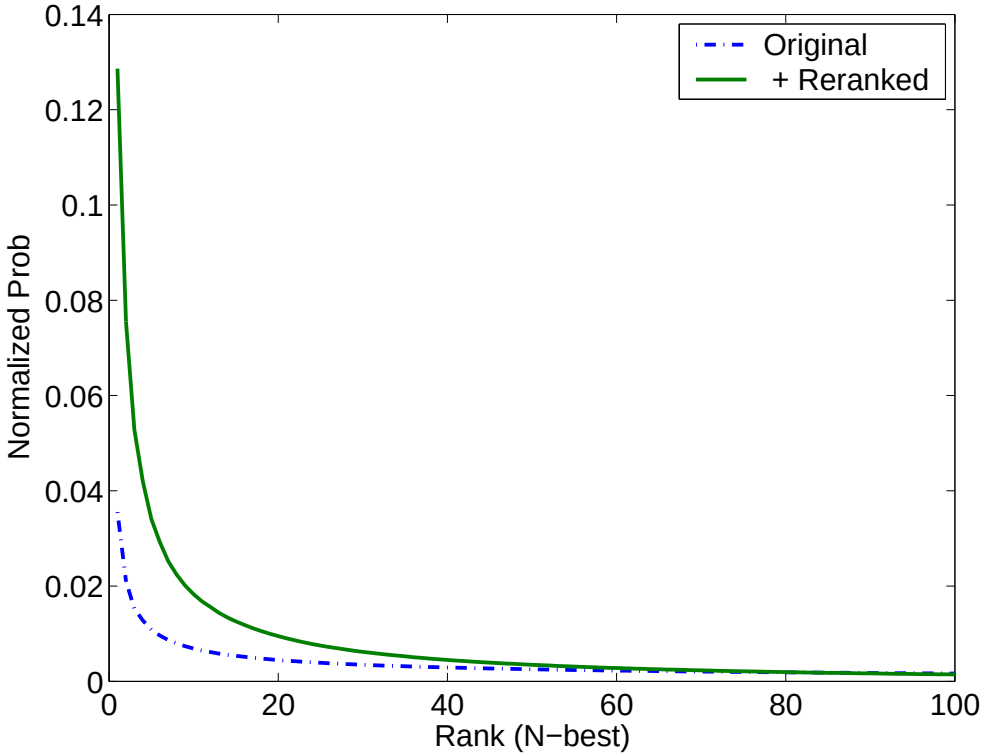

Figure 9.10: Average normalized probabilities of the 1000-best list hypotheses of the eval03 set

N-best list hypotheses. Figure 9.10 shows the plot for the first 100 best hypotheses. Note that the hypotheses whose scores yield the "+ Reranked" curve are different from the hypotheses whose scores yield the "Original" curve. The probability scores of the top hypotheses after rescoring and reranking have been boosted compared to the original N-best list scores. This is consistent with, e.g. MLLR adaptation in an ASR system where the per-frame log likelihood of the acoustics under the adapted models is higher than under the unadapted models. The larger dynamic range of the probabilities of the adapted LM however suggests that combining them with the translation model probabilities may require some adjustment, e.g. the LM scale factor.

9.7 Summary

We have demonstrated that language model adaptation benefits statistical machine translation even when 100's of millions of words of training texts are available. We have also shown that while the coarse likelihood-based document specific adaptation technique results in improving both the perplexity of the reference translation and the BLEU score of the resultant output. The nature of BLEU implies that perplexity-based fine-tuning has little additional benefits. This suggests the needs for investigating other secondary criteria for tuning the adaptation parameters.

Chapter 10

Conclusions

Statistical language modeling is indispensable in many speech and natural language applications. The performance of language models is inevitably influenced by the nature of training data and therefore, the success of language modeling depends on the abundance and nature of training text data. In other words, we are not able to build successful language models if 1) abundant training texts are not available (the data deficiency problem) or 2) training texts are not similar to the test data (the data mismatch problem). This dissertation has investigated aspects of language model adaptation to address the two problems in language modeling.

10.1 Main Contributions

10.1.1 Solution for the Data Deficiency Problem

First, we have proposed the methods for solving data deficiency problems when there is little or no data which is suitable for statistical language model training—via language model adaptation. Language model adaptation is an approach to

dynamically adjust language model parameters such as N-gram probabilities and the vocabulary of the language model according to the characteristics of the test set subunit. This subunit could be any of topic, genre, style, or document—as we have proposed in this dissertation. Typically, these parameters to be adjusted are estimated from a small amount of selected texts which is similar to the test data. In the end, these subunit-specific language models are combined with a general language model by some techniques such as linear interpolation or maximum entropy models.

Our adaptation approaches are novel in the sense that the source of adaptation comes from side texts which are written in a different language, which is preferably resource-rich, from the language of interest. The essence of our cross-lingual language model adaptation lies in the efficient combination of two techniques: cross-lingual information retrieval and machine translation. In other words, our cross-lingual language model adaptation consists of two steps: 1) cross-lingual information retrieval to identify useful texts from texts in a side language and 2) machine translation to *project* useful information in side language texts into the language of interest.

10.1.2 Obtaining the Translation Lexicons

This projection is possible only when translation lexicons are available, which is a serious demand especially when our language of interest is resource-deficient. If a sentence-aligned parallel corpus is available, we can use a public statistical machine translation toolkit, GIZA++ (Och and Ney, 2000), which is based on the IBM-models (Brown et al., 1990, 1993). However, a sentence-aligned parallel corpus may be an expensive resource, and it may not be easily available or constructed. We have proposed two novel alternatives to automatically build

translation lexicons from a document-aligned corpus, which is much easier to obtain. We have achieved similar results from a document-aligned corpus to the ones from a sentence-aligned parallel corpus—by using cross-lingual lexical triggers and cross-lingual latent semantic analysis.

One benefit from using cross-lingual triggers and cross-lingual latent semantic analysis is that they work even when the document-aligned corpus may not be exact translations of each other! Since both methods are based on word usage within a document pair along the comparable corpus, they can capture dependencies of highly correlated word pairs in two languages. In other words, if a pair of words are closely related to each other, and appear in paired documents, this correspondence may be captured by cross-lingual triggers or by cross-lingual latent semantic analysis even though they are not a translation pair. Therefore, a pair of news articles about the same event or topic can be used for cross-lingual triggers and cross-lingual latent semantic analysis while it cannot be used for GIZA++ training.

10.1.3 Story-Specific Likelihood-Based Optimization

Finding the optimal parameter happens frequently in many of real world engineering applications and is an important issue. As shown in Section 6.3 if we blindly choose any parameter among many possible choices, it is hard to expect to achieve significant gains. Rather, we are able to achieve further gains by careful investigation into the optimal parameters from the statistics collected from available resources. In this dissertation, we have proposed a novel method to find optimal parameters for language model adaptation—based on likelihood-based story-specific optimization.

10.1.4 Investigation into the Data Mismatch Problem

The contribution here is that we have applied our language model adaptation techniques to a different application, Chinese to English statistical machine translation. If decent quality translations are available between a source and a target language, we don't have to rely on cross-lingual information retrieval. Rather, directly applying monolingual information retrieval for language model adaptation would be a better solution. Furthermore, we set our main language of interest to English which is resource-rich. In other words, we don't have to worry about the data deficiency problem since extensive amounts of English texts are available.

10.2 Summary of the Results

We have demonstrated a statistically significant improvement in ASR WER (1.4% absolute) on a Chinese task and in perplexity (23%) by exploiting cross-lingual side-information even when a nontrivial amount of training data *is* available, as seen on the 13M-word XINHUA corpus. Our methods are even more effective when LM training text is hard to come by in the language of interest: 47% reduction in perplexity and 1.3% absolute in WER as seen on the 96K-word HUB-4NE corpus. Most of these gains come from the optimal choice of adaptation parameters. The ASR test data we used in our experiments is derived from a different news source than the text corpus on which the translation models are trained, which points to the robustness of the inferred statistics.

The techniques work even when the bilingual corpus is merely document-aligned, which is a realistic reflection of the situation in a resource-deficient language. Effectively, we have proposed methods to build cross-lingual language models which do not require machine translation. By using mutual information

statistics and latent semantic analysis from a *document-aligned* corpus, we can extract a significant amount of information for language modeling. Experimental results show that performance statistically equal to the methods predicated on MT capabilities can be achieved by those methods.

Finally, our statistical machine translation experimental results have shown statistically significant improvements (0.9 absolute BLEU score for the 3gram LM baseline and 0.5 absolute BLEU score for the 4gram LM baseline) on a state-of-the-art baseline trained on the Gigaword corpus by applying our story-specific language model adaptation. Notice that even with the state-of-the-art English baseline which is constructed from almost a half billion word corpus—which means that it is difficult to beat—we have been able to obtain improvements. These results strongly indicate that our language model adaptation is effective even with resource-rich languages.

10.3 Future Work

10.3.1 Maximum Entropy Model

Recall that the maximum entropy model from a family of models with constraints on the marginal probability of all frequently observed unigrams, bigrams and trigrams has an exponential form

$$P_{\mathrm{ME}}(c_k|c_{k-1}, c_{k-2}) = \frac{\alpha_{c_k}^{f_1(c_k)} \alpha_{c_{k-1},c_k}^{f_2(c_{k-1},c_k)} \alpha_{c_{k-2},c_{k-1},c_k}^{f_3(c_{k-2},c_{k-1},c_k)}}{Z(c_{k-1}, c_{k-2})}\,, \tag{10.1}$$

where the f_N's are indicator functions of the N-grams whose marginal probability is constrained, and the α's are the associated free parameters. If the expected values of the indicator functions are constrained to be the relative frequencies of

the respective N-grams in the LM training corpus, then the maximum entropy model is also the *maximum likelihood* estimate from the family of exponential models defined above.

To exploit cross-lingual side-information, we condition the LM for a Chinese document d_i^C on its English counterpart d_i^E, adding a set of trigger features, for example, so that

$$P_{\text{CL-ME}}(c_k|c_{k-1}, c_{k-2}, d_i^E) = \frac{\alpha_{c_k}^{f_1(c_k)} \alpha_{c_{k-1},c_k}^{f_2(c_{k-1},c_k)} \alpha_{c_{k-2},c_{k-1},c_k}^{f_3(c_{k-2},c_{k-1},c_k)} \prod_{e^* \in d_i^E} \alpha_{e^*,c_k}^{f_\tau(e^*,c_k)}}{Z(c_{k-1}, c_{k-2}, d_i^E)},$$

where the cross-lingual triggers $f_\tau(e^*, c^*)$'s are active only when English trigger words e^*, selected as described above, are present in d_i^E, *and* when $c_k = c^*$. The target expectations of the $f_\tau(e^*, c^*)$'s may be constrained to be the unigram frequencies $\hat{P}(c^*|e^*)$ of c^* in the subset of Chinese documents whose English counterparts contain e^*.

Unlike the interpolation of (4.6), which raises or lowers the probability of $c_k = c^*$ in every N-gram context (c_{k-1}, c_{k-2}), a maximum entropy model with triggers is able to make the necessary adjustments — if a particular context has powerful N-gram predictors which correctly account for the probability of c^*, the triggers make little difference; if the context fails to adequately discriminate between cases when c^* is likely and unlikely, the triggers appropriately exert greater influence. Use of such models remains to be investigated.

10.3.2 Bootstrapping the Translation Dictionary

Our cross-lingual language model adaptation approaches can be summarized as two steps: first, identify relevant documents in a resource-rich language and extract a useful statistic and then translate the statistic into the main target lan-

guage. In both of these steps, inevitably, the translation dictionary plays an important role. We have also proposed methods to construct translation lexicons automatically from a parallel corpus. That is, if a sentence-aligned parallel corpus—which is difficult to obtain—is available, we can use a statistical machine translation toolkit. If a document-aligned corpus, instead, is available, we can use the cross-lingual trigger approach or the cross-lingual latent semantic analysis approach. However, even the document-aligned corpus may not be easily available. Under the assumption that small size translation lexicons are available, we can automatically build the comparable corpus using the small size translation lexicons as a seed dictionary by bootstrapping.

Suppose a small-size seed translation dictionary is available either from a machine readable dictionary, a sentence-aligned parallel corpus, or a document-aligned corpus. Suppose further an additional monolingual corpus in a resource-deficient language is available. There may or may not be relevant documents in a resource-rich language to the resource-deficient language document. In either case, we can run CLIR using the seed translation dictionary to identify the most similar documents in the resource-rich language. Since the result of CLIR is typically a ranked list of documents along with their similarity scores, we can identify the most and *strongly* similar document in the resource-rich language to the one in the resource-deficient language by looking at the similarity score—whether it is a translation or not. Consequently, taking the document pair as aligned, we can extend the collection of document pairs. Then, we can extend the translation dictionary from the extended document pairs by using either cross-lingual triggers or cross-lingual latent semantic analysis.

Note that initial requirements for this approach are the seed dictionary and documents in the resource-deficient language. As an extensive amount of resource-

rich language corpus would be easily available, it is likely that there *is* a relevant document to the resource-deficient language document. We believe that the two initial resources are the minimal requirements for our cross-lingual language model adaptation, and as long as these initial conditions are satisfied, our approaches are expected to bring significant improvements.

Bibliography

Y. Akiba, K. Imamura, and E. Sumita. Using multiple edit distances to automatically rank machine translation output. In *Proc. MT Summit VIII*, pages 15–20, 2001.

J. Allan, D. Caputo, D. Gildea, R. Hoberman, H. Jin, V. Lavrenko, M. Rajman, and C. Wayne. Topic-based novelty detection. *Johns Hopkins Summer Workshop*, 1999. http://www.clsp.jhu.edu/ws99.

R. A. Baeza-Yates, R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

J. R. Bellegarda. A multispan language modeling framework for large vocabulary speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6(5): 456–467, September 1998.

J. R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88:1279–1996, 2000a.

J. R. Bellegarda. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1): 76–84, January 2000b.

A. Berger and R. Miller. Just-in-time language modeling. In *Proc. ICASSP*, volume 2, pages 705–708, Seattle WA, USA, 1998.

M. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM review*, 37(4):573–595, 1995.

M. W. Berry, Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999. ISSN 0036-1445.

M. Bisani and H. Ney. Bootstrap estimates for confidence intervals in asr performance evaluation. In *Proc. ICASSP*, volume 1, pages 409–412, Montreal, Quebec, Canada, May 2004.

P. F. Brown, J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

W. Byrne, P. Beyerlein, J. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and W. Wang. Towards language independent acoustic modeling. In *Proc. ICASSP*, pages 1029–1032, Istanbul, Turkey, 2000. IEEE.

W. Byrne, S. Khudanpur, W. Kim, S. Kumar, P. Pecina, P.Virga, P. Xu, and D. Yarowsky. The Johns Hopkins University 2003 Chinese-English Machine Translation System. In *Machine Translation Summit IX*, 2003.

C. Chelba. *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, Johns Hopkins University, Baltimore, MD, USA, 2000.

L. Chen, J.-L. Gauvain, L. Lamel, and G. Adda. Unsupervised language model adaptation for broadcast news. In *Proc. of ICASSP*, volume 1, pages 220–223, Hong Kong, China, 2003.

S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University, 1998.

S. Chen and R. Rosenfeld. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, 1999.

S. Chen, K. Seymore, and R. Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models. In *Proc. ICASSP*, Seattle, WA, USA, 1998.

C. Christopher, D. Graff, N. Martey, and S. Strassel. The TDT-3 text and speech corpus. In *Proc. Topic Detection and Tracking Workshop*, Vienna, VA, USA, 2000.

P. Clarkson and A. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proc. ICASSP*, volume 2, pages 799–802, Munich, Germany, 1997.

N. Coccaro and D. Jurafsky. Towards better integration of semantic predictors in statistical language modeling. In *Proc. ICSLP*, volume 6, pages 2403–2406, Sydney, Aus., 1998.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., N. Y., 1991. ISBN 0-471-06259-6.

I. Csiszár. A geometric interpretation of darroch and ratcliff's generalized iterative scaling. *The Annals of Statistics*, 17(3):1409–1413, 1989.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(3):1470–1480, 1972.

M. Davis and W. Ogden. QUILT: Implementing a large-scale cross-language text retrieval system. In *Proc. SIGIR*, pages 92–98, Philadelphia, PA, July 1997.

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.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

D. Doermann, H. Ma, B. Karagol-Ayan, and D. Oard. Translation lexicon acquisition from bilingual dictionaries. In *Proc. SPIE Photonic West article Imaging Conference*, pages 37–48, San Jose, CA, 2002.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, USA, 1974.

S. Dumais, T. Landauer, and M. Littman. Automatic cross-language retrieval using latent semantic indexing. In *AAAI Spring Symposuim on Cross-Language Text and Speech Retrieval*, pages 18–24, 1997.

S. T. Dumais, T. K. Landauer, and M. L. Littman. Automatic cross-linguistic information retrieval using latent semantic indexing. In *Proc. SIGIR - Workshop on Cross-Linguistic Information Retrieval*, pages 16–23, 1996.

M. Federico. Efficient language model adaptation through MDI estimation. In *Proc. of Eurospeech*, pages 1583–1586, Budapest, Hungary, 1999.

M. Federico and N. Bertoldi. Broadcast news lm adaptation using contemporary texts. In *Proc. of Eurospeech*, Aalborg, Denmark, 2001.

R. Florian and D. Yarowsky. Dynamic non-local language modeling via hierarchical topic-based adaptation. In *Proceedings of ACL*, College Park, MD, USA, 1999.

P. Fung, W. Byrne, T. Zheng, V. Venkatramani, U. Ruhi, L. Yi, Z. Song, and T. Kamm. Pronunciation modeling of mandarin casual speech. *Johns Hopkins Summer Workshop*, 2000. http://www.clsp.jhu.edu/ws2000/groups/mcs.

D. Gildea and T. Hofmann. Topic-based language models using EM. In *Proc. Eurospeech*, pages 2167–2170, Budapest, 1999.

J. Godfrey, E. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. ICASSP*, pages 517–520. IEEE, 1992.

G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. pub-JHUP, third edition, 1996.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(16):237–264, 1953.

Y. Gotoh and S. Renals. Topic-based mixture language modelling. *Journal of Natural Language Engineering*, 5:355–375, 1999.

G. Grefenstette and G. Grefenstette. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001.

P. Husbands, H. Simon, and C. Ding. the use of singular value decomposition for text retrieval. In M. Berry, editor, *Proc. of SIAM Comp. Info. Retrieval Workshop*, October 2000.

R. Iyer and M. Ostendorf. Modeling long-distance dependence in language: topic-mixtures vs dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7:30–39, 1999.

F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997. ISBN 0-262-10066-5.

S. Khudanpur and W. Kim. Using cross-language cues for story-specific language modeling. In *Proc. ICSLP*, volume 1, pages 513–516, Denver, CO, 2002.

S. Khudanpur and W. Kim. Contemporaneous text as side-information in statistical language modeling. *Computer Speech and Language*, 18(2):143–162, April 2004.

S. Khudanpur and J. Wu. A maximum entropy language model to integrate n-grams and topic dependencies for conversational speech recognition. In *Proc. ICASSP*, volume 1, pages 553–556, Phoenix AZ, USA, 1999.

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

W. Kim and S. Khudanpur. Language model adaptation using cross-lingual information. In *Proc. Eurospeech*, pages 3129–3132, Geneva, Switzerland, 2003b.

W. Kim and S. Khudanpur. Cross-lingual latent semantic analysis for language modeling. In *Proc. ICASSP*, volume 1, pages 257–260, Montreal, Quebec, Canada, 2004a.

W. Kim and S. Khudanpur. Lexical triggers and latent semantic analysis for cross-lingual language model adaptation. *ACM Transactions on Asian Language Information Processing*, 3(2):94–112, June 2004b.

W. Kim, S. Khudanpur, and J. Wu. Smoothing issues in the structured language model. In *Proc. Eurospeech*, volume 1, pages 717–720, Aalborg, Denmark, 2001.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. ICASSP*, volume 1, pages 181–184, 1995.

R. Kneser, H. Peters, and D. Klakow. Language model adaptation using dynamic marginals. In *Proc. Eurospeech*, pages 1971–1974, Rhodes, Greece, 1997.

K. Knight. A statistical machine translation tutorial workbook, 1999. Available at http://www.isi.edu/natural-language/mt/wkbk.rtf.

R. Kuhn and R. D. Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, 1990.

S. Kumar and W. Byrne. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proc. HLT-NAACL*, Edmonton, Canada, 2003.

S. Kumar, Y. Deng, and W. Byrne. A weighted finite state transducer translation template model for statis tical machine translation. *Journal of Natural Language Engineering*, 2004a. To appear.

S. Kumar, Y. Deng, C. Schafer, W. Kim, P. Virga, N. Habash, D. Smith, F. Jurcicek, W. Byrne, S. Khudanpur, Z. Shafran, and D. Yarowsky. The Johns Hopkins University 2004 Chinese-English and Arabic-English MT evaluation systems, 2004b. http://www.nist.gov/speech/tests/mt.

T. Landauer and L. Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31–38, 1990.

R. Lau. Adaptive statistical language modelling. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1994.

LDC. The TDT4 corpus, 2002. http://www.ldc.upenn.edu/Projects/TDT4.

C.-Y. Lin and F. J. Och. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proc. COLING*, Geneva, Switzerland, August 2004.

X. Ma. Hong Kong news parallel text corpus, 2000. http://www.ldc.upenn.edu/Catalog/ LDC2000T46.html.

S. Martin, J. Liermann, and H. Ney. Adaptive topic-dependent language modelling using word-based varigrams. In *Proc. Eurospeech*, pages 1447–1450, Rhodes, Greece, 1997.

S. Martin, H. Ney, and C. Hamacher. Maximum entropy language modeling and the smoothing problem. *IEEE Transactions on Speech and Audio Processing*, 8(5):626–632, Sept. 2000.

I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In *Proc. HLT/NAACL*, Edmonton, Canada, 2003.

M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

M. Mohri, F. Pereira, and M. Riley. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231(1):17–32, 2000.

NIST. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, 2002. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

D. Oard. Alternative approaches for cross-language text retrieval. Technical report, AAAI Technical Report SS-97-05, 1997.

F. Och, D. Gildea, A. Sarkar, S. Khudanpur, K. Yamada, A. Fraser, S. Kumar, D. Smith, V. Jain, K. Eng, Z. Jin, and D. Radev. Syntax for statistical machine translation. *Johns Hopkins Summer Workshop*, 2003. http://www.clsp.jhu.edu/ws2003/groups/translate.

F. Och and H. Ney. Improved statistical alignment models. In *Proc. ACL*, pages 440–447, Hong Kong, China, October 2000.

F. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proc. EMNLP*, pages 20–28, College Park, MD, USA, 1999.

D. Pallett, W. Fisher, and J. Fiscus. Tools for the analysis of benchmark speech recognition tests. In *Proc. ICASSP*, volume 1, pages 97–100, Alburquerque, NM, 1990.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, Philadelphia, PA, July 2002.

S. D. Pietra, V. J. D. Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (4):380–393, 1997.

D. Radev, S. Teufel, W. Lam, H. Saggion, D. Liu, H. Qi, E. Drabek, J. Blitzer, and A. Celebi. Automatic summarization of multiple (multilingual) documents. *Johns Hopkins Summer Workshop*, 2001. http://www.clsp.jhu.edu/ws2001.

J. Rice. *Mathematical statistics and data analysis*. Duxbury Press, Belmont CA, second edition, 1995.

R. Rosenfeld. *Adaptvie Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10:187–228, 1996.

G. Salton and M. E. Lesk. The smart automatic document retrieval systems–an illustration. *Communications of ACM*, 8(6):391–398, 1965. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/364955.364990.

G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

P. Scheytt, P. Geutner, and A. Waibel. Serbo-croatian LVCSR on the dictation and broadcast news domain. In *Proc. ICASSP*, volume 2, pages 897–900, Seattle WA, USA, 1998.

T. Schultz and A. Waibel. Language independent and language adaptive large vocabulary speech recognition. In *Proc. ICSLP*, volume 5, pages 1819–1822, Sydney, Australia, 1998.

K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In *Proc. Eurospeech*, volume 4, pages 1987–1990, Rhodes, Greece, 1997.

C. Tillmann and H. Ney. Word trigger and the EM algorithm. In *Proc. of the Workshop Computational Natural Language Learning (CoNLL 97)*, pages 117–124, Madrid, Spain, 1997.

S. Vogel, F. Och, C. Tillmann, S. Nieen, H. Sawaf, and H. Ney. *Statistical Methods for Machine Translation*. Wolfgang Wahlster (ed.). Springer Verlag, Berlin Germany, July 2000.

J. Wu. *Maximum Entorpy Language Modeling with Non-Local Dependencies*. PhD thesis, Johns Hopkins University, Baltimore, MD, USA, 2002.

D. Yarowsky, G. Ngai, and R. Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proc. HLT*, pages 109–116, Santa Monica, CA, 2001.

Y. Zhang, S. Vogel, and A. Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proc. LREC*, Lisbon, Portual, May 2004.

Vita

Woosung Kim was born in Taejon, South Korea. He graduated from Korea Advanced Institute of Science and Technology with a Bachelor of Science in Computer Science, in 1990. He also received a Master of Science and Engineering in Computer Science and Engineering department from Pohang University of Science and Technology, in 1992. After speding 6 years working as a researcher in Korea Telecom Research and Development Group—mainly focusing on building a Korean speech recognizer and a Korean to Japanese speech translation system—he decided to come to the U.S. to pursue further education. In 2000, he earned a Master of Science from the Computer Science department, the Johns Hopkins University. In 2001, he worked as a summer intern for SpeechWorks International which is now ScanSoft Inc. His research topics include language modeling, automatic speech recognition, natural language processing, machine translation, and information retrieval.

Baltimore, 2004