Detecting Translational Equivalences in Comparable Corpora

Sanjika Hewavitharana

CMU-LTI-12-006

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Stephan Vogel (Chair)
Alon Lavie
Noah A. Smith
Pascale Fung, Hong Kong University of Science and Technology

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

One of the major bottlenecks in developing machine translation systems for many language pairs is the lack of parallel data. Due to the time consuming and expensive effort required to create them, these corpora are limited in quantity, genre and language coverage. For most language pairs, parallel data is nonexistent. Comparable corpora, which are more widely available, offer a solution to this data sparseness problem. These are collections of bilingual articles that may not be exact translations, but contain similar information. Multilingual newswire articles produced by news organizations such as AFP and BBC are a good example for a comparable corpus.

This thesis focuses on detecting translational equivalences in comparable corpora. We present a hierarchical framework to identify parallel data at different levels of granularity, namely document, sentence and phrase level. In each step, part of the data extracted is used as the input to the next step. Different algorithms are proposed for each of the tasks. In particular, this work focuses on identifying parallel phrases embedded in comparable sentence pairs. We propose a phrase extraction approach that does not rely on the Viterbi path of word alignment models, which makes it well suited for the task.

The proposed algorithms are evaluated intrinsically against manually aligned data, as well as extrinsically by using the extracted resources in building translation systems. We conduct experiments in different scenarios of data availability, varying from large to small, to no initial parallel data situations. Finally, we demonstrate the effectiveness of the proposed methods by building machine translation systems using comparable data for low resource languages, for which MT is otherwise not possible.

Acknowledgments

As I look back at the time spent at Carnegie Mellon, I am reminded of the many people who helped me in making this journey a success.

First and foremost, I would like to thank my advisor Stephan Vogel for the invaluable guidance, encouragement, and support throughout my graduate studies at Carnegie Mellon. He introduced me to the field of machine translation, and over the years helped me build the analytical skills and critical thinking that are essential in research. I would also like to express gratitude to my thesis committee members Alon Lavie, Noah A. Smith and Pascale Fung for their insightful feedback. Their valuable comments and suggestions helped in making this thesis much stronger.

I have benefited greatly from the knowledge and expertise of the LTI faculty members. Thanks to Tanja Shultz, Alon Lavie, Noah A. Smith, Alan Black, Jaime Carbonell, Ralf Brown, Robert Frederking, Teruko Mitamura, Eric Nyberg, Lori Levin, Maxine Eskenazi and Roni Resenfeld for your support during many courses and projects.

I thank Yaser Al-Onaizan for giving me the great opportunity of working at the IBM T.J. Watson Research Center as an intern in the machine translation group. I learned a great deal from the discussions with Christoph Tillmann, Yung-Suk Lee and Bing Zhao. The collaboration with them had a profound effect on many aspects of this thesis.

Many thanks to SMT team members: Fei Huang, Bing Zhao, Ying Zhang, Ashish Venugopal, Matthias Eck, Andreas Zollmann, Silja Hildebrand, Mohamed Noamany , Nguyen Bach, ThuyLinh Nguyen, Qin Gao and Mridul Gupta for your support and constructive feedback that helped me immensely in my research. It was a pleasure working with you.

InterACT provided with me an excellent environment to pursue my studies. I thank Alex Waibel for giving me the opportunity to be part of it. Thanks to my friends and colleagues at InterACT over the past years Paisarn Charoenpornsawat, Sharath Rao, Susi Burger, Thomas Schaaf, Ian Lane, Mark Fuhs, Chiori Hori, Stan Jou, Wilson Tam, Qin Jin, Kornel Laskowski, Udhyakumar Nallasamy Roger Hsiao, Celine Carraux, Anthony D'Auria, Isaac Harris, Lisa Mauti, Steven Valent, Shirin Saleem and Sameer Badaskar for the wonderful time.

I had the pleasure of great company and collaboration of many friends at LTI. Thanks to Jaedy Kim, Vamshi Ambati, Alok Parlikar, Vitor Carvalho, Wen Wu, Shinjae Yoo, Rashmi Gangadharaiah, Satanjeev Banerjee, Guy Lebanon, Katharina Probst, Narges Sharif Razavian, Aaron Phillips, Betty Cheng, Chih-yu Chao, Rohit Kumar, David Huggins-Daines, Nikesh Garera, Shilpa Arora, Hideki Shima and many others for making my time at Carnegie Mellon a memorable one. Your hard work and commitment have been an inspiration for me. Jaedy has been my faithful friend throughout the time at Carnegie Mellon. Thank you for all the help especially in my early days in Pittsburgh. Paisarn, Alok and Linh thank you for being such wonderful officemates. I will miss our enlightening discussions and laughs over coffee breaks.

I would like to thank the LTI staff members Radha Rao, Linda Hager, Stacey Young, Kelly Widmaier and Mary Jo Bensasi for their kind help.

The Sri Lankan community in Pittsburgh and especially fellow graduate students in Oakland area had supported me throughout the years. Thanks Ananda, Chandrasiri, Kavan, Reza, Ishara, Nusran and many others for being my family away from home.

My undergraduate advisors at the University of Colombo - N.D kodikara, Chandrika Fernando and Ruvan Weerasinghe were instrumental in directing me to the research in natural language processing. I owe my pursuit of graduate studies to them. Thank you for all the support and encouragement.

My deepest gratitude to my parents for their unwavering support, love and encouragement throughout the years. Many thanks to my sister and brother-in-law for their love and support. This thesis would not have been possible without the strength and love I received from my wife Shalika. Thank you for being with me through the tough times during this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Parallel corpora are an important resource for many natural language processing tasks, including statistical machine translation (SMT) [Brown et al., 1993; Och and Ney, 2002] and cross-lingual information retrieval [Oard, 1997]. Unfortunately, parallel corpora are not readily available in desired quantities. Due to the special effort that is required to create them, which is time consuming and costly, these corpora are limited in quantity, genre and language coverage. Large parallel corpora are only available for a handful of language pairs including French-English, Spanish-English, Chinese-English and Arabic-English. The majority of this data comes from parliamentary proceedings (Canadian Hansard, European Parliament, or the United Nations) and a limited amount of newswire text is also available. For a vast majority of other language pairs, there is a severe dearth of publicly available parallel corpora.

The lack of parallel data is especially problematic for SMT because it needs a considerable amount of training data to produce reliable models. The translation performance of SMT systems directly depends on the quantity and the quality of the available parallel data. Disparity in genre could affect translation quality when a system trained on data from a particular genre is used to translate text from a different genre. This makes building MT systems for language pairs with low resources, as well as non-common genres infeasible.

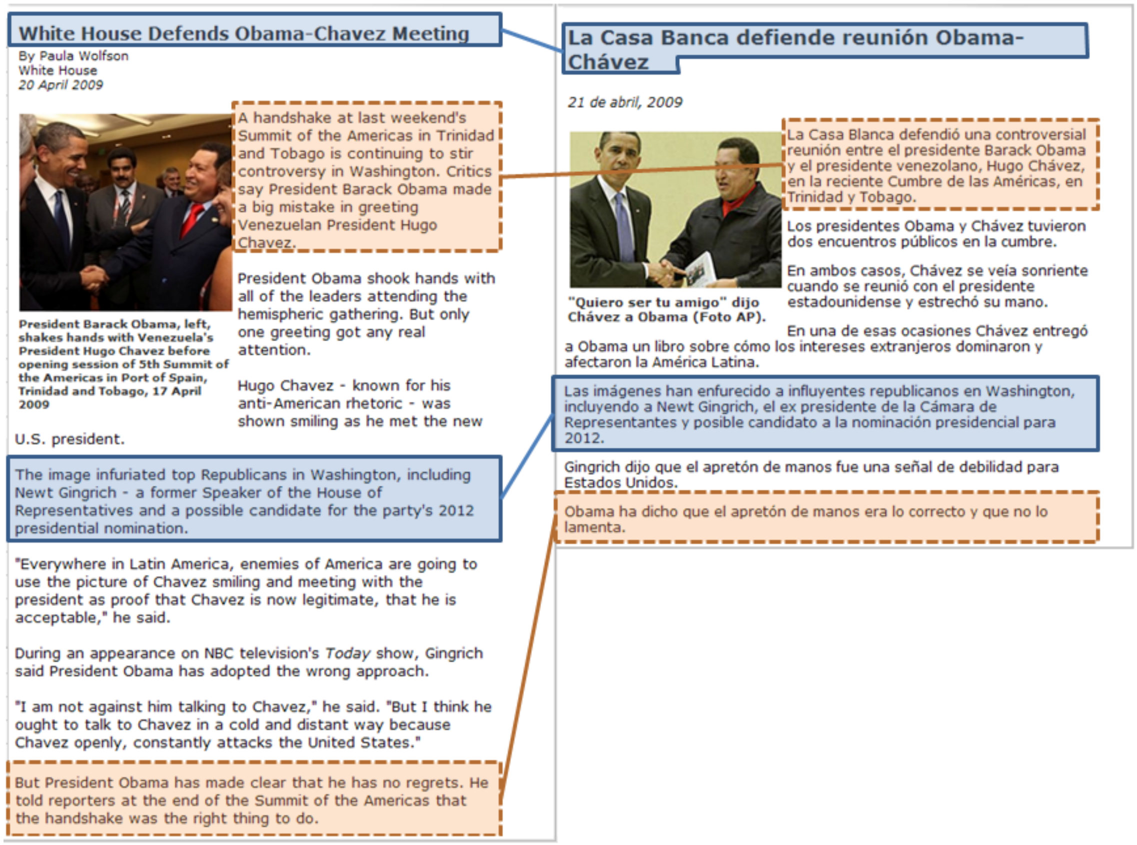
One solution to the bottleneck of data sparseness is to exploit the much richer body of comparable corpora. These are not exactly bilingual translations of each other. Rather, they are collections of documents that contain similar information. Fung and Cheung [2004b] provides a classification of bilingual corpora on different degrees of parallelism: A *Parallel corpus* is a sentence-aligned corpus containing bilingual translations of the same text. A *noisy parallel corpus* contains non-aligned sentences, but are mostly parallel. A *comparable corpus* contains non-sentence-aligned, non-translated bilingual documents that are topic-aligned. A *very-non-parallel* corpus contains disparate, non-parallel bilingual documents that can either be on the same topic or not. A good example of comparable corpora is the multilingual newswire texts that are produced by news organizations such as Agence France Presse, Reuters and BBC. These texts often describe the same event in multiple languages and with a varying degree of details. They may contain sentences that are fairly good translations of each other, especially when the story written in one language was used as a starting point to report about the event in some other language.

The Web is by far the largest and most accessible source of comparable texts [Resnik and Smith, 2003]. It contains an ever-expanding body of texts in multiple languages, including the already mentioned news articles, but also multilingual websites of companies, technical documentation and even blogs by different people may contain comparable documents when discussing the same topic.

A particularly attractive, more structured web-based resource is the online encyclopaedia Wikipedia, which is a vast domain-independent pool of user generated articles. An article is typically available in more than one language, although much of the content is developed independently for different languages. The rich collection of manually defined terms, concepts and relations in multiple languages, makes it a very promising resource.

These resources can be used in a number of ways to enhance translation systems. The most straightforward way is to mine for parallel document pairs from the web and other multilingual sources. These parallel documents can then be used directly in training translation models. Most of the time, however, it is hard to find entire documents of parallel text. Comparable documents are much easier to find. Parallel sentence pairs can be identified from comparable documents that contain the same information. Even if two comparable documents have few or no parallel sentence pairs, there could still be parallel sub-sentential fragments, including word translation pairs, named entities, and long phrase pairs. Figure 1.1 highlights these different levels of parallelism present in a comparable document pair.

Detecting translational equivalences in comparable data at sentence level or sub-sentential level is a challenging task. Even in a comparable document pair that contains the same information, there could still be a great variation at the sentence level. The order in which information is presented might be different in the two documents. Traditional sentence alignment algorithms [Gale and Church, 1991; Wu, 1994; Moore, 2002] that are designed to work on parallel text will perform poorly on comparable corpora due to this reordering problem. Current word alignment approaches [Brown et al., 1993; Vogel et al., 1996] which



White House Defends Obama-Chavez Meeting

By Paula Wolfson
White House
20 April 2009

A handshake at last weekend's Summit of the Americas in Trinidad and Tobago is continuing to stir controversy in Washington. Critics say President Barack Obama made a big mistake in greeting Venezuelan President Hugo Chavez.

President Barack Obama, left, shakes hands with Venezuela's President Hugo Chavez before opening session of 5th Summit of the Americas in Port of Spain, Trinidad and Tobago, 17 April 2009

President Obama shook hands with all of the leaders attending the hemispheric gathering. But only one greeting got any real attention.

Hugo Chavez - known for his anti-American rhetoric - was shown smiling as he met the new U.S. president.

The image infuriated top Republicans in Washington, including Newt Gingrich - a former Speaker of the House of Representatives and a possible candidate for the party's 2012 presidential nomination.

"Everywhere in Latin America, enemies of America are going to use the picture of Chavez smiling and meeting with the president as proof that Chavez is now legitimate, that he is acceptable," he said.

During an appearance on NBC television's *Today* show, Gingrich said President Obama has adopted the wrong approach.

"I am not against him talking to Chavez," he said. "But I think he ought to talk to Chavez in a cold and distant way because Chavez openly, constantly attacks the United States."

But President Obama has made clear that he has no regrets. He told reporters at the end of the Summit of the Americas that the handshake was the right thing to do.

La Casa Banca defiende reunión Obama-Chávez

21 de abril, 2009

La Casa Blanca defendió una controversial reunión entre el presidente Barack Obama y el presidente venezolano, Hugo Chávez, en la reciente Cumbre de las Américas, en Trinidad y Tobago.

"Quiero ser tu amigo" dijo Chávez a Obama (Foto AP).

Los presidentes Obama y Chávez tuvieron dos encuentros públicos en la cumbre.

En ambos casos, Chávez se veía sonriente cuando se reunió con el presidente estadounidense y estrechó su mano.

En una de esas ocasiones Chávez entregó a Obama un libro sobre cómo los intereses extranjeros dominaron y afectaron la América Latina.

Las imágenes han enfurecido a influyentes republicanos en Washington, incluyendo a Newt Gingrich, el ex presidente de la Cámara de Representantes y posible candidato a la nominación presidencial para 2012.

Gingrich dijo que el apretón de manos fue una señal de debilidad para Estados Unidos.

Obama ha dicho que el apretón de manos era lo correcto y que no lo lamenta.

Figure 1.1: Illustration of the different levels of parallelism present in an English (left) and Spanish (right) comparable document pair. The sentences marked with blue solid lines are parallel sentences. The sentences marked with red broken lines are not parallel, but contain sub-sentential fragments that are parallel. Both documents are from the Voice of America News (VOANews).

also assume a parallel sentence pair will not work properly when applied to comparable sentences. As comparable sentences contain both parallel segments as well as (possibly larger) non-parallel segments, special alignment algorithms are required to identify parallelism at sub-sentential level.

1.2 Thesis Summary

This thesis explores a framework to identify translation equivalences from comparable corpora at different levels of granularity, namely document, sentence, and phrase level.

We start with mining comparable document pairs from monolingual document collections using a cross-lingual information retrieval approach. Next we extract comparable sentence pairs from the identified comparable document pairs. For this task, we explore a maximum entropy based binary classifier.

Next, we focus on identifying parallel phrase pairs from comparable sentences. We explore two algorithms to extract phrases from comparable sentences, including the standard Viterbi approach and the proposed extension of PESA alignment [Vogel, 2005] method to cater for comparable sentences. Unlike traditional phrase extraction approaches, the latter method does not rely on the Viterbi path generated by a word alignment model. Rather, it identifies the best target phrase that matches a given source phrase without explicitly generating phrase alignments for the full sentence pair, which makes it well suited for the task of extracting phrase translation pairs in comparable sentences.

The proposed algorithms are evaluated both intrinsically against a manually aligned test corpus for alignment/classification accuracy, as well as extrinsically by using them in building translation systems.

Most of the proposed algorithms use an initial seed parallel corpus to generate lexical resources. Also, the monolingual text collections used as the source for comparable corpora contain documents at different levels of comparability varying from fairly parallel to non-parallel. The effectiveness of the proposed algorithms depends on both these factors. We conduct experiments in different scenarios of data availability, varying from large to small, to no initial parallel data situations. In the last scenario, we explore methods to identify initial word-level alignments from comparable sentences, which can then be used in the framework to extract more resources. Wikipedia is also be used as a special source for comparable data. Previous results [Hewavitharana and Vogel, 2008] show that the most gain from the extracted parallel resources is when the initial parallel data is limited.

1.3 Contributions

This thesis explore a framework to detect and extract translational equivalences from comparable corpora, with a focus on low resource scenarios. This includes languages with low parallel resources and cross-domain translations. Translational equivalences are identified at different levels including document, sentence and phrase level. We summarize the thesis contributions as follows:

- We introduce a hierarchical framework to extract translational equivalences from comparable corpora. Similar frameworks have been used before by previous researchers [Zhao and Vogel, 2002a; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005], but we are unaware of previous attempts to extract parallel phrases from comparable corpora.

- We propose an algorithm to extract parallel phrases from comparable sentence pairs. We demonstrate that this algorithm is more accurate as measured in F_1 than the standard phrase extraction approaches and has significant improvement in translation performance.

- We study the effect of different levels of initial parallel data to the extraction algorithms and demonstrate that the algorithms are robust for low-resource situations.

- We demonstrate that even for languages where no parallel data is available, a translation system can be bootstrapped using parallel data extracted from comparable corpora, and thereby enable machine translation for those languages.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- In Chapter 2 we review the literature on previous approaches to exploiting comparable corpora to extract parallel information.

- Chapter 3 presents the proposed framework for extracting translational equivalences at various levels, and describes the parallel and comparable data resources that will be used in evaluating the proposed approaches.

- In Chapter 4 we describe the sentence extraction process. The evaluation is performed on two language pairs: Arabic-English and Urdu-English.

- The proposed approach for extracting parallel phrases from comparable sentences is presented in Chapter 5.
- In Chapter 6, we used the proposed extraction approaches to bootstrap a machine translation system for a low-resourced language (Sinhala-English) using comparable corpora. We demonstrate that the bootstrapping can be performed even when there is no initial parallel corpus available.
- Chapter 7 gives the thesis contributions and concluding remarks.

Chapter 2

Related Work

There have been several attempts in the past to exploit comparable corpora to identify resources that are useful for NLP tasks. Many of the early efforts focused on learning bilingual word translations from monolingual sources [Rapp, 1995; Fung, 1995; Fung and Yee, 1998; Rapp, 1999; Diab and Resnik, 2002; Koehn and Knight, 2000]. The underlying assumption in these efforts is that a word and its translation appear in similar context in the respective languages, and therefore co-occurrence statistics can be used to detect them. Rapp [1995] used German and English sentences represented as co-occurrence vectors and showed that the vectors are most similar when the order of the words in the two languages is the same. Fung [1995] defines the context in terms of number of different words appearing on both sides of the word of interest. Several other context similarity measures such as term frequency (*tf*) and inverse document frequency (*idf*) have been studied in Fung and Yee [1998].

The problem of finding matching documents in the target language given a query in the source language has been widely studied in cross-lingual information retrieval (CLIR) research [Davis and Dunning, 1995; Oard, 1997]. This usually involves translating the query into the language of the documents using a dictionary and performing the standard information retrieval on the translated query. This method has been used to identify parallel documents, by translating the source document and using the translation to generate a pseudo query in the target language [Zhao and Vogel, 2002b; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005]. Resnik and Smith [2003] find similar article pairs from the Web by exploiting the similarities in the URL structure, and the structure of meta information in the articles among other clues. Parallel articles are selected by using a similarity score calculated based on the word-level alignment. Zhang et al. [2006] and Shi et al. [2006] use similar approaches to extract Chinese-English parallel articles from the Web.

Searching for parallel sentences within comparable corpora has also been tried in different ways. One of the earliest is by Smith [2002], who proposes an algorithm to identify parallel document pairs as well as parallel sentence pairs based on a translational similarity measure called *tsim*. He successfully demonstrates that the proposed approach can be used to extract parallel data with high precision from different levels of noisy data. Both Zhao and Vogel [2002a] and Utiyama and Isahara [2003] extend parallel sentence alignment algorithms to identify parallel sentence pairs. They first find parallel document pairs, and then sentence align them. Munteanu and Marcu [2005] and Fung and Cheung [2004a] work on non-parallel documents, and do not assume that every sentence on source document has a corresponding translation on the target document. Each source sentence is matched against candidate target sentences and the sentence pairs above a certain similarity measure is picked as parallel. Munteanu and Marcu [2005] use a maximum entropy classifier for this task, while Fung and

Cheung [2004a] uses cosine similarity measure. The latter uses an iterative refinement where previously identified sentence pairs are added to the system in later iterations. Munteanu and Marcu [2005] demonstrate that this method can be successfully used to improve the performance of a machine translation system. Abdul-Rauf and Schwenk [2009] uses a slightly different approach. The source documents are translated into the target language using a fully SMT system, and matched against a target document collection using word error rate (WER) as the similarity measure. The approach is used to extract parallel data from out-of-domain documents, which shows a significant improvement on the translation performance.

The first attempt to detect sub-sentential fragments from comparable sentences was proposed by Munteanu and Marcu [2006]. After identifying sentence pairs that have at least partial overlap, they search for parallel fragments using an approach inspired by signal processing. Each word is marked with a score indicating the presence of a corresponding translation in the other sentence. A moving average filter is applied to the values treating it as a signal. Parallel fragments are identified as corresponding to the positive valued segments in the signal. Quirk et al. [2007] later extended this work by introducing generative models that explicitly align comparable sentence pairs. They showed improvements in translation performance when the method was applied on cross-domain test data. Most of these research attempts were concentrated on large document collections containing newswire and political dialogs.

Wikipedia, the online multilingual encyclopaedia, has recently become an attractive source for mining parallel data. Smith [2002] describe an approach to extract parallel sentence pairs from Wikipedia articles. Their approach is similar to [Munteanu and Marcu, 2005], but uses additional features that exploit the specific link structure within Wikipedia.

They demonstrate that the extracted sentence have a significant impact on translation performance, when combined with a mid-sized parallel corpus. Several attempts to extract named entity translations using comparable corpora also exist [Klementiev and Roth, 2006; Sproat et al., 2006].

Kumano et al. [2007] have proposed a phrasal alignment approach for comparable corpora based on the joint probability SMT model [Marcu and Wong, 2002]. While this approach is appealing for low-resource scenarios as it does not require any seed parallel corpus, the high computational cost is a deterrent in its applicability to large corpora. Tillmann and Hewavitharana [2011] present a single unified search algorithm that can be used flexibly on all the different extraction tasks handled by the various algorithms cited above. Translation pair extraction is handled as a chunk-alignment problem with no document-level pre-filtering, and data processed directly at the sentence level. The results indicate significant gains by using the extracted parallel data, especially when using the extracted fragments.

Our work is most related to the work by Munteanu and Marcu [2005] and Quirk et al. [2007]. They both based their work on the Gigaword collections by LDC (Arabic/English and Chinese/English for the former, and Spanish/English for the latter). These collections contain news articles by various multilingual news agencies. There is a high likelihood of finding parallel document pairs in these collections. Our interest is to use comparable corpora to identify parallel resources for language pairs that have less parallel resources such as Urdu-English and Sinhala-English. Document collections such as Gigaword are non-existent for these languages. Further, the likelihood of finding parallel document pairs in these collections is low. Therefore we focus on methods that require fewer initial parallel resources for the task.

We see great potential in using comparable corpora to enhance translation systems. However, most of the previous efforts have focused on extracting entire parallel document pairs or entire parallel sentence pairs from comparable corpora. We think most of the parallel data in comparable corpora are in the sub-sentential level. Only two attempts in the past have been aimed in this direction [Munteanu and Marcu, 2006; Quirk et al., 2007]. Most of the attempts so far have focused on major languages. There has been little effort to use it to cater to languages with less resources. We address these issues in this thesis.

Chapter 3

Detecting Translation Equivalences in Comparable Corpora

As outlined in the introduction, translation equivalences can be identified in comparable data at different levels of granularity, ranging from parallel document pairs through sentence pairs, down to word translation pairs. In this chapter we explore a general framework that can be used to identify these resources. Each step in this framework is used as an initial filtering of the data that can be used in the subsequent steps, and therefore forms a hierarchical extraction process. Similar hierarchical approaches to extract parallel sentences and fragments have been successfully used in the past by several researchers [Zhao and Vogel, 2002a; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005]. We propose extending it further to extracting parallel phrases from comparable sentences.

3.1 Framework

Our primary interest is to extract parallel resources that are helpful in improving the performance of translation systems. Figure 3.1 summarizes the extraction process. We start with two monolingual document collections for the language pairs of interest. The *document selection* process identifies comparable document pairs. If the two document collections are closely related, as in the case of multilingual newswire collections published by the same news organization, this process may yield document pairs that are already parallel. But this is not always true. Therefore we refer to the resulting documents as comparable documents, which may include parallel documents.

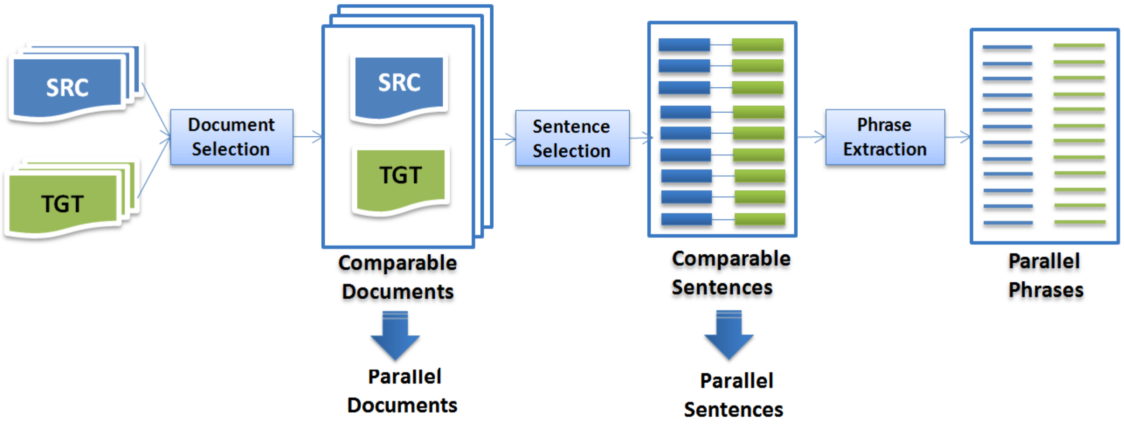


Figure 3.1: Framework to detect translational equivalences

The next step, *sentence selection*, identifies comparable sentence pairs within those comparable document pairs. Here too, some of the sentence pairs could be parallel. We use a Maximum Entropy (ME) classifier for the sentence selection task. Sentence pairs with an alignment score above a certain threshold can be considered parallel and they can be used as new training data for the translation models. Those sentence pairs that are below the threshold, although not fully parallel, may contain parallel phrases that are common to both sentences, as well as phrases that are non-parallel. These sentences are used as input for the next step of the pipeline, *phrase extraction*, to identify parallel phrase pairs. For each language pair, the threshold is determined empirically on a development test set. In chapter 4 we propose a novel approach to extract parallel phrases embedded in comparable sentences.

In this framework we assume that the initial document collections have clear document level boundaries. Although this is a reasonable assumption for clean data collections such as the Gigaword corpora, this might not hold for other data collections. Section **??** looks at a possible approach to handle such situations.

Each of the three steps in the pipeline uses a translation lexicon as the only bilingual resource for the extraction process. This lexicon can be obtained automatically by training a word alignment model using a small seed parallel corpus. It can also be in the form of a manually generated dictionary. For some low-resource languages neither a parallel corpus nor manual dictionaries are available. In chapter 6 we automatically obtain an initial lexicon with available comparable resources, and then bootstrap the resource extraction process to obtain more parallel data.

3.2 Evaluation and Resources

The parallel data obtained through the proposed framework can help to improve translation systems in several ways. It can provide translations for words that are not already covered by the initial seed parallel corpus. It can also provide additional longer n-grams that match with the test data. This helps to improve translation quality by avoiding erroneous re-orderings produced by the decoder when using a collection of shorter phrases. Additional parallel sentences can be added to the training data to train improved word alignment models. Additional parallel phrase pairs can be used directly in the decoding process.

We evaluate each extraction process intrinsically for its accuracy. This is done by comparing the results with manually generated data, whenever such data is available. Standard evaluation metrics such as precision, recall and F-Score are used for the evaluation. The usefulness of the extracted resources will also be evaluated in an end-to-end statistical machine translation systems. We build two translation systems: a baseline system with the initial parallel corpus, and a second system by combining the newly extracted parallel data and the initial parallel corpus. The two systems are evaluated on their performance when translating an unseen test set. Standard MT evaluation metrics BLEU [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005], and TER [Snover et al., 2006] are used for the evaluation. Test sets from previous NIST MT evaluations[1] are used as development and unseen test sets.

Arabic-English

To validate the proposed approaches, we primarily experiment with Arabic-English language pair. The choice of Arabic was somewhat arbitrary, mainly motivated by the availability of a baseline Arabic system and other resources. As the comparable corpus for Arabic-English, we use the Arabic Gigaword corpus and English Gigaword corpus. Each of these corpora contains a large collection of documents published by a number of news organizations, including Agence France Presse (AFP) and Xinhua, which provides newswire services in multiple languages. The articles are clean text with similar document structures and therefore ideal for

[1] http://www.itl.nist.gov/iad/mig/tests/mt/

initial experiments. The Gigaword collection has been used in previous experiments reported in the literature [Munteanu and Marcu, 2005]. Gigaword collections are also available for Chinese, Spanish and French.

A fairly large amount ($\sim$ 120 million English tokens) of Arabic-English parallel data is already available. We use this corpus to simulate low-resource scenarios by using only part of the data as initial parallel data, from which the lexicons used in various experiments are generated. We generated three parallel corpora with 1/3, 1/9 and 1/27 of the original corpus size by picking every n^{th} sentence pair from the full corpus. To simulate even more low-resource situations, we created two more lexica with roughly one million and one hundred thousand English words. These corpora match the resource levels for many languages. We will also use data from BTEC [Kikui et al., 2003] corpus. This corpus contains conversational data from the travel domain which is from a different genre than the document collections. Compared to other corpora it is quite small ($\sim$ 190 thousand English tokens).

One of the aims of the proposed work is to enable machine translation for low-resource languages such as Urdu, Pashto, Dari, and Sinhala by using the parallel resources extracted from comparable corpora. It is for these languages that we hope to see the most benefit from the extracted resources. Large monolingual data collections, such as the Gigaword corpora, are not readily available for these languages. Therefore we started a data collection effort for a number of low-resource languages that can be utilized in the experiments. Details of this effort are given in the next section.

3.2.1 Comparable Data Sources

Our data collection effort focused on two major sources of comparable data for low resource languages: news stories published by major newswire sites, and Wikipedia. There are other possible sources such as multilingual websites maintained by governments and business organizations, operating manuals, technical documentations, the Bible, etc. We utilized them whenever they were easily accessible, especially when no other parallel data was available.

Newswire Articles

Newswire services which disseminate news in multiple languages are ideal sources of comparable corpora. Several such sources exist, such as BBC, Voice of America (VOA), Agence France Press (AFP) and Xinhua. We focused on two such sources, BBC and VOA, because they publish news articles in a number of low-resource languages.

BBC currently publishes news articles in 32 languages on its website BBC News Online[2]. These include widely spoken languages such as English, Chinese and Arabic as well as less commonly spoken languages such as Urdu, Pashto, Nepali and Kinyarwanda. VOA[3] publishes news in 45 languages including many African and East Asian languages. The amount of articles available in these languages varies a lot. However, in many cases, articles can be found in multiple languages that describe the same event. Often these are translations, albeit not completely parallel, of the original English article.

During the past few years we have crawled BBC and VOA news articles in English, and several other languages. We have also crawled several other news websites for Sinhala, Tamil

[2] http://www.bbc.co.uk/worldservice/languages/

[3] http://www.voanews.com

and English language articles. The organization of these multilingual news websites varies greatly, making the designing of crawlers difficult. Even within the same news organizations, the site architecture may be different among languages. For example, BBC Urdu language page has a different structure than the BBC English language portal. This may be because the two sites are designed by two different groups within the organization. Further, the page naming conventions, updating schedules, etc varied greatly among the websites. The format of a particular website has changed over the years, which required designing different crawlers for different time ranges, especially when crawling for archival data.

We crawled data for English, Dari, Farsi, Hindi, Pashto, Sinhala, Tamil and Urdu. A considerable effort was devoted to clean the downloaded articles by removing non-text portions and resolving formatting & encoding issues. Sentence boundary detection algorithms were used to split large paragraphs into sentences. Finally, the articles were converted in to SGML format, similar to what LDC uses. Table 3.1 lists the amount of articles collected.

Source	English	Dari	Farsi	Hindi	Pashto	Sinhala	Tamil	Urdu
BBC	300	-	14	38	7	*[a]	*[b]	47
VOA	280	9	-	47	9	-	-	18
Other	24	-	-	-	-	26	10	-

[a] Although BBC has a Sinhala language page, the content are only in English.

[b] BBC Tamil language news is presented in a single article as an aggregate of news reports for a given day.

Table 3.1: Amount of data collected for each language (in thousands of articles)

The amount of available data varies significantly among different languages. Urdu and Hindi have a large collection of comparable documents compared to other languages. However, when compared with the Gigaword corpora for Arabic and English, these collections are orders of magnitude smaller. Therefore, our proposed methods when used to extract data from low resource situations should also take this fact into account. i.e. focusing on only high precision data extraction in low resource situations may only yield fewer amount of data. A balance between precision and recall will have to be maintained.

The amount of available initial parallel data also varies among these languages. For some languages such as Sinhala and Tamil, no significant amount of sentence aligned parallel data exists.

Under the REFLEX-LCTL program, a Linguistic Data Consortium (LDC) has been collecting resources for 19 Less Commonly Taught Languages (LCTL) [Simpson et al., 2008] which includes Urdu, Thai, Hungarian, etc. These language packs include monolingual text, parallel text (with English) and, for some languages, other resources such as lexica and word segmenters. The parallel corpus typically consists of news stories, which were then translated manually. With only a few hundred documents and about 200k tokens these corpora are very small compared to language pairs such as Arabic-English and Chinese-English. The Urdu language package was released for the NIST Open-MT evaluation campaign in 2008.

Due to the availability of this initial parallel corpus, in our experiments we selected Urdu-English for the *small data scenario.* Sinhala-English was selected for the *no parallel data scenario* as explained in section 3.2.2.

Wikipedia as a Comparable Corpus

Wikipedia, the web-based free multilingual encyclopedia where articles are generated through public collaboration, currently contains more than 9 million articles in 250 languages. About one quarter of the articles are in English, and over 90 languages have more than 10K articles. Wikipedia contains a vast collection of manually defined terms, concepts and relations which makes it particularly interesting resource for NLP tasks.

Wiktionary - a companion of Wikipedia is a multilingual dictionary generated in the same way as Wikipedia. This includes not only dictionary entries, but also a thesaurus and phrase books. All the content in Wikipedia, including Wiktionary, is available for free download.

Wikipedia has some special features that makes it especially interesting as a source of comparable documents. We can view two articles in different languages that describe the same concept as comparable documents. Many Wikipedia articles link directly to the counterpart articles describing the same concept in other languages. These links make it trivial to extract a high quality parallel corpus made up of just the titles. However, because the articles are not available in all the languages, and because these link have to be made manually, not all articles have links between a given language pair. Other useful information such as translations of named entities, technical terms, etc. could also be extracted from titles or other templates alone.

Articles on a particular topic are often created independently, and are available only in a few languages. The level of detail in them varies drastically. The general outline and selection of content in an article could also be very different. Therefore, extracting reliable translation equivalences from the articles becomes a challenging task.

Table 3.2 lists the number of articles available on Wikipedia and Wiktionary for the languages of interest.

We use Wikipedia article titles and Wiktionary to obtain a translation lexicon when other parallel data is not available. Details of this process are described in chapter 6.

Source	English	Dari	Farsi	Hindi	Pashto	Sinhala	Tamil	Urdu
Wikipedia	3114	-	20	42	2	2	20	12
Wiktionary	1510	-	16	6	1	1	101	5

Table 3.2: Number of articles (in thousands) available in Wikipedia and Wiktionary for the languages of interest

We did a preliminary study on extracting translation equivalences from Arabic, German and English Wikipedia. Using Arabic-English and German-English pairs we extracted 43K and 430K sentence pairs from the titles of the articles. After filtering pairs that differ significantly in length, we obtained a parallel corpus of 40K and 420K sentences. A manual inspection of a subset of the extracted content found that they are of high quality.

In the following sections we explain how we will incorporate parallel data extraction methods proposed in chapters 4 and 5 to build translation systems for low-resource languages.

3.2.2 Data Availability Scenarios

An important goal of this thesis is to improve existing translation system using parallel resources extracted from comparable corpora and to allow building translation systems even in the case of very limited parallel data. Therefore, the techniques developed for learning translation equivalences need to be evaluated in the context of machine translation. To better understand the impact of information extracted from comparable corpora, we will conduct experiments in 3 different levels of data availability:

Large Data Scenario Fairly large amount of parallel data (tens of millions of words) exists, enough to build a translation system of high accuracy. Some example language pairs in this category are Arabic-English (120 million words), Chinese-English (¿ 100 million words), and German/French/Italian/Spanish-English (¿ 50 million words). Additional information from comparable corpora may give a modest improvement. We expect that in this situation comparable corpora will be most beneficial when the application domain is not well covered by the parallel corpus, where the comparable corpus can provide additional coverage. We use Arabic-English under this scenario.

Small Data Scenario Some amount of parallel data (hundreds of thousands of words) exists, but not enough to build a strong translation system. Some example language pairs in this category are Urdu-English (one million words), Hindi-English (one million words) and Pashto-English (eight hundred thousand words). Adding information extracted from comparable corpora can result in significant improvements in translation quality. Urdu-English language pair is used in this scenario.

No parallel data is available No sentences-aligned parallel corpus is readily available to build a machine translation system. In this case, exploiting comparable corpora will enable new translation systems which so far are not possible. This is also the most difficult case, where bootstrapping is required. We use Sinhala-English language pair as an example for this scenario.

Large Data Scenario

Our existing translation systems (e.g. [Hildebrand et al., 2008]) fall into this category. Here, we use the extracted parallel data to enhance the existing translation system. This can be done in several ways. Parallel sentence pairs can be used as additional training data to the translation lexicons, and phrase tables. Phrase translation pairs can be directly added to the phrase table to be used in the translation decoder.

Another important aspect of using these resources is to translate unknown words, i.e., words in a source sentence that are not found in the translation model. When unknown words are encountered, we search in comparable corpora for phrases that contain the unknown words, and extract translations for them. This will help to improve the coverage of the translation model.

Additionally, we will try to answer the following questions:

- How much performance improvement can be achieved by using the extracted resources?
- What is the difference in performance between the same amount of parallel data and parallel resource extracted using comparable data?

To do so, we will use the UN corpus, parallel news corpus, and parallel resources extracted from comparable corpora.

Small Data Scenario

In this scenario, the available parallel data is not enough to build a strong translation system with high vocabulary coverage, even for limited domain translation tasks. The available parallel data will be used to train an initial, limited translation lexicon. This lexicon will then be used to identify parallel resources from comparable corpora.

Most of the less commonly spoken languages fall into this data scenario. Among the languages in which we have collected comparable data so far, Urdu, Pashto and Dari are some examples. The existing parallel data amounts to 1M words each for Dari-English, Pashto-English and Urdu-English. Urdu has been used in NIST MT Eval evaluations in

2008 and 2009, and therefore established test sets are also available. We will use them to evaluate the performance.

We will try to answer the following questions:

- How much performance improvement can be achieved by using different resources extracted?
- How much improvement can be achieved in terms of test set coverage?

No Parallel Data Scenario

This is the most challenging situation, but also the case where we expect to gain the most benefit from our techniques. From the languages we have collected data, Sinhala, Tamil and Farsi languages fall into this category. With current translation technology, no data-driven translation system can be built for these languages. However, in some cases, a small manually generated translation dictionary may be available. In the absence of a dictionary, we obtain an initial lexicon as explained in chapter 6. As no test set is available for these languages, we manually generate a small test set to evaluate the system.

Here, we try to answer the following question:

- When starting with a noisy parallel corpus, can bootstrapping be used to improve MT performance?

Chapter 4

Detecting Comparable Sentence Pairs

The first step in the hierarchical extraction framework is to detect the comparable document pairs. The identified comparable document pairs are then analysed to identify comparable sentence pairs. In this chapter we describe how each of these steps are performed. We evaluate the extraction process using two language pairs: Arabic-English and Urdu-English. Part of the work described here has been presented earlier in [Hewavitharana and Vogel, 2008] and [Tillmann and Hewavitharana, 2011].

4.1 Mining for Comparable Document Pairs

In this step we match comparable document pairs in the two monolingual document collections. For each source language document, we want to select target language documents that contain the best matching content. This can be performed using the cross-lingual information retrieval method [Oard and Dorr, 1996; Grefenstette, 1998]. A given source document is translated into a sequence of target language words. This translation is then used in a bag-of-word representation to produce a pseudo query, which is used against a target language document collection to mine candidate documents. The retrieved documents are expected to have content comparable to the source document. We use the *Lemur* information retrieval toolkit [Ogilvie and Callan, 2001] for the retrieval task.

There are several options to translate the source document into the target language. A standard machine translation system (e.g. *Google Translation*) can be used to generate the translations. Another option is to use a translation lexicon or a dictionary to translate each source word into target words. Because at the end we transform the translation into a bag-of-word representation, this second method is more suitable for the task. In this thesis, we work with low-resource languages which do not always have enough resources to build a full translation system. Therefore using a translation lexicon for the task is more appropriate. For the experiments described in this chapter, we use lexica of different sizes to observe its effect on the retrieval process. The document selection algorithm is as shown in Algorithm 1.

Algorithm 1 Algorithm to select comparable document pairs

```
Input: Source document collection $\mathcal{D}_S$, target document collection $\mathcal{D}_T$,
    source-to-target lexicon $Lex_{S \to T}$, and stopwords list $StopWords$
Output: List of comparable document pairs $Docs$
$Docs = \{\}$
for each document $\mathbf{S}$ in $\mathcal{D}_S$ do
    $Query = \{\}$
    for each sentence $S_1^J = s_1 \ldots s_J$ in $\mathbf{S}$ do
        for each word $s_j$ in $S_1^J$ do
            $Query = Query \cup \{t_k | t_k = Lex_{S \to T}(s_j) \text{ and } k \leqslant 5\}$
    $Query = Query - StopWords$
    $Docs = Docs \cup \{(\mathbf{S}, \mathbf{T}_i) |\text{ Find } \mathbf{T}_i \text{ in } \mathcal{D}_T \text{ using } tf.idf \text{ similarity, and } i \leqslant 20\}$
return $Docs$
```

4.2 Detecting Comparable Sentence Pairs

Once comparable document pairs are identified, the next step involves detecting which of the sentences in the document pair are translations of each other. If the document pair is parallel, each sentence in the source language document will have a translation on the target document (it is possible that multiple sentences in the source language document can correspond to one or more sentence in the target document, and vice versa). In such instances, identifying sentence pairs can be addressed as finding the sentence alignment between the two documents. A dynamic programming based approach [Gale and Church, 1991] can be used to perform the alignment.

However, typically only a few of the sentences in a comparable document pair are mutual translations. The majority of the sentences may not have a full or partial translational equivalent in the other document. Further, for the sentences that have a translation on the other side, the order in which they appear can be different in the document pair. A dynamic programming based sentence alignment methods will not give the desired results. The sentence detection method should be able to identify comparable sentence pairs that appear anywhere in the document.

One way to avoid this (reordering) problem is to take every combination of sentence pairs in a document pair (i.e. Cartesian product) and check if they are comparable. Obviously this is computationally expensive. A filtering process can be used to remove unpromising sentence pairs as a first step and a more detailed detection can be performed on the second step as in [Smith, 2002].

Munteanu and Marcu [2005] have demonstrated that a maximum-entropy based classifier can be successfully used to detect parallel sentence pairs. We use the same principle here, but with the following distinctions:

- Our focus here is not only to identify parallel sentences, but also comparable sentences that have partial translations, which are then used to detect parallel phrases in chapter 5. As such we experiments with a set of features that takes this into account. Details of the classifier are given in section 4.2.1.

- When training the classifier, we use manually annotated comparable sentence pairs, in comparison to the synthetically generated training examples used in [Munteanu and Marcu, 2005].

The results in section 4.3.4 show that these changes help to extract additional useful sentences.

The classifier approach is appealing to a low-resource scenario, because the features for the classifier can be generated with minimal translation resources (i.e., a translation lexicon). When enough parallel resources are available to reliably build a translation system, we can explore other avenues to detect comparable sentence pairs.

4.2.1 Maximum Entropy Classifier

The Maximum Entropy (ME) principle had been widely studied and applied to several NLP tasks. It uses the training data to set constraints on the model in the form of feature functions. Feature functions express desired properties of the training data distribution that should be matched in the learned model. It can be shown that there is always a unique maximum entropy model that satisfies the constraints imposed by the training data. The maximum entropy classifier probability can be defined as follows:

$$Pr(c_i|S,T) = \frac{1}{Z(S,T)} \exp\left(\sum_{j=1}^{n} \lambda_j \cdot f_{ij}(c_i,S,T)\right) \tag{4.1}$$

where $S = s_1^J$ is a source sentence of length J and $T = t_1^I$ is a target sentence of length I. c_i is a variable representing the classes into which we want to classify the sentences. For this task, there are two classes of sentences: c_0 indicating the sentence pair is *non-parallel* and c_1 indicating the sentence pair is *comparable.* $p(c|S,T) \in [0,1]$ is the probability where a value $p(c_1|S,T)$ close to 1.0 indicates that S and T are translations of each other. $f_{ij}(c_i,S,T)$ are feature functions that are co-indexed with respect to the class variable c_i. Notice that, in the feature vector for a given sentence pair, each feature f_j appears twice, once for each class c_0 and c_1. The parameters λ_j are the weights for the feature functions obtained by optimizing on a training data set. $Z(S,T)$ is an appropriately chosen normalizing constant. i.e. $Z(S,T) = \exp\left(\sum_j \lambda_j \cdot f_{0j}(c_0,S,T)\right) + \exp\left(\sum_j \lambda_j \cdot f_{1j}(c_1,S,T)\right)$.

Features

The feature functions for the ME classifier should distinguish between the two classes of sentences we are interested in: *non parallel* and *comparable.* The primary factor that decides this is how much of each sentence is a translation of the other. This can be quantified by computing the lexical overlap between the two sentences according to a translation lexicon. When a seed parallel corpus is available, we can automatically generate word alignments using a model such as IBM-Model-1 [Brown et al., 1993] which will produce a probabilistic translation lexicon. Due to the non-symmetry in the word alignment models, two alignments can be generated, one for source-to-target direction and the other for target-to-source direction. The lexica produced by the two directional alignment models typically contain different distributions. Therefore we can define two sets of lexical features using the two lexica.

The Viterbi alignment of IBM Model-1 between a sentence pair can also give an indication about its parallelism. A large number of unaligned words indicates that the two sentences are not parallel. On the other hand, a long sequence of aligned words indicates that there are parallel segments in the sentence pair. However, our extraction process involves classifying a large number of source/target sentence pairs. Generating the full Viterbi alignment for each of the sentence pairs is a computationally expensive task. Therefore, we use the probabilities from the translation lexica to determine the alignment links between the sentence pair, and define a set of coverage features based on the approximated alignment. A probability threshold $\epsilon = 5 \cdot 10^{-4}$ is used to determine the word alignment links.

Another important consideration is the length of the sentences. A pair of parallel sentences typical have similar lengths. A pair of comparable sentences can have different lengths. A source/target sentence pair with only one embedded parallel phrase pair may be considered as a comparable sentence pair. However, our interest is not in such sentence pairs. We are interested in identifying sentence pairs that contain the same information, with the possibility of a few additional words in either of the sentences. Hence, the lengths of the sentences are still an important consideration and they should be in the same range. We define the following set of feature functions:

Lexical Probability (f_1,f_2) : Two lexical probability scores based on IBM Model-1 are computed for source-to-target ($P_{S \to T}$) direction and target-to-source ($P_{T \to S}$) direction respectively, as follows:

$$P_{S\to T}(S,T) = \frac{1}{J}\prod_{i=1}^{I}\sum_{j=1}^{J} p(t_i|s_j)$$

$$P_{T\to S}(S,T) = \frac{1}{I}\prod_{j=1}^{J}\sum_{i=1}^{I} p(s_j|t_i)$$

Maximum Fertility (f_3, f_4) : We define the fertility of a source word s as the number of target words $t \in T$ for which the lexical probability $p(s|t) > \epsilon$. A word with high fertility is typically indicative of the presence of non-parallel text. We pick the highest fertility value for the sentence as a feature. Target fertility is defined similarly.

Coverage (f_5, f_6) : A source word s is covered if there is a target word $t \in T$ such that the lexical probability $p(s|t) > \epsilon$. Target is defined similarly. We use the number of source and target positions covered in the alignment path as features. A higher number of covered positions indicates parallelism.

Consecutive Coverage (f_7, f_8) : Consecutive source coverage is the length of the longest consecutive subsequence of covered (i.e. $p(s|t) > \epsilon$) words in the source sentence. Consecutive target coverage is defined similarly. A long sequence of covered word positions typically indicates the presence of parallel text.

Length Ratio (f_9, f_{10}) : Ratios between source length and target length are included as features: J/I and I/J. These are similar to the length penalty features used in SMT decoding.

Length Difference (f_{11}) : Difference between the source sentence length and the target sentence length: $J - I$.

Transliteration (f_{12}) : For each source word s_j, we compute its Levenshtein distance with respect to all target words $t_i \in T$. We use Buckwalter transliteration scheme for languages that use Arabic script. Using the edit distance, we compute the following feature function:

$$f_{12} = \left[1.0 - \max_{\substack{1 \leqslant i \leqslant I \\ 1 \leqslant j \leqslant J}} d(tr(s_j), t_i) \right]$$

where, $d(t', t)$ is the Levenshtein distance between the transliterated source word t' and the i-th target word t_i.

Intersection (f_{13}) : The intersection is defined as the number of alignment points that overlap on the alignment paths for the two directions, source-to-target and target-to-source. The alignment path are computed by simply selecting the source and target word pairs (s,j) for which the lexical probability $p(s|t) > \epsilon$.

Empirically, we have found that normalizing the feature values with respect to either source length J or target length I improves the classification accuracy. Therefore we apply the sentence length normalization to the features described above, and define an auxiliary feature vector F as follows:

$$F = (f_1, f_2, f_3/J, f_4/I, f_5/J, f_6/I, f_7/J, f_8/I, f_9, f_{10}, f_{11}/J, f_{12}/J, f_{13}/J)$$

The features f_1 and f_2 have already been normalized by definition. We used the auxiliary feature vector F to train the ME classifier. We experimented with several other features including variations of the above features that discounted high frequency words, features to model unaligned words, and subsets of the above features. We found that this particular feature combination (f_1 to f_{13}) gives the best performance with respect to F_1 score on development data.

These features are straightforward and easy to compute. Only a probabilistic transliteration lexicon, one each for source-to-target and target-to-source directions, is needed to compute all but the length and transliteration features. We avoided using more complex features (e.g. part-of-speech tags, dependency structures, etc) because they require additional resources (e.g. parsers) that may not be readily available to low resource languages that we are interested in applying these techniques. Those features are computationally expensive to compute, as well.

4.3 Experiments

Using different sizes of parallel corpora mentioned in the previous section, we generated word-alignments by running GIZA++ [Och and Ney, 2003] up to IBM model 4. For each corpus a translation lexicon was extracted from the resulting word-alignments. A lexicon may contain multiple translations for a given source word or a target word. Table 4.1 gives statistics of the Arabic-English and Urdu-English lexica that are used for the experiments in the following sections.

4.3.1 Document Extraction

We followed the cross-lingual information retrieval approach described in section 4.1 to extract comparable document pairs from Arabic-English and Urdu-English document collections. The Arabic-English data comes from the Arabic Gigaword and English Gigaword corpora. The Gigaword corpus contains news articles from several sources. For the experiments, we used data from two sources: Xinhua and AFP. Arabic Gigaword and English

Lexicon	**Corpus size**	**Vocabulary**	**Word Pairs**
Arabic-English			
Lex-120M	120M	592,133	8,949,604
Lex-40M	40M	375,162	4,618,977
Lex-13M	13.3M	232,652	2,313,551
Lex-4.5M	4.5M	142,527	1,197,740
Lex-1M	1M	66,468	400,213
Lex-100K	100K	19,066	69,958
Lex-BTEC	20K	17,008	66,067
Urdu-English			
Lex-Ur-En	1M	35,715	234,568

Table 4.1: Characteristics of Arabic-English and Urdu-English translation lexica used for sentence selection experiments. For each lexicon, the corpus size (tokens on the English side), vocabulary size (source side) and the number of entries in the lexicon are given.

Gigaword contain articles from the same time period. The articles are arranged according to the year and month of publication. Each article is marked for the document and segment (paragraph) boundaries. It also contains meta information giving the publication date, title and other information. We use this information to restrict the search space when detecting comparable document pairs.

The Urdu-English comparable data was collected from two news sources: VOA and BBC. The collection contains news articles from the period 2004–2011.

Using the Indri search engine [Strohman et al., 2005] we built an index for the English document collection. The Krovetz stemmer [Krovetz, 1993] and an English stop-words list (4K words) provided with the toolkit were used in the process. Each word in the source document was independently translated into an English word using the source-to-target lexicon, allowing up to 5 top translations per source word. An English query was then generated using all the translated words as a bag-of-words. This query was used to retrieve matching documents from the English document collection using the Indri search engine. For each query, top 20 matching documents were retrieved using the *tf.idf* term-weight model.

To evaluate the accuracy of the retrieval process we conducted an oracle experiment. We selected 670 parallel document pairs from an LDC (Linguistic Data Consortium) Arabic-English parallel corpus. These are newswire stories from Xinhua News Agency from the period 2001–2002. They follow the same formatting as the Gigaword Corpora. We added the English parallel documents to the Xinhua portion of English Gigaword and built an index. For each Arabic parallel document we obtained a list of matching English documents. Figure 4.1 (a) summarizes the performance of the retrieval process for different lexica. It gives the cumulative percentage of known-parallel documents retrieved (i.e., recall) at each rank in the list.

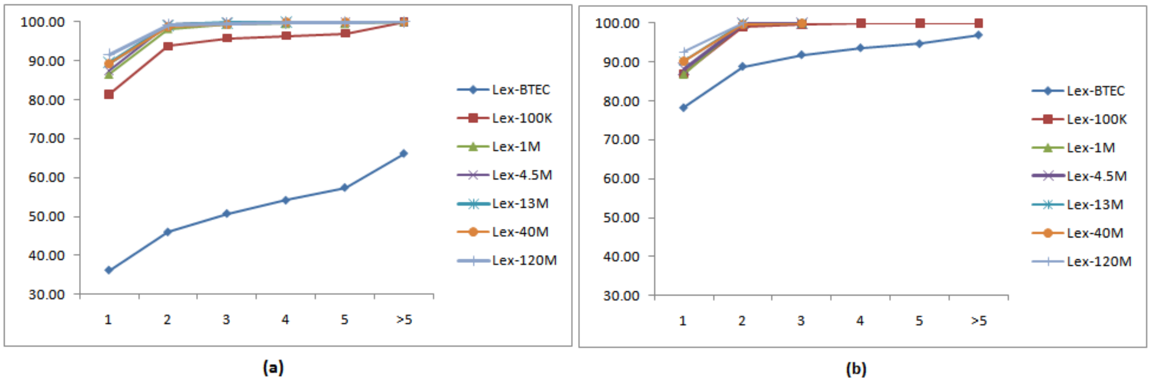


Figure 4.1: Cumulative percentage of parallel documents retrieved at each rank (a) without restricting (b)with a ± 5 day window

We notice that for most of the lexica, close to 90% of the documents are correctly retrieved at rank 1. The size of the lexicon had a minimal effect on the performance, except for Lex-BTEC, which is quite small in size and coverage as it is from a different genre than the test documents. For the five largest lexica, close to 100% of the documents were within the top 10 positions.

Because the parallel documents and the monolingual Gigaword collections are from the same news agency, and within the same time period, there is a possibility that the combined collection may contain duplicate documents. This will, in some instances, force the English parallel document to be ranked lower. We made no attempt to remove these duplicates from the collection.

Another artifact of using a news article corpus is the presence of several related articles with different levels of detail. When news breaks, the first article explains it in less detail and could be rather short. During the course of time, more details are added and new articles are released. In multilingual news organizations, these are then translated into several languages. Therefore, there can be a number of articles in the collection that are comparable to the source article we are interested in. We manually inspected the top 20 documents extracted for some of the Arabic documents and noticed the presence of this phenomenon. Length of the document can be used to guide the process of selecting the best matching document, but we do not attempt to do it here because the sentence selection method processes each pair of source/target sentence separately. Further, we noticed that some of the top ranked documents are exact translations of the source document. We identify them to produce a parallel corpus. This step is explained in section 4.3.3.

News articles that describe the same incident typically occur within a few days of each other. As each article in the news collection contains a publication date we can use this information to further constrain the retrieval process to a specific time period. Now, when retrieving matching English documents for a given Arabic document, we specify a time window of ± 3 days around the publication date of the Arabic document. Lemur will only consider an article which falls within the time window, thus making the retrieval process faster. The restricted search also improves the accuracy, especially for *Lex-BTEC* as shown in Figure 4.1 (b).

We also noticed that the presence of duplicate query terms affects the speed and accuracy of the retrieval process. Removing duplicate query terms increased the accuracy. This could be due to the fact that high frequency words appear only once, thereby reducing the dominance against low frequency words.

The real usage of the document extraction system is slightly different from the evaluation we conducted. In the evaluation, we were looking for a document which was already present in the collection, which is a form of known-item retrieval. When using the system to identify comparable document pairs, we have to make the decision if the top ranked documents are, in fact, comparable. At this step we emphasize recall and assume all top ranked candidates are comparable to the source document.

4.3.2 Training and Testing the Classifier

To train the classifier, we used a manually annotated collection of sentence pairs with varying degree of parallelism. These sentences were obtained as follows.

We ran the sentence pair extraction process on a sample set of the comparable data. In place of the classifier, we used the average of the two lexical probability scores (f_1 and f_2) to score each comparable sentence pair. From the resulting collection, we sampled sentence pairs uniformly across the entire score range (10 bins for the score range [0.0 - 1.0]). Within each score range, the sampling process picked sentences uniformly across all lengths (10 bins for lengths between 1 word and 100 words). This sampled sentence pair collection was then passed on to a human annotator (1 or 2 annotators per each language pair) who is a native speaker in the source language and fluent in the target language. The annotator marked each sentence pair with the percentage of source and target words that have a corresponding translation in the other sentence (Instead of the actual percentage of parallel words, we used 5 bins: up to 0%, 25%, 50%, 75% and 100%). A sentence pair is considered parallel (i.e. positive example) if at least 75% of the words have a corresponding translation on the other side. Otherwise it is considered to be non-parallel (i.e. negative example). The resulting sentence pairs were used for the training and testing of the classifier.

To build the ME classifier, we used the YASMET[1] toolkit. It uses the Generalized Iterative Scaling (GIS) algorithm to obtain the weights λ_j in equation 4.1. We used the trained ME model to classify the test set, and observe how many instances are classified correctly.

We measure precision, recall and F_1-score as follows. Let

P = Number of parallel examples in the test set

P_c = Number of instances that are classified as parallel by the classifier

P_t = Number of truly parallel instances in the set identified as parallel

[1] http://www.fjoch.com/YASMET.html

$$\begin{aligned}
Precision &= \frac{P_t}{P_c} \cdot 100 \\
Recall &= \frac{P_t}{P} \cdot 100 \\
F_1 score &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}
\end{aligned} \tag{4.2}$$

The classifier performance is demonstrated for two language pairs: Arabic-English and Urdu English.

Arabic-English

We used 800 manually annotated Arabic-English comparable sentence pairs to train the classifier. These sentences were selected from the Arabic Gigaword corpus and English Gigaword corpus. Roughly 80 percent of the data set was used to train the classifier and the rest was used for testing.

Features	**Precision**	**Recall**	**F_1-score**
Lexical Probabilities $P_{S \to T}$,$P_{T \to S}$	93.96	95.84	94.89
+Src/Tgt Max Fertility	93.81	95.84	94.81
+Src/Tgt Coverage	94.56	95.51	95.03
+Src/Tgt Cons. Coverage	95.29	95.01	95.15
+Length Features	96.11	94.68	95.39
+Transliteration	96.11	94.51	95.39
+Intersection	96.43	94.51	95.46

Table 4.2: Classifier evaluation in terms of Precision/Recall/F_1-score on the manually annotated data for Arabic-English. Each line indicates a classifier which includes all the features from preceding lines.

In Table 4.2, we present the classification results for the Arabic-English ME classifier. It demonstrates the effect of using different feature sets. The simplest model (i.e. the fist line) uses just the lexical probability features f_1 and f_2. Each successive line adds an additional feature (or two, if the feature can be generated using the lexica trained for two directions). The final classifier (i.e. the last line) uses all the features including the alignment intersection feature.

Including additional features does not always improve the classification accuracy. For example, the Transliteration feature (f_{12}), which was added to give a boost for sentence pairs containing named entities, did not improve the F_1-score for Arabic-English. The intersection feature improved the precision, but not the recall. Overall, the additional features help improve the F_1-score by 0.6 percent over just using the lexical probabilities. For the extraction experiments reported later in this chapter, we always used the classifier with the full feature set.

To see the effect of the threshold of comparability in training data at which we set the class boundaries (at 75% in Table 4.2) on classifier performance, we conducted another experiment where we lowered the threshold to 50% level. The results are shown in Table 4.3.

Features	**Precision**	**Recall**	**F_1-score**
Lexical Probabilities $P_{S\rightarrow T}$,$P_{T\rightarrow S}$	65.86	79.82	72.17
+Src/Tgt Max Fertility	66.42	79.50	72.37
+Src/Tgt Coverage	66.57	79.44	72.44
+Src/Tgt Cons. Coverage	67.67	79.10	72.94
+Length Features	67.93	78.96	73.03
+Transliteration	69.11	78.95	73.70
+Intersection	69.46	77.75	73.37

Table 4.3: Classifier evaluation in terms of Precision/Recall/F_1-score on the manually annotated data for Arabic-English with class boundary set at 50% comparability level. Each line indicates a classifier which includes all the features from preceding lines.

We notice that the precision, recall and F_1 drops considerably due to the inclusion of more noisy training data into the parallel class. The general effect of different features remains the same as in Table 4.2, where the lexical probability features f_1 and f_2 have the largest effect on classification accuracy.

To observe the effect of having different sizes of initial parallel data on the classification task, we trained the classifier features with each of the lexica described in section 3.2. The size of the parallel data that was used to train these lexica range from 120 million words to 100,000 words. The corpus statistics are given in Table 4.1. All the classifiers share the same set of feature functions as explained in section 4.2.1. The classification results are given in Table 4.4.

Lexicon	**Precision**	**Recall**	F_1**-score**
Lex-120M	96.43	94.51	95.46
Lex-40M	94.13	88.30	91.12
Lex-13M	91.59	83.47	87.34
Lex-4.5M	91.17	77.52	83.79
Lex-1M	91.12	69.33	78.74
Lex-100K	89.06	48.71	62.98

Table 4.4: Classifier evaluation in terms of Precision/Recall/F_1-score on the manually annotated data for Arabic-English using lexica trained on different sizes of initial parallel data.

The results indicate that the classifier achieves high precision even with the lexica trained on smaller amounts of parallel data. However, the recall drops as the lexicon becomes smaller. This is an encouraging results as we apply this technique to obtain parallel data for low resource languages where the amount of available parallel data is small.

Urdu-English

To train the Urdu-English classifier we used a batch of 600 manually annotated comparable sentence pairs. The annotation was performed in a similar manner to Arabic-English sentences described above. Here too, we used roughly 80 percent of the data set for training the classifier and the rest for the evaluation. The amount of initial parallel data available for Urdu-English is substantially smaller than for Arabic-English. The translation lexicon that was used to generate the features were trained on 1 million words, compared to 120 million that was used to train the largest Arabic-English lexicon.

Table 4.5 shows the classification results for the Urdu-English classifier. The effect of different features on the classification performance shows a pattern similar to the Arabic-English classifier. Here too, the largest contribution comes from the two lexical probability features f_1 and f_2. The other features improve the F_1-score by almost 2 percent. For the extraction experiments we used the classifier with the full feature set.

Features	**Precision**	**Recall**	**F_1-score**
Lexical Probabilities $P_{S\to T}$,$P_{T\to S}$	85.10	93.46	89.08
+Src/Tgt Max Fertility	87.57	90.80	89.16
+Src/Tgt Coverage	85.47	92.64	88.91
+Src/Tgt Cons. Coverage	87.48	91.41	89.40
+Length Features	91.12	90.18	90.65
+Transliteration	91.10	89.98	90.53
+Intersection	91.17	90.80	90.98

Table 4.5: Classifier evaluation in terms of Precision/Recall/F_1-score on the manually annotated data for Urdu-English. Each line indicates a classifier which includes all the features from preceding lines.

4.3.3 Sentence Extraction

We use the ME classifier trained in the previous section to extract comparable sentence pairs from the document pairs identified in section 4.3.1. In Arabic-English experiments we are interested in observing the effect of extracted data from different sizes of initial parallel data. As such we want to obtain the best possible sentence pairs. For the experiments reported below, we used the lexicon Lex-120M for both document selection and sentence selection steps. The translation lexicon for Urdu-English was trained on all available parallel data.

For each comparable document pair, we produce the Cartesian product of sentence pairs and apply the ME classifier to each sentence pair. To avoid the computationally expensive feature calculation for each (mostly non promising) sentence pairs, we apply the following filtering step that removes unpromising pairs early:

- A sentence is no more than twice the length of its translation,
  i.e., Length ratio between the source sentence and target sentence is less than 2.
- At least 50% of the words have a translation on the other sentence.
  i.e., The ratio of words for which a translation is not found on the other side is less than 0.5, for both source and target sentences.

The classifier is only applied to sentence pairs that pass through the above filtering. A similar filtering step is also used in [Smith, 2002] and [Munteanu and Marcu, 2005] with improved performance. It also speeds up the extraction process substantially. Table 4.6 gives the details of the extracted data.

We have observed that there is a considerable amount of parallel document pairs present in the Gigaword Corpora. This is due to the fact that most of these documents come from

Corpus	Sentence Pairs	Source Words	Target Words
Arabic-English			
Xinhua	1,219,779	31,072,567	38,762,429
AFP	725,637	19,192,313	23,570,826
Urdu-English			
VOA	205,756	5,218,938	5,495,066
BBC	319,993	8,931,670	8,049,782

Table 4.6: Corpus statistics for the extracted data for Arabic-English and Urdu-English. For each language pair, two news sources are used for the extraction task.

news agencies who disseminate news in several languages. Typically a news article is written in one language and translated it into another language, sometimes verbatim. The sentence detection algorithms do not always identify all the sentence pairs in a parallel document pair as being parallel. Therefore we lose part of the parallel data. This is particularly problematic in low-resource situations.

We can use the percentage of comparable sentences that are identified in a document pair to guide the process of detecting the parallel documents. If this number is high, the likelihood of the document pair being parallel is also high.

We used this simple heuristic to detect parallel documents in the Arabic and English Gigaword collections. Any document pair with more than 50% sentences aligned were considered as parallel. Out of 400K document pairs, 75K were identified as parallel. Among them, we selected 45K documents that have the same number of sentences in both source and target side. Assuming monotone sentence alignment, we paired sentences at each position to produce candidate parallel sentence pairs. This data was used for translation experiments in the following section.

A better approach is to utilize parallel document alignment algorithms to align sentences in the identified document pairs. There are publicly available tools[2] to perform the alignment task. Dynamic programming based document alignment algorithms assume that the order of the sentences in the two documents are monotone. This can be achieved by selecting document pairs that have fewer crossed alignments.

4.3.4 Translation Experiments

Our primary interest is to evaluate the quality of the extracted sentences in improving the performance of a machine translation system. We do so by selecting the parallel sentences from the extracted comparable sentence pairs and using them as additional training data for training the translation model. Details of the translation experiments are described below; first for Arabic-English and then for Urdu-English.

Arabic-English

For Arabic-English data, the sentence selection step produced 1.9 million comparable sentence pairs. After selecting parallel sentences the extracted corpus had about 1 million sentence pairs.

N-gram Coverage

One of the ways the additional training data can help in improving translation performance is by providing translations for words that are not already covered by the primary corpus. It can also provide additional long n-grams that match with the test data. This helps to

[2] http://champollion.sourceforge.net/

improve translation quality by avoiding erroneous re-orderings produced by the decoder when using a collection of shorter phrases.

We compared the n-gram coverage of the MT05 Arabic test set for 120M Arabic-English parallel corpus (*Parallel*), extracted parallel corpus (*Extracted*) and both corpora combined (*Combined*). Table 4.7 gives the coverage statistics up to 7-grams.

N	**# N-grams**	**Parallel**	**Extracted**	**Combined**
1	28,293	28,144 (99.5)	27,956 (98.8)	28,194 (99.7)
2	27,237	23,333 (85.7)	21,776 (80.0)	24,111 (88.5)
3	26,181	13,839 (52.9)	11,996 (45.8)	15,604 (59.6)
4	25,125	6,351 (25.3)	5,606 (22.3)	8,160 (32.5)
5	24,069	2,794 (11.6)	2,550 (10.6)	4,068 (16.9)
6	23,015	1,354 (5.9)	1,322 (5.7)	2,229 (9.7)
7	21,962	719 (3.3)	759 (3.5)	1,333 (6.1)

Table 4.7: N-gram coverage statistics of Arabic MT05 test set for three corpora: the Arabic-English 120M parallel corpus, the extracted parallel corpus and both corpora combined. The percentage of matching n-grams is given within parenthesis.

As expected, the large *Parallel* corpus has better n-gram coverage than the smaller *Extracted* corpus. However, the combined corpus has a considerably better coverage than the two individual corpora, especially for higher order n-grams. This shows that the additional data has the potential to improve translation quality, if we can reliably identify translations for those n-grams.

Translation Results

The most straightforward way to evaluate the effect of additional data is to build two translation systems with and without it, and compare the performance of the two systems. We built a baseline translation system using the large parallel corpus. Then we built a second translation system by combining both the parallel corpus and the extracted corpus. For comparison, we also built a translation system that only used the extracted corpus.

We also wanted to observe the effect of combining the automatically extracted sentences with different sizes of parallel corpora to build translation systems. For these experiments, we used Arabic-English parallel corpora with 120M, 40M, 13M and 4.5M words. The smaller corpora are roughly 1/3, 1/9, and 1/27 of the larger corpus. These are the same corpora that were used to generate initial lexica in section 4.1. For each parallel corpus, we generated a baseline translation system and a second system by adding the *Extracted* corpus to the baseline corpus.

As we had used the lexicon generated with the full corpus to extract the parallel sentences, we have clearly used more initial parallel data than what is used in the smaller systems. These experiments will show an upper bound for these systems. I.e., how much improvement in the performance can we expect if we are to obtain high quality parallel data from comparable corpora?

We generated IBM word alignment models by running GIZA++ [Och and Ney, 2003] with the parallel text. These alignments were then used to extract the phrases using the Moses toolkit [Koehn et al., 2007]'s grow-diag-final heuristic. A 5-gram suffix array language model [Zhang and Vogel, 2006] was used for all the experiments. To train the language model we used the English side of the full parallel corpus as well as part of the English Gigaword corpus. The phrase table and the language model were then used in a standard phrase-based SMT decoder [Vogel, 2003] to translate the test sets. The decoder parameters are optimized with the minimum error-rate training [Och, 2003] on the development set. All the systems used MT05 as the development set and MT06 as the unseen test set (MT06).

The Arabic-English data extracted using the Munteanu and Marcu [2005] has been released as an LDC corpus *ISI Arabic-English Automatically Extracted Parallel Corpus*[3]. This data has been extracted from the same data sources we have used in our experiments (i.e. Arabic Gigaword and English Gigaword), and hence can be used to compare with our experiments. The corpus statistics for the ISI corpus are given in Table 4.8.

	Sentences	**Arabic Words**	**English Words**
ISI Extracted Corpus	1,124,609	30,639,122	35,292,131

Table 4.8: Corpus statistics of the ISI Arabic-English extracted corpus.

With the ISI extracted data, we built a similar set of translation systems by combining it with different sizes of initial parallel data. We also built a translation system that uses only the ISI extracted data.

Table 4.9 compares the translation results for different systems that uses only extracted data. Here, the baseline system is built using the largest parallel corpus (120M). The translation results are given in case-insensitive BLEU% [Papineni et al., 2002], METEOR[4] [Banerjee and Lavie, 2005] and TER[5] [Snover et al., 2006]. Table 4.10 compares the results for systems with different sizes of initial parallel data.

The combined corpus *Baseline+Extracted* shows no improvement over the baseline for the development set. We see a drop in the performance for the unseen test set. The system trained only on the *Extracted* corpus (line 3) gives the lowest scores. This is to be expected when considering its relatively small size and the fact that it was automatically extracted from comparable sources. However, for the development set, the score is very close to the

[3] LDC catalog number LDC2007E07

[4] Version 1.3

[5] Version 0.7.25

System	**MT05 (Dev)**			**MT06 (Test)**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline	53.37	40.43	38.99	40.73	33.38	48.09
Extracted Only	48.85	38.30	41.58	33.20	29.67	52.78
Baseline+Extracted	52.90	40.09	38.79	39.24	33.05	49.22
ISI Extrated Only	49.46	38.85	41.29	33.37	30.31	52.99
Baseline+ISI Extracted	53.35	40.61	38.92	40.12	33.17	48.78

Table 4.9: Translation results for Arabic-English extracted corpus and the ISI extracted corpus. The baseline system used here is the full parallel corpus of 120M words.

baseline. This shows that the extracted parallel sentences are of high quality. A similar pattern can be observed with the ISI extracted corpus. The *Baseline+ISI Extracted* (line 5) did not outperform the larger baseline system. However, it performs better than the *Baseline+Extracted* system.

When adding the extracted corpus, we see significant improvement in performance for both the 4.5M corpus and the 13M corpus. When the parallel corpus and the automatically extracted corpus are of the similar size (i.e., 30M), we see performance starting to degrade.

When compared with the ISI extracted corpus, our extracted corpus performs significantly better for smaller parallel corpora. With larger parallel corpora, the ISI corpus performs better than our system. This is because the sentence extraction method used in the ISI corpus extraction process is targeted towards extracting parallel sentences. That classifier is trained using parallel sentences as positive examples, and synthetically generated negative examples, giving a much cleaner training setup than the compared training examples we use in our extraction process. The feature set that we use in the classifier is also targeted towards extracting comparable sentence pairs that may not be strictly parallel. These additional sentences help improve the translation performance significantly for smaller

System	**Corpus**	**MT05 (Dev)**			**MT06 (Test)**		
		BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline	**4.5M**	46.18	37.59	44.18	33.99	30.26	52.68
	13M	49.22	38.53	41.97	35.64	31.26	51.12
	40M	51.97	39.84	39.98	38.21	32.31	49.84
	120M	53.37	40.43	38.99	40.73	33.38	48.09
Baseline+Extracted	**4.5M**	49.81	38.74	40.89	35.53	31.04	51.37
	13M	50.68	39.07	40.21	37.07	31.94	50.24
	40M	51.96	39.61	39.59	38.00	32.46	50.03
	120M	52.90	40.09	38.79	39.24	33.05	49.22
Baseline+ISI Extracted	**4.5M**	47.90	38.39	42.44	35.01	31.07	51.91
	13M	50.64	39.35	40.47	37.30	32.07	50.37
	40M	51.85	39.75	39.43	38.06	32.29	49.47
	120M	53.35	40.61	38.92	40.12	33.17	48.78

Table 4.10: Translation results for Arabic-English extracted data when combined with different sizes of initial parallel data. Each column indicates the size of the initial parallel corpus used as the baseline system.

sized corpora, but degrades the performance when added to larger sized corpora. Our focus in this thesis is on low resource languages, which better match the resource situation with the smaller sized corpora in these experiments. Hence, we believe that this approach may have a significant impact on the low resource languages.

Next, we evaluate our extraction process on Urdu-English, a language pair that with low resources.

Urdu-English

For Urdu-English experiments, we used the same setup that was used for the Arabic-English experiments. The alignment model training, phrase extraction, the decoder and the language model remained the same. We trained a baseline translation system using the existing parallel data of around 1 million words.

The amount of extracted Urdu-English comparable sentence pairs was about 500 thousand. We selected two subsets of the extracted data based on the ME classifier probability: top 5% and top 25% of the sentence pairs. Three translation systems were built by combining each set of the extracted sentences with the baseline parallel corpus. The amount of extracted data used in each of the translation systems is given in Table 4.11.

Corpus	**Sentences**	**Urdu**		**English**	
		Words	**Voc**	**Words**	**Voc**
Baseline	54,820	1,082,982	35,715	1,048,583	31,595
+Extracted Top 5%	74,574	1,703,395	41,751	1,597,470	36,194
+Extracted Top 25%	116,751	3,011,689	50,268	2,907,796	42,234
+Extracted All	260,576	6,301,920	58,500	6,543,649	49,760

Table 4.11: Corpus statistics Urdu-English data used in each of the translation systems.

We used the newswire (NW) part of the Urdu-English MT08 and MT09 sets as the development set and the test set, respectively. The translation results are given in Table 4.12.

System	**MT08-NW (Dev)**				**MT09-NW (Test)**			
	BLEU	**MET**	**TER**	**OOV%**	**BLEU**	**MET**	**TER**	**OOV%**
Baseline	21.89	28.61	64.28	2.60	24.13	29.41	61.20	3.40
+Extracted Top 5%	24.48	30.39	62.83	2.14	27.43	31.00	59.38	2.24
+Extracted Top 25%	23.93	29.63	63.17	1.90	26.33	30.71	60.10	1.93
+Extracted All	22.89	28.96	63.68	1.75	25.24	30.01	61.86	1.65

Table 4.12: Translation results for Urdu-English. Different sizes of extracted data are combined with the baseline corpus to build translation systems. Out-of-vocabulary(OOV) rate is also given for each test set.

The translation results indicated that the extracted data improves the performance significantly. We achieve the best performance improvement when we use only top 5% of the extracted sentences. Even though the OOV rate drops as the corpus size increases, adding all the extracted sentences gives the smallest improvement. This indicates that the extracted sentences are not all completely parallel. They contain additional n-grams that are present only in one of the sentences. Although many of the sentences are aligned fairly accurately, the presence of some spurious alignments results in erroneous phrase translations been generated. Detection of these non-parallel n-grams and extracting parallel phrases from these sentence pairs will help to improve the performance. We discuss this in the next chapter.

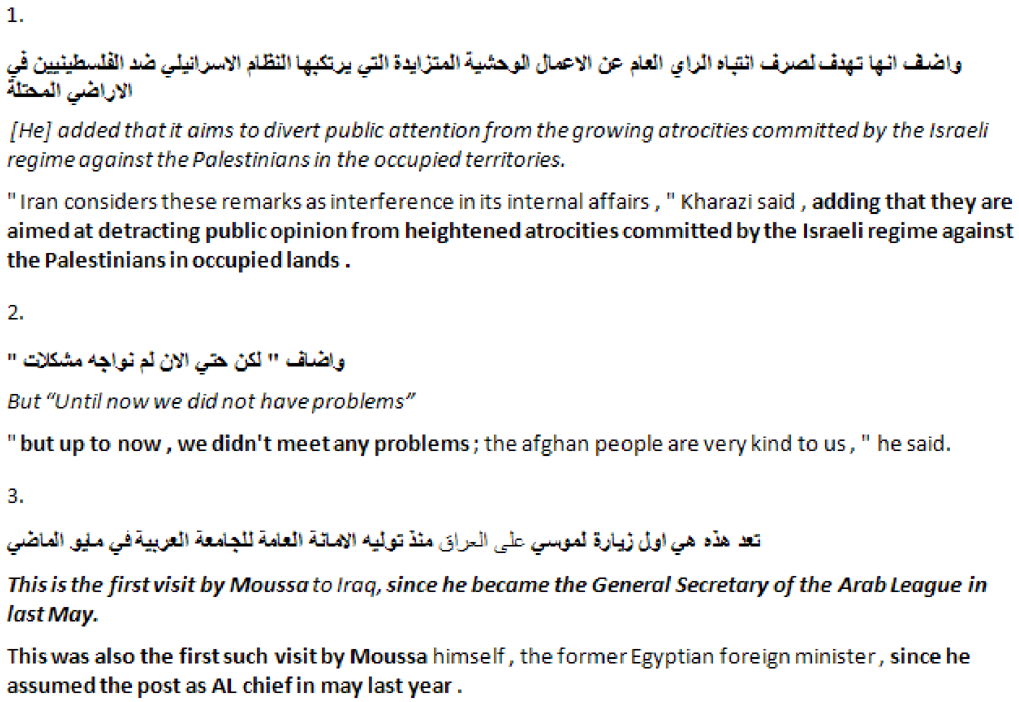
Chapter 5

Extracting Parallel Phrases from Comparable Sentences

In the previous section we identified comparable sentence pairs from two monolingual document collections. The resulting sentence pairs were further analyzed to identify parallel sentences. Even when we cannot find fully parallel sentences in a document pair there could still be sentences that have significant parts which are mutual translations. In this section we propose a new phrase extraction algorithm to identify parallel phrases in comparable sentence pairs. Part of the work described here has been presented earlier in [Hewavitharana and Vogel, 2011] and [Hewavitharana and Vogel, 2012].

Figure 5.1 shows three sample sentences that were extracted from Gigaword Arabic and Gigaword English collections.

For each comparable sentence pair, the Arabic sentence is given first, followed by its literal English translation (in italics). The English sentence is given next. The parallel sections in



1.

واضف انها تهدف لصرف انتباه الراي العام عن الاعمال الوحشية المتزايدة التي يرتكبها النظام الاسرائيلي ضد الفلسطينيين في الاراضي المحتلة

[He] added that it aims to divert public attention from the growing atrocities committed by the Israeli regime against the Palestinians in the occupied territories.

" Iran considers these remarks as interference in its internal affairs , " Kharazi said , **adding that they are aimed at detracting public opinion from heightened atrocities committed by the Israeli regime against the Palestinians in occupied lands .**

2.

واضاف " لكن حتي الان لم نواجه مشكلات "

But "Until now we did not have problems"

" **but up to now , we didn't meet any problems** ; the afghan people are very kind to us , " he said.

3.

تعد هذه هي اول زيارة لموسي على العراق **منذ توليه الامانة العامة للجامعة العربية في مايو الماضي**

This is the first visit by Moussa *to Iraq,* ***since he became the General Secretary of the Arab League in last May.***

This was also the first such visit by Moussa himself , the former Egyptian foreign minister , **since he assumed the post as AL chief in may last year .**

Figure 5.1: Sample comparable sentences that include parallel phrases

each sentence are marked with boldface. In the first two sentence pairs, the English sentence contains the full translation of the Arabic sentence. But there are additional phrases on the English side that are not present on the Arabic sentence. They appear at the beginning of sentence 1 and at the end of sentence 2. In sentence 3, there are parallel phrases as well as phrases that appear only on one side. The phrase "to Iraq" appears only in the Arabic sentence while the phrase "the former Egyptian foreign minister" appears only in the English side.

Standard word and phrase alignment algorithms are formulated to work on parallel sentence pairs. Therefore these standard algorithms are not well suited to operate on partially parallel sentence pairs. Presence of non-parallel phrases results in undesirable alignments.

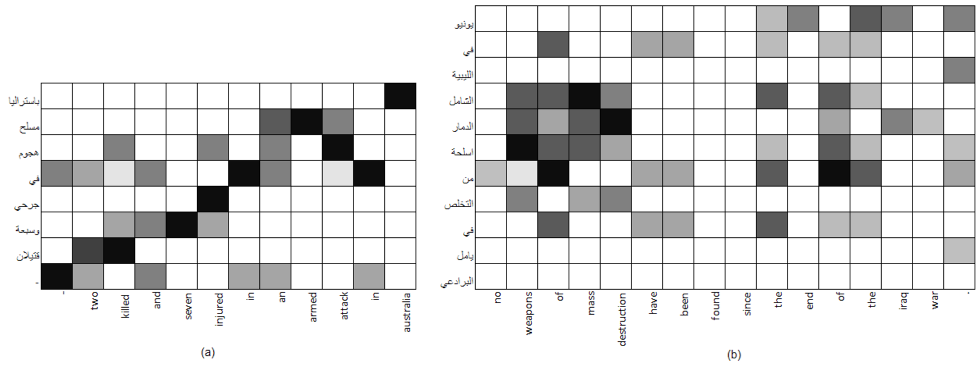


Figure 5.2: Word-to-word alignment pattern for (a) a parallel sentence pair (b) a non-parallel sentence pair

Fig. 5.2 illustrates this phenomenon. It compares a typical word alignment pattern in a parallel sentence pair (a) to one in a non-parallel sentence pair (b). The darkness of a square indicates the strength of the word alignment probability between the corresponding word pair. In 2(a), we observe high probability word-to-word alignments (dark squares) over the entire length of the sentences. In 2(b), we see one dark area above "weapons of mass destruction", corresponding to the parallel phrase pair, and some scattered dark spots, where high frequency English words pair with high frequency Arabic words. These spurious alignments pose problems to the phrase alignment, and indicate that word alignment probabilities alone might not be sufficient.

Our aim is to identify such translation equivalences from comparable sentence pairs. If parallel phrases can be identified, they can be useful as additional parallel data, especially when parallel data is scarce. In this section we propose a new phrase extraction algorithm to identify parallel phrases in comparable sentence pairs and compare it against other alternative approaches.

5.1 PESA Alignment

A phrase alignment algorithm called "PESA" that does not rely on the Viterbi path is described in [Vogel, 2005]. Unlike the standard phrase extraction methods, PESA does not attempt to generate phrase alignments for the full source and target sentence pair. Rather, it identifies the best target phrase that matches a given source phrase. PESA identifies the boundaries of the target phrase by aligning words inside the source phrase with words inside the target phrase, and similarly for the words outside the boundaries of the phrase pair. PESA requires a statistical word-to-word lexicon. A seed parallel corpus is required to automatically build this lexicon.

This algorithm seems particularly well suited in extracting phrase pairs from comparable sentence pairs, as it is designed to not generate a complete word alignment for the entire sentences, but to find only the target side for a phrase embedded in the sentence. We briefly explain the PESA alignment approach below.

Instead of searching for all possible phrase alignments in a parallel sentence pair, this approach finds the alignment for a single source phrase $\tilde{f} = f_1 \dots f_l$. Assume that we have a parallel sentence pair (f_1^J, e_1^I) which contains the source phrase $\tilde{f}$ in the source sentence f_1^J (i.e., $\tilde{f}$ is a prefix of f_1^J). Now we want to find the target phrase $\tilde{e} = e_1 \dots e_k$ in the target sentence e_1^I which is the translation of the source phrase (i.e., $\tilde{e}$ is a prefix of e_1^I).

A constrained IBM1 alignment model is now applied as follows:

- Source words inside phrase boundary are aligned only to the target words inside the phrase boundary. Source words outside the phrase boundary are only aligned to target words outside the phrase boundary.

- Position alignment probability for the sentence, which is $1/I$ in IBM1 model, is modified to be $1/k$ inside the source phrase and to $1/(I-k)$ outside the phrase.

Figure 5.3 shows the different regions. Given the source sentence and the source phrase from position j_1 to j_2, we want to find the boundaries of the target phrase, i_1 and i_2. The dark area in the middle is the phrase we want to align. The size of the blobs in each box indicates the lexical strength of the word pair.

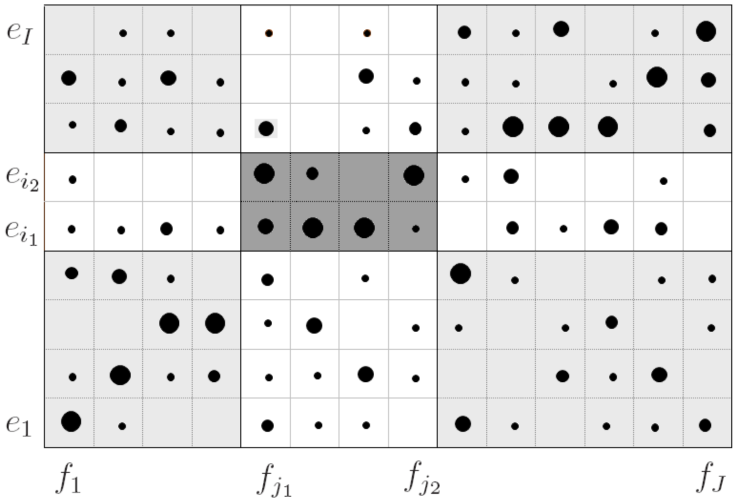


Figure 5.3: Phrase alignment for parallel sentences

The constrained alignment probability is calculated as follows:

$$
\begin{aligned}
P(f|e) &= \left(\prod_{j=1}^{j_1-1} \sum_{i \notin (i_1 \ldots i_2)} \frac{1}{I-k} P(f_j|e_i) \right) \\
&\times \left(\prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} \frac{1}{k} P(f_j|e_i) \right) \\
&\times \left(\prod_{j=j_2+1}^{J} \sum_{i \notin (i_1 \ldots i_2)} \frac{1}{I-k} P(f_j|e_i) \right) \qquad (5.1)
\end{aligned}
$$

$P(e|f)$ is similarly calculated by switching source and target sides in equation 5.1:

$$\begin{aligned} P(e|f) \quad &= \quad \left(\prod_{i=1}^{i_1-1} \sum_{i \notin (j_1 ... j_2)} \frac{1}{J-l} P(e_i|f_j) \right) \\ &\times \quad \left(\prod_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \frac{1}{l} P(e_i|f_j) \right) \\ &\times \quad \left(\prod_{i=i_2+1}^{I} \sum_{j \notin (j_1 ... j_2)} \frac{1}{J-l} P(e_i|f_j) \right) \end{aligned} \tag{5.2}$$

To find the optimal target phrase boundaries, the two probabilities in equations 5.1 and 5.2 are interpolated log linearly and takes the pair (i_1, i_2) that gives the highest probability.

$$(i_1, i_2) = \underset{i_1, i_2}{\operatorname{argmax}} \ (1-\lambda) \ log(P(f|e)) + \lambda \ log(P(e|f)) \tag{5.3}$$

The value of λ is estimated using held-out data.

PESA can be used to identify all possible phrase pairs in a given parallel sentence pair by iterating over every source phrase. An important difference is that each phrase is found independently of any other phrase pair, whereas in the standard phrase extraction they are tied through the word alignment of the sentence pair.

In the following section we describe the proposed approach based on PESA for aligning comparable sentences.

5.2 Proposed Approach

Now we want to extract parallel phrases from comparable sentence pairs using the constrained alignment. The sentence pair (f_1^J, e_1^I) now contains parallel parts as well as non-parallel parts.

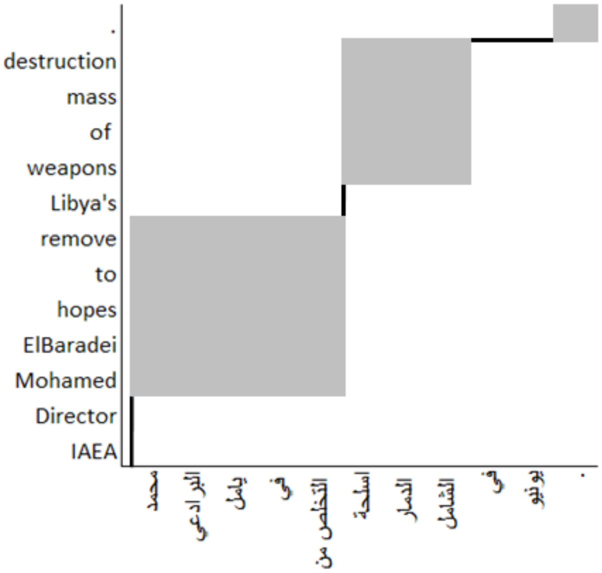


Figure 5.4: Phrase alignment for comparable sentences

Figure 5.4 gives an example of such a sentence. The gray squares indicated parallel phrases that are common to both sentences. Black lines indicate non-parallel phrases that are present only on one side.

There are several ways we can adopt the non-Viterbi phrase extraction to comparable sentences.

- Apply the same approach assuming the sentence pair is parallel. The inside of the source phrase is aligned to the inside of the target phrase, and the outside, which can be non-parallel, is aligned the same way.

- Disregard the words that are outside the phrase we are interested in. Find the best target phrase by aligning only the inside of the phrase. This will considerably speed-up the alignment process.

- Do not assume the outside of the phrase to be parallel. Explicitly model the non-parallel sections of the phrase and integrate with PESA.

The first two options only require minor modifications to the original model. Here we propose a model for the third option.

Following [Quirk et al., 2007], we hypothesize a monolingual model, such as an n-gram language model, is better suited to model the non-parallel parts. Conversely, a translation model will better model the parallel phrases than the language model. I.e., if the phrase pair $(\tilde{f}, \tilde{e})$ is parallel then $P(\tilde{f}, \tilde{e}) > P(\tilde{f}) \cdot P(\tilde{e})$

We have two monolingual models; L_S for the source language and L_T for the target language. Also, two translation models, X_{ST} and X_{TS}.

Using Bayes' rule we can break down the join probability into two decompositions of translation model and language model.

$$Pr(\tilde{f}, \tilde{e}) = Pr(\tilde{e}) \cdot Pr(\tilde{f}|\tilde{e}) = Pr(\tilde{f}) \cdot Pr(\tilde{e}|\tilde{f})$$

However, our models trained on limited amounts of data are only approximations. Therefore, the equation will not strictly be true and the expressions $Pr(\tilde{e}) \cdot Pr(\tilde{f}|\tilde{e})$ and $Pr(\tilde{f}) \cdot Pr(\tilde{e}|\tilde{f})$ may differ. We take the average of the two estimates.

Thus, the probability of aligning the phrase pair $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$ is as follows:

$$Pr(f_{j_1}^{j_2}, e_{i_1}^{i_2}) = \frac{1}{2}\left[L_S(f_{j_1}^{j_2}) \cdot X_{TS}(e_{i_1}^{i_2}|f_{j_1}^{j_2})\right] + \frac{1}{2}\left[L_T(e_{i_1}^{i_2}) \cdot X_{ST}(f_{j_1}^{j_2}|e_{i_1}^{i_2})\right] \tag{5.4}$$

where,

$$L_S(f_{j_1}^{j_2}) = \prod_{k=j_1}^{j_2} L_S(f_k|f_1^{k-1}) \tag{5.5}$$

and

$$X_{TS}(e_{i_1}^{i_2}|f_{j_1}^{j_2}) = \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} \frac{1}{(i_2 - i_1)} X_{TS}(f_j|e_i) \tag{5.6}$$

$L_T(e_{i_1}^{i_2})$ and $X_{ST}(f_{j_1}^{j_2}|e_{i_1}^{i_2})$ are defined similarly.

To find the best parallel phrase pair in a comparable sentence pair, we assume that the rest of the sentence outside the phrase boundaries, is non-parallel. Hence, this outside area can be modeled using language models. This assumption is not always true, as there can be other parallel phrases in the sentence pair. At this point, however, we are only interested in finding the best parallel phrase pair in a given comparable sentence pair. We will address this issue later.

We compute the joint probability of the sentence pairs $P(f, e)$ as follows:

$$\begin{aligned} P(f,e) &= L_S(f_1^{j_1-1}) \cdot L_T(e_1^{i_1-1}) \\ &\times Pr(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \\ &\times L_s(f_{j_2+1}^{J}) \cdot L_T(e_{i_2+1}^{I}) \end{aligned} \tag{5.7}$$

and find the boundaries (j_1, j_2, i_1, i_2) of the best pair by optimizing $P(f, e)$:

$$(i, j, k, l) = \underset{j_1, j_2, i_1, i_2}{\operatorname{argmax}} \ P(f, e) \tag{5.8}$$

Phrase Extraction

To extract parallel phrases from the comparable sentence pairs we adopt the following procedure: First assume that there is only one parallel phrase embedded in the comparable sentence pair, and find the best parallel phrase. This process divides the sentence pair into several regions as in figure 5.3. Then perform a greedy search in the non-blocked regions to identify remaining parallel phrases. The following sections explain the proposed algorithm.

It is important to point out that the phrase extraction algorithm does not restrict the word alignment within the phrases to be monotone. There can be non-monotone word alignment, as well as unaligned words within a phrase pair.

Find the Best Parallel Phrase Pair

The straightforward way to find the best phrase pair is by computing the alignment for every value of the boundary pairs (j_1, j_2) and (i_1, i_2) within the boundaries of the sentences and selecting the boundaries with the highest probability. This method is computationally expensive and has runtime complexity $O(J^2 I^2)$. However, most of the boundary pairs in this calculation will not produce any meaningful phase pairs. The boundary pair (1,1) and (1,15), for example, is unlikely to be a reasonable phrase alignment as it aligns a single source word with 15 target words. Therefore, the search space can be restricted to those values for the boundaries that have the potential to produce meaningful phrases.

We introduce minimum and maximum phrase lengths to both source phrase (l_{min}, l_{max}) and target phrase (k_{min}, k_{max}). The best parallel phrase is then identified using algorithm 2.

This process can also be thought of as follows: From among a set of allowed phrase patterns of different sizes (e.g. 3×8, 4×6, etc), we want to find the best placement for that pattern in the sentence that would produce the best parallel phrase pair.

The length restriction reduces the run-time complexity of the algorithm to $O(c.JI)$ where c is the number of different source and target phrase length combinations allowed. By

Algorithm 2 Algorithm to identify the best phrase translation pair

```
Input: Source sentence $f_1^J$, target sentence $e_1^I$, phrase length limits $(l_{min}, l_{max}, k_{min}, k_{max})$
Output: The boundaries of the best phrase translation pair $(j_1, j_2, i_1, i_2)_{best}$
$BestProb = 0$
$(j_1, j_2, i_1, i_2)_{best} = (0, 0, 0, 0)$
for $l = l_{min}$ to $l_{max}$ do
    for $j = 1$ to $(J - l + 1)$ do
        for $k = k_{min}$ to $k_{max}$ do
            for $i = 1$ to $(I - k + 1)$ do
                if $NewProb = Pr_{(j,(j+l-1),i,(i+k-1))}(f, e) > BestProb$ then
                    $BestProb = NewProb$
                    $(j_1, j_2, i_1, i_2)_{best} = (j, (j + 1 - 1), i, (i + k - 1))$
return $(j_1, j_2, i_1, i_2)_{best}$
```

restricting (l_{min}, l_{max}) and (k_{min}, k_{max}), we can keep c in a manageable level. For example, if we are interested in phrases of 3 to 7 words long in both source and target sides, c is 25.

Similar phrase length restrictions are enforced when extracting phrase translation pairs from parallel sentences using various heuristics to combine word-alignments.

The relationship between source phrase and target phrase is not uniform. Morphologically rich languages, such as Arabic, typically have fewer words in a sentence than English. In parallel sentences, the length ratio between Arabic and English is between 1 and 1.5. We can use this ratio to further restrict the search with the following conditions:

$$\frac{1}{c_1} \cdot l - c_2 \leq k \leq c_1 \cdot l + c_2 \tag{5.9}$$

where c_1 is the length ratio and c_2 is an additive constant c_1 and c_2 were empirically selected using a held out data set.

Lexical models used in the alignment process may contained erroneous alignments that will affect the phrase extraction process. The models could give highest score to an alignment that is not the best, or the correct one. Therefore we select not only the best phrase pair, but all candidate phrase pairs whose scores lie within a margin of the best one.

Find Other Parallel Phrases in the Sentence

There are four possible regions (regions 1,3,5, and 7 in figure 5.3, counting clockwise starting from bottom left region) in which the remaining parallel phrases can be present. To detect them, a greedy search can be performed as follows:

1. Define a set $\mathcal{A}$ which contains tuples (i, j) to indicate the word positions that are already covered by previously identified phrase pairs. This is initialized with values from the best parallel phrase pair identified as in the previous section.
2. For each of the possible phrase pair patterns, iterate over all possible placements of them in the sentence pair. They are considered as candidates only if they can be placed completed on an unaligned area. I.e. The word alignment pairs corresponding to the phrase pair are not included in $\mathcal{A}$.
3. Out of all possible such phrase pair candidates, pick the pair that gives the best overall sentence alignment.
4. The phrase pair is extracted as parallel only if the alignment probability is better than the previous sentence alignment probability. i.e. The phrase pair is better explained using the translation model than the language model. $\mathcal{A}$ is updated accordingly.
5. This process is iterated until no further parallel phrases can be identified.

The search algorithm is as shown in Algorithm 3.

Algorithm 3 Search for other parallel phrases

```
Input: Source sentence $f_1^J$, target sentence $e_1^I$, list of phrase patterns $(m \times n)$,
    best phrase pair boundaries $(j_1, j_2, i_1, i_2)_{best}$ found using Algorithm 2
Output: List of phrase pairs $(f_j^{j+m}, e_i^{i+n})$
$OldProb = Pr_{(j_1,j_2,i_1,i_2)}(f, e)$
$\mathcal{A} = \{(j, i) | j_1 \leqslant j \leqslant j_2; i_1 \leqslant i \leqslant i_2\}$
while $Continue$ do
  $BestProb = 0$
  for each phrase pattern $(m \times n)$ do
    for $j = 1$ to $(J - m + 1)$ do
      for $i = 1$ to $(I - n + 1)$ do
        if $(x, y) \notin \mathcal{A} \; \forall \{(x, y) | j \leqslant x \leqslant j + m; i \leqslant y \leqslant i + n\}$ then
          compute $Pr(f_j^{j+m}, e_i^{i+n})$ using equation (5.4)
          compute $L_S(f_j^{j+m})$ using equation (5.5)
          similarly compute $L_T(e_i^{i+n})$
          $NewProb = \frac{OldProb \cdot Pr(f_j^{j+m}, e_i^{i+n})}{L_S(f_j^{j+m}) \cdot L_T(e_i^{i+n})}$
          if $NewProb > BestProb$ then
            $BestProb = NewProb$
  if $BestProb > OldProb$ then
    $OldProb = BestProb$
    $\mathcal{A} = \mathcal{A} \cup \{(j, i) | j_{best} \leqslant j \leqslant j_{best} + m; i_{best} \leqslant i \leqslant i_{best} + n\}$
    Output the phrase pair $(f_{j_{best}}^{j_{best}+m}, e_{i_{best}}^{i_{best}+n})$
    $Continue = True$
  else
    $Continue = False$
```

Several computational steps in the above algorithm can be reduced by incremental calculation of the partials sums of probabilities.

Additional Features

In our initial experiment we found that the phrases extracted using the equation 5.7 tend to have incorrect word boundaries. Most of the time, the extracted phrase contained more words on either side of the boundary. This is due to the effect of two language model scores for the left and right of the parallel phrase, where the model prefers a smaller language model score. It was necessary to introduce other features to better balance the phrase length. We defined the following features:

- Phrase Length Ratio (1): This is the ratio between the source phrase length and the target phrase length. i.e. $|j_2 - j_1|/|i_2 - i_1|$.
- Phrase Length Difference (1): Difference in source phrase length and the target phrase length. i.e. $|j_2 - j_1| - |i_2 - i_1|$.
- Target Language Model Score for outside region (2): These score are normalized with respect to the source or target phrase length.
- Phrase alignment score for the inside region (1): This is computed as in equation 5.4.

The phrase alignment score in equation 5.7 is combined with the features defined features in a weighted log-linear model as in equation 5.10:

$$(i_1, i_2)_{best} = \underset{(i_1,i_2)}{\operatorname{argmax}} \sum_{k=1}^{6} \lambda_k F_k(f_1^J, e_1^I, j_1, j_2) \tag{5.10}$$

where, F_ks are feature functions as defined above and λ_ss are the weights associated with feature function which are optimized.

Instead of searching for the best parallel phrase pair in a given comparable sentence pair as in algorithm 2, here we used a modified version, where the target phrase boundaries are obtained for a given source phrase. The search algorithm is as is shown in Algorithm 4

Algorithm 4 Algorithm to identify the best target boundaries for a given source phrase

```
Input: Source sentence $f_1^J$, target sentence $e_1^I$, source phrase boundary $(j_1, j_2)$,
    target phrase length limits $(k_{min}, k_{max})$
Output: The boundaries of the best target phrase translation pair $(i_1, i_2)_{best}$
$BestProb = 0$
$(i_1, i_2)_{best} = (0, 0, 0, 0)$
for each source phrase $\tilde{e}$ do
  Select comparable sentence pairs $(f_1^J, e_1^I)$ such that $e$ contains $\tilde{e}$
  for each $(f_1^J, e_1^I)$ do
    for $k = k_{min}$ to $k_{max}$ do
      for $i = 1$ to $(I - k + 1)$ do
        Compute $NewProb = Pr(f, e)$ using the log-linear model
        if $NewProb > BestProb$ then
          $BestProb = NewProb$
          $(i_1, i_2)_{best} = (i, (i + k - 1))$
    return $(i_1, i_2)_{best}$
```

When extracting phrases for a test set, we first generate all possible source phrases (with phrase length ranging between l_{min} and l_{max}), and extract the desired target phrases using algorithm 4 above. By restricting the search for target boundaries for only a given source phrase, we reduce the run-time complexity of the algorithm to $O(c.I)$ where c is the number of different source and target phrase length combinations allowed. This is faster than aligning all the words in the sentence pairs as in the Viterbi alignment.

Parameter Optimization

To optimize the parameters of the extended phrase extraction model we used a manually word-aligned parallel corpus. From the corpus we extracted parallel phrase pairs that are consistent with the underlying word alignments. Each source phrase was allowed to have multiple reference target phrases. The details of the extraction process are explained in section 5.4.2.

We used the online margin infused relaxed algorithm (MIRA) [Crammer et al., 2006] to obtain the optimized model weights. MIRA is defined by an update rule, which is subject to max-margin constraints with respect to a loss function that is computed for the predicted hypothesis against the reference phrase. The weights are updated after seeing each training phrase pair in each iteration (hence *online*). Change in the weight vector is constrained so that the margin between the hypothesis and the reference should be at least as large as the loss value. Formally, the update rule in MIRA is as follows:

$$\underset{w}{\text{argmax}} \;\; ||w^{i+1} - w|| + C \cdot \sum_{k} \xi_k(y_t, y') \tag{5.11}$$

$$s.t., \quad s(x_t, y_t) - s(x_t, y_k) + \xi_k(y_t, y') \geq L(y_t, y'); \;\; y' \in best_k(x_t, w^i); \;\; \xi_k(y_t, y') \geq 0$$

where w is the weight vector, y_t is the target candidate closest to the reference phrase, y' is in the k-best candidate list, $s(x, y)$ is the scoring function, $L(y_t, y')$ is the computed loss. ξ_k is the slack variable whose value is always non-negative and C is the slack constant the determines how aggressively the weight vector is updated after each instance. The weights obtained after each update are averaged at the end of the training step in order to avoid overfitting.

We optimized the model performance towards an oracle target phrase which is selected from a k-best list of candidate target phrase translations. Reference target phrases obtained from the manual alignment may not always be reachable by the phrase extraction algorithm. This is due to the fact that we extract target candidate phrases from sentence pairs, which contain the source phrase in the source sentence, but may not always contain the reference phrase in the target sentence (i.e., an alternative translation). Thus, from the k-best list of candidate target phrase, we selected an m-best list of target phrase that are closest to one or multiple references. In the experiments we set m=1 (i.e., only the first-best oracle phrase was considered).

To compute the loss function we used a modified smoothed version of BLEU ($mBLEU$). The BLEU metric was defined on the document level which considers n-gram precision (typically up to 4). To compute BLEU-like scores at the phrase level we defined a smoothed version that considers only unigram matches in its computation. The smoothing technique is similar to the NIST BLEU version[1]. It is computed by adding a partial count of $(1/2^k)$ for each precision score whose matching n-gram count is zero, where $k = 1$ for the first n values for which the n-gram match count is zero.

The loss function using the smoothed-BLEU is defined as follows:

$$L(y_t, y' = 1 - mBLEU(y_t, y')$$

The error rates in training and testing sets for MIRA is evaluated in terms of normalized Levenshtein distance at word level (i.e., word error rate). This metric awards partial "credit" for candidate target phrases when the reference phrase is not always reachable.

With the extracted phrase pairs from the manually aligned corpus, we used a 75/25% split for training/testing. Each source phrase is allowed to have multiple reference phrase pairs in the setup. We generated a 10-best target phrase candidate list for each source phrase and evaluated the error against the first best candidate. We ran MIRA for 10 iterations.

[1] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13.pl

The final averaged weight vector obtained after the optimization was used for the phrase extraction experiments in Section 5.4.3.

Further details of the optimization is described in [Gupta et al., 2011].

5.3 Other Approaches

We compare the proposed phrase extraction with two other phrase alignment approaches methods.

5.3.1 Standard Viterbi Alignment

Here we use the typical phrase extraction approach used by Statistical Machine Translation systems: obtain word alignment models for both directions (source to target and target to source), combine the Viterbi paths using one of many heuristics, and extract phrase pairs from the combined alignment. We used the Moses toolkit [Koehn et al., 2007] for this task. To obtain the word alignments for comparable sentence pairs, we performed a forced alignment using the trained models.

5.3.2 Binary Classifier

We used a Maximum Entropy classifier as our second approach to extract parallel phrase pairs from comparable sentences. This is similar to the classifier used in Chapter 4, but here we apply it at phrase level rather than at sentence level. The classifier probability is defined as:

$$p(c|S,T) = \frac{exp\left(\sum_{i=1}^{n} \lambda_i f_i(c,S,T)\right)}{Z(S,T)}, \tag{5.12}$$

where $S = s_1^L$ is a source phrase of length L and $T = t_1^K$ is a target phrase of length K. $c \in \{0,1\}$ is a binary variable representing the two classes of phrases: *parallel* and *not parallel*. $p(c|S,T) \in [0,1]$ is the probability where a value $p(c=1|S,T)$ close to 1.0 indicates that S and T are translations of each other. $f_i(c,S,T)$ are feature functions that are co-indexed with respect to the class variable c. The parameters λ_i are the weights for the feature functions obtained during training. $Z(S,T)$ is the normalization factor. In the feature vector for phrase pair (S,T), each feature appears twice, once for each class $c \in \{0,1\}$.

We use a subset of the features from the classifier in section 4.2. The features are computed primarily on IBM Model-1 probabilities. We do not explicitly compute IBM Model-1 alignments. To compute coverage features, we identify alignment points for which IBM Model-1 probability is above a threshold. We produce two sets of features based on IBM Model-1 probabilities obtained by training in both directions. All the features have been normalized with respect to the source phrase length L or the target phrase length K. We use the following 11 features:

1. Lexical probability (2): IBM Model-1 log probabilities $p(S|T)$ and $p(T|S)$

2. Phrase length ratio (2): source length ratio K/L and target length ratio L/K

3. Phrase length difference (1): source length minus target length, $L - K$

4. Number of words covered (2): A source word s is said to be covered if there is a target word $t \in T$ such that $p(s|t) > \epsilon$, where $\epsilon = 0.5$. Target word coverage is defined accordingly.

5. Number of words not covered (2): This is computed similarly to 4. above, but this time counting the number of positions that are not covered.

6. Length of the longest covered sequence of words (2)

To train the classifier, we used parallel phrase pairs extracted from a manually word-aligned corpus. In selecting negative examples, we followed the same approach as in [Munteanu and Marcu, 2005]: pairing all source phrases with all target phrases, but filter out the parallel pairs and those that have high length difference or a low lexical overlap, and then randomly select a subset of phrase pairs as the negative training set.

5.4 Experiments

5.4.1 Evaluation Setup

We want to compare the performance of the different phrase alignment methods in identifying parallel phrases embedded in comparable sentence pairs. Using a manually aligned parallel corpus and two monolingual corpora, we obtained a test corpus as follows: From the manually aligned corpus, we obtain parallel phrase pairs (S, T). Given a source language corpus $\mathcal{S}$ and a target language corpus $\mathcal{T}$, for each parallel phrase pair (S, T) we select a sentence s from $\mathcal{S}$ which contains S and a target sentence t from $\mathcal{T}$ which contains T. These sentence pairs are then non-parallel, but contain parallel phrases, and for each sentence pair the correct phrase pair is known. This makes it easy to evaluate different phrase alignment algorithms.

Ideally, we would like to see the correct target phrase T extracted for a source phrase S. However, even if the boundaries of the target phrase do not match exactly, and only a partially correct translation is generated, this could still be useful to improve translation quality. We therefore will evaluate the phrase pair extraction from non-parallel sentence pairs also in terms of partial matches.

To give credit to partial matches, we define precision and recall as follows: Let W and G denote the extracted target phrase and the correct reference phrase, respectively. Let M denote the tokens in W that are also found in the reference G. Then

$$Precision = \frac{|M|}{|W|} * 100 \tag{5.13}$$

$$Recall = \frac{|M|}{|G|} * 100 \tag{5.14}$$

These scores are computed for each extracted phrase pair, and are averaged to produce precision and recall for the complete test set. Finally, precision and recall are combined to generated the F score in the standard way:

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5.15}$$

5.4.2 Alignment Accuracy

We conducted our experiments on Arabic-English. To test the phrase extraction accuracy we used manually word-aligned Arabic-English sentence pairs, which were released under the GALE project. These sentences are from the news-wire domain and consist of 300 pairs. From these sentences we extracted phrase pairs up to 10 words long that are consistent with the underlying word alignment. I.e., a source phrase pair is extracted only if all the words inside the phrase are aligned to the words inside the target phrase, and vice versa. We allowed unaligned words to be present within a phrase. From the resulting list of phrase pairs, we removed the 50 most frequently occurring pairs as well as those only consisting of punctuation. Almost all high frequency phrases are function words, which are typically covered by the translation lexicon. Line 1 in Table 5.1 gives the n-gram type distribution for the source phrases.

	1	2	3	4	5	6	7	8	9	10	All
# N-grams	2,826	3,665	3,447	3,048	2,718	2,414	2,076	1,759	1,527	1,378	24,858
# Found	2,746	2,655	1,168	373	87	29	7	2	1	0	7,068

Table 5.1: N-gram type distribution of manually aligned phrases set

Using the phrase pairs extracted from the manually aligned sentences, we constructed a comparable corpus as follows:

1. For each Arabic phrase, we search the Arabic Gigaword corpus for sentences that contain the phrase and select up to 5 sentences. Similarly, for each corresponding English phrase we select up to 5 sentences from English Gigaword.

2. For each phrase pair, we generate the Cartesian product of the sentences and produce a sentence pair collection. I.e., up to 25 comparable sentence pairs were constructed for each phrase pair.

3. We only select sentences up to 100 words long, resulting in a final comparable corpus consisting of 170K sentence pairs.

Line 2 in Table 5.1 gives the n-gram type distribution for the phrase pairs for which we found both a source sentence and a target sentence in the monolingual corpora. As expected, the longer the phrase the less likely it is to find them in even larger corpora.

We consider the resulting set as our comparable corpus which we will use to evaluate all alignment approaches. In most sentence pairs, except for the phrase pair that we are interested in, the rest of the sentence does not typically match the other side.

We obtained the Viterbi alignment using standard word alignment techniques: IBM4 word alignment for both directions, Viterbi path combination using heuristics ('grow-diag-final') and phrase extraction from two-sided training, as implemented in the Moses package [Koehn et al., 2007]. Because the non-parallel segments will lead the word alignment astray, this may have a negative effect on the alignment in the parallel sections. Alignment models trained on parallel data are used to generate the Viterbi alignment for the comparable sentences. We then extract the target phrases that are aligned to the embedded source phrases. A phrase pair is extracted only when the alignment does not conflict with other word alignments in the sentence pair. The alignments are not constrained to produce contiguous phrases. We allow unaligned words to be present in the phrase pair. This method is similar to the one used to extract phrases from the manually word-aligned corpus. For each source phrase we selected the target phrase that has the least unaligned words.

The classifier is applied at the phrase level. We generate the phrase pair candidates as follows: For a given target sentence we generate all n-grams up to length 10. We pair each n-gram with the source phrase embedded in the corresponding source sentence to generate a phrase pair. From the 170 thousand sentence pairs, we obtained 15.6 million phrase pair candidates. The maximum entropy classifier is then applied to the phrase pairs. For each source phrase, we pick the target candidate for which $p(c = 1, S, T)$ has the highest value.

For the phrase alignment algorithm proposed in section 5.2 we used both inside and outside alignments explained earlier. For each source phrase pair, we select the best scoring

Lexicon	Viterbi				Classifier				Proposed			
	Exact	P	R	F	Exact	P	R	F	Exact	P	R	F
Lex-120M	43.56	65.71	57.99	61.61	54.46	81.79	89.10	85.29	67.94	93.61	86.59	89.96
Lex-40M	42.95	65.68	56.69	60.85	53.57	81.32	88.34	84.69	67.28	92.91	85.96	89.30
Lex-13M	41.10	63.60	51.15	56.70	52.38	80.30	86.64	83.35	65.81	91.95	84.52	87,93
Lex-4.5M	41.02	62.10	49.38	55.01	52.51	80.51	83.84	82.14	63.23	89,08	81.84	85.31
Lex-BTEC	19.10	26.94	23.63	25.18	18.76	45.90	36.17	40.46	17.45	46.36	36.07	40.57

Table 5.2: Alignment accuracy for the three approaches: standard Viterbi, classifier and the proposed approach.

target phrase.

As our focus is on low resource situations, we tested each method with several lexica, trained on different amounts of initial parallel data. These are the same corpora used in sentence selection experiments in Chapter 4: *Lex-120M*, *Lex-40M*, *Lex-13M*, *Lex-4.5M* and *Lex-BTEC*.

Table 5.2 gives the results for all three alignment approaches. Results are presented as percentages of: exact matches found (Exact), precision (P), recall (R) and F. The Viterbi alignment gives the lowest performance. This shows that the standard phrase extraction procedure, which works well for parallel sentences, is ill-suited for partially parallel sentences.

Despite the fact that the classifier incorporates several features including the lexical features, the performance of the proposed algorithm, which uses only the lexical features, has consistently higher precision than the classifier approach. This demonstrates that computing both inside and outside probabilities for the sentence pair helps the phrase extraction. The classifier lacks this ability because the phrase pair is evaluated in isolation, without the context of the sentence.

Except for the BTEC corpus, the performance degradation is smaller as the lexicon size is reduced. This shows that the approaches are robust for smaller amounts of parallel data.

Instead of using token precision, an alternative method of evaluating partial matches, is to give credit based on the length of the overlap between the extracted phrase and the reference. Precision and recall can then be defined based on the longest common contiguous subsequence, similar to [Bourdaillet et al., 2010]. Results we obtained using these methods were similar to the results in Table 5.2.

In the translation experiments below, we compare the performance of the standard Viterbi approach and the proposed approach.

5.4.3 Translation Results

Arabic-English

In Chapter 4 we extracted sentence pairs from the comparable corpus using a ME classifier, and only those pairs with a classification probability that were above an empirically determined threshold were selected for the translation experiments. For the phrase extraction experiments in this section we used the remaining part of the extracted sentences that were below the cut-off threshold.

From the comparable sentences, two sets of phrase pairs were extracted: one using the Viterbi alignment approach and the other using the proposed approach as in algorithm 4. The extraction process followed the same steps explained in Section 5.4.2. The phrases extracted from the Viterbi alignment were sampled to include only those needed to cover the source side of the test sets.

The algorithm 4 extracts target translations for a given set of source phrases. For each test set, we generated all source n-grams up to length 7, and used them to extract phrases using the proposed approach. The model parameters were optimized using the MIRA as described under *Parameter Optimization* using manually aligned data. For each source phrase, we extract up to 10 target translation alternatives. From the comparable sentences we extracted 881,161 phrase pairs, out of which 107,528 were new source phrases. The extracted phrases had an average length of 2.7 words (compared to 2.0 words for the phrases extracted using the Viterbi alignment).

For each approach, the extracted phrases were combined with the phrases extracted form the sentence extraction step in Chapter 4. The combined phrase table was then used to decode the test sets. We used the same setup that was used in Chapter 4 for the translation experiments, including the decoder and the language models.

The extracted phrases were combined with phrases extracted from different sizes of initial parallel data. These were the same systems that were used in sentence extraction in Chapter 4 (i.e., 4.5M, 13M, 40M and 120M). Translation results for Arabic-English are given in Table 5.3. The results are given in BLEU, METEOR and TER metrics. For each test set, results are shown for 3 systems. The first system, *Baseline+Sentences*, corresponds to the initial parallel data plus the extracted parallel sentences from comparable data (see Table 4.10 in Chapter 4). This serves as the *baseline* for the phrase extraction step, which we try to surpass with the extracted phrases. The second system, *+Phrases(Viterbi)*, corresponds to the phrases extracted using the Viterbi alignment when combined with phrases from *Baseline+Sentences*. The third system, *+Phrases(Proposed)*, corresponds to the phrases extracted using the proposed approach when combined with phrases from *Baseline+Sentences*. MT05 was used as the development set to optimize the decoder parameters using MER training. MT06 was used as an unseen test set.

System	**Corpus**	**MT05 (Dev)**			**MT06 (Test)**		
		BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline+Sentences	**4.5M**	49.81	**38.74**	**40.89**	35.53	31.04	51.37
	13M	50.68	39.07	**40.21**	**37.07**	**31.94**	50.24
	40M	**51.96**	**39.61**	39.59	**38.00**	**32.46**	50.03
	120M	**52.90**	**40.09**	38.79	39.24	**33.05**	49.22
+Phrases(Viterbi)	**4.5M**	49.19	37.74	41.21	35.02	30.81	51.73
	13M	50.26	38.26	40.44	36.45	31.41	**50.20**
	40M	51.45	38.92	39.87	37.57	32.19	49.55
	120M	52.87	40.09	38.85	38.13	32.38	49.35
+Phrases(Proposed)	**4.5M**	**50.04**	38.15	41.36	**36.00**	**31.38**	**50.83**
	13M	**51.92**	**39.52**	40.46	36.79	31.85	50.27
	40M	51.74	39.40	**39.55**	37.98	32.16	**49.58**
	120M	52.89	40.05	**38.64**	38.67	32.61	**48.64**

Table 5.3: Translation results for Arabic-English extracted phrases when combined with the extracted sentences plus different sizes of initial parallel data. Each row indicates the size of the initial parallel corpus used as the baseline system.

The results show that the proposed phrase extraction method is consistently better than extracting phrases with the standard Viterbi algorithm. When the new phrases are added, there is an improvement in performance for the 4.5M system. This difference is statistically significant (at 95% level) for MT06. However, for other systems, the extracted phrases fail to improve the performance over the combined Baseline+Sentence system. Although the phrase extraction brings in additional phrases, the improvement in source side coverage compared to the phrases in the baseline system range between 2% and 5% for different systems. As the phrases are extracted from comparable data, the additional phrase pairs may introduce noise to the combined system. For the smallest systems with the lowest lexical coverage, the additional phrases help improve the performance. When the size of the system increases we notice that the additional phrases starts having less effect, and leads to degrade in the results.

Urdu-English

Similar phrase extraction experiments were done for Urdu-English as well. Here, we used the "Baseline+Extracted Top 25%" system (see Table 4.12 from Chapter 4) as the baseline system. This system only uses the top 25% of the extracted sentences. We used the remaining sentences for the phrase extraction task, and extracted two sets of phrases, one each using the Viterbi approach and the proposed approach as in Algorithm 4. Similar to Arabic-English experiments in the previous section, we generated all source n-grams up to length 7 from the two test sets , and extracted up to 10 target translations for each source n-gram. From the comparable sentences we extracted 424,870 phrase pairs, out of which 182,262 were new phrase pairs. These phrases covered 20,832 additional source n-grams, which is a 30% increase in source n-gram coverage compared to the "Baseline+Extracted Top 25%" system. The extracted phrases had an average length of 3.24 words (compared to 2.1 words for the phrases extracted using the Viterbi alignment). The Viterbi approach added 723,738 new phrase pairs which covers 53,886 new source n-grams.

We combined the extracted phrases with the phrases extracted from the "Baseline+Extracted Top 25%" system and used in the translation experiments. The translation setup including the decoder and language model remained the same as for experiments in Chapter 4. We used the MT08 development set to optimize the decoder parameters using MER training. Two translation systems were built, one each from the phrases extracted using the Viterbi approach and the proposed approach. Translation results for Urdu-English are given in table 5.4. Results are given in BLEU, METEOR and TER metrics. For each test set, the first line gives the results for the baseline. The second and third lines give the results for the phrase extraction using Viterbi approach and the proposed approach, respectively. The fourth line is the "Baseline+Extracted All" system from Table 4.12 in Chapter 4, included here for comparison.

System	**MT08-NW (Dev)**			**MT09-NW (Test)**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline+Extracted Top 25%	23.93	29.63	63.17	26.33	30.71	60.10
+Phrases(Viterbi)	23.78	29.82	63.25	26.52	30.89	59.92
+Phrases(Proposed)	**25.03**	**30.45**	**62.44**	**28.64**	**31.05**	**58.16**
Baseline+Extracted All	22.89	28.96	63.68	25.24	30.01	61.86

Table 5.4: Translation results for Urdu-English extracted phrases when combined with the baseline corpus.

The results show that the proposed phrase extraction method is significant better than the Viterbi phrase extraction. This is despite the fact that the Viterbi extraction approach adding much more phrases than the proposed approach. It is clear that much of these phrases have not helped in improving the translation performance due to the noisy nature of the translations. When the extracted phrases are combined with the baseline system, it shows statstically significant improvement (at 95% level) for both test sets, compared to all the extracted sentences (last line in Table 5.4. This shows that selectively extracting phrase level parallel data leads to better translation performance than generating new models with the full extracted corpus.

From the experiments with Arabic-English and Urdu-English we can see that extracting parallel data at phrase level helps translation performance when the lexical coverage is limited. When the amount of available parallel data increases the effect of the extracted data diminishes.

Chapter 6

Enabling MT for Low Resource Languages using Comparable Corpora

In the previous chapters we proposed several algorithms to extract parallel resources from comparable corpora. In this chapter we use them to build new translation systems for language pairs with no prior sentence-aligned parallel data available. Results from the translation experiments in Chapter 4 revealed that we achieve the best impact from the extracted data when there is limited amount of initial parallel data. One of the aims of the proposed work in this thesis is to enable machine translation for low-resource languages such as Pashto, Dari, Urdu, and Sinhala by using the parallel resources extracted from comparable corpora. It is for these languages that we hope to see the most benefit from the extracted resources. In Chapter 4, we used Urdu-English to demonstrate the effect of the extracted data under the *small data scenario*. In this chapter we explore the *no parallel data* scenario. The language pair we used is Sinhala-English.

Sinhala belongs to the Indo-Aryan family of languages, and is one of the official languages in Sri Lanka. It has about 18 million speakers.[1] It has high inflectional morphology to indicate cases. As with many languages in the South Asian region, Sinhala has distinct diglossia: the written form and the spoken form of the language differ significantly with respect to the vocabulary, grammar and usage. Sinhala has its own alphabetic writing system, which was incorporated into the Unicode standard only recently. As a result, much of the digital content available online is not Unicode compliant and uses various ad hoc encoding schemes. Although English is widely used among Sinhala speakers, a sentence-aligned Sinhala-English parallel corpus does not exist to our knowledge. However, there are several online news websites that publish articles in Sinhala and English. Most articles are not parallel, but can be considered to be comparable. In our data collection effort, as explained in Chapter 3, we have collected a comparable news collection for Sinhala and English from several such news sources. This collection is used as the source to extract parallel content.

Weerasinghe [2002] describes an initial attempt to bootstrap a Sinhala-English translation system using news documents obtained from the web. The author mentions that the results were not satisfactory because of the poor alignment between the source and target text.

6.1 Extracting Initial Translation Lexicon

Our proposed alignment algorithms depend primarily on lexical features. Other features, for example in the ME classifier, are also mostly derived from the lexical probabilities. When we

[1] http://en.wikipedia.org/wiki/Sinhala_language

have some initial parallel data, we apply the standard word alignment algorithms to create the initial lexicon. In the case of low resource languages, especially when no available parallel data exist, a question arises as to who we can obtain the initial lexicon. We explored several different knowledge sources to obtain the lexicon as follows:

- Manually created dictionaries that are available on-line.

  Manually created dictionaries are useful, especially because they provide much cleaner word alignments. However we were unable to find any Sinhala-English dictionary that was available online. A smaller travel phrase book was available which contained a few dozen phrases that we included into our initial corpus. The Wiktionary, which is a multilingual dictionary was initially thought to be a very useful source. However, upon close inspection we noticed that most words are not linked to the corresponding foreign language word pages. We were only able to extract about 200 words from it.

- Titles of Wikipedia articles

  Titles of Wikipedia article in different languages, which are linked to each other, are in most cases reasonably parallel. By following the links to articles in other language, a parallel corpus can be automatically generated. Although there were several thousand articles in the Sinhala language Wikipedia, not all of them had a link to the corresponding English article. We were able to identify 500 Sinhala-English title pairs from Wikipedia.

- We also identified several websites that were available in both languages where the content was reasonably parallel. Moreover, the sites were arranged in such a way that

URL structure allows the identification of the aligned parallel documents.[2] Similar structural information and other clues have been used in identifying parallel article pairs in [Resnik and Smith, 2003]. We extracted data from three similar sites. After removing the non-text portion and segmenting the text into sentences, we removed the document pairs that had mismatching number of sentences. Assuming uniform sentence alignment between the documents, we extracted about 900 sentence pairs containing 15K words.

In the end, we were able to extract a corpus of 25K words. Note that the size of this corpus is considerably smaller than the initial data that was used in experiments in Chapter 4. We used this corpus to generate a translation lexicon using the standard word alignment approach. This is also used to generate a baseline translation system. A separate set of 200 sentences was manually translated to serve as development and test sets. These are translations of newswire articles, which is the same genre as the comparable data.

With the initial lexicon, we apply the sentence extraction algorithms to the comparable corpus. From the resulting parallel sentences, a subset of top ranking sentences are used to augment the initial parallel corpus. This bootstrapping process is used to improve the baseline translation system in each iteration. Similar bootstrapping strategies have been successfully used in the past [Zhao and Vogel, 2002a; Fung and Cheung, 2004a].

In section 6.3 we explore an active learning method to obtain the initial lexicon.

[2] E.g. English version at http://www.parliament.lk/secretariat/introduction.jsp
and Sinhala version at http://www.parliament.lk/languages/secretariat/introduction.jsp

6.2 Bootstrapping Experiments

The comparable document collection was selected from several online news sources which publishes news articles in both Sinhala and English. Due to the mismatch in encoding formats between different sources, only part of the collected data could be used for the extraction task. We selected articles from two sources for further experimentation: LEN[3] and LNW[4]. The corpus statistics are given in Table 6.1.

Language	Corpus	Articles	Sentences	Words
Sinhala	LEN	12,589	175,095	4,401,084
	LNW	3,114	31,978	732,372
English	LEN	7,045	105,501	2,450,836
	LNW	2,386	28,551	720,759

Table 6.1: Corpus statistics for the comparable data used in the extraction task.

Similar to the Arabic-English and Urdu-English experiments in Chapters 4 and 5, the comparable data extraction here involves three steps:

- A document alignment step, which uses the CLIR methods.
- A sentence extraction step, which uses a ME Classifier with the same set of features as in Chapter 4.
- A phrase extraction step, which uses the Algorithm 4 proposed in Chapter 5.

The document alignment process was performed identical to Arabic-English and Urdu-English: translate a source document word-by-word (up to 5 translations per source word) into a bag of word query in English and retrieve matching English documents (up to 20 documents per source document) using *tf.idf* similarity measure.

[3] http://www.lankaenews.com

[4] http://www.lankanewsweb.com

To train the ME classifier we manually annotated a set of Sinhala-English comparable sentences for comparability, using the same annotation scheme as in the previous experiments in Chapter 4. This set consists of 200 news-wire sentences drawn from the same news sources as the comparable corpus. The ME classifier trained with manually annotated sentences was used to obtain a collection of comparable Sinhala-English sentences. Due to the smaller size of the parallel corpus that was used to obtain the initial lexicon, the classification accuracy is clearly low.

Previous translation experiments on Urdu-English in Chapter 4 showed that a smaller high-precision subset of the extracted sentences results in better translation performance than using all the extracted data. Therefore, we added only the top 25% of the extracted sentences with respect to the classification probability to the baseline corpus. A new translation lexicon (i.e., IBM Model 4) was then obtained after word alignment of the combined corpus. This process was repeated for 10 iterations. In each iteration, we use the same comparable corpus to extract sentences, but with a lexicon trained with increasingly more data. The statistics of the extracted parallel sentences in each iteration is shown in in table 6.2. For each iteration the number of sentences, words and the vocabulary for Sinhala and English side of the corpus are given.

The baseline system contains only 1826 sentence pairs which are at best noisy parallel. This is considerably smaller than other smaller-sized parallel corpora (e.g. BTEC [Takezawa et al., 2002] corpus contains 20K sentence pairs). As seen in Table 6.2 the first iteration extracts data roughly five times the size of the initial corpus. The increase is roughly eight times for iteration 2, and converges towards 50K sentences in successive iterations. We notice that there is a sharp increase in the source vocabulary coverage during initial iterations, but

Corpus	**Sentences**	**Sinhala**		**English**	
		Words	**Voc**	**Words**	**Voc**
Baseline	1,826	24,040	6,817	25,424	4,699
Iteration 1	8,322	179,617	20,698	185,619	11,963
Iteration 2	14,930	363,569	14,930	352,272	14,345
Iteration 3	24,130	668,150	33,149	615,862	17,181
Iteration 4	33,976	1,043,157	39,580	928,687	19,965
Iteration 5	41,284	1,510,055	46,301	1,141,900	22,412
Iteration 6	44,611	1,755,232	46,912	1,248,391	23,317
Iteration 7	46,395	1,834,343	47,657	1,314,430	23,626
Iteration 8	48,058	1,942,878	49,876	1,367,924	23,893
Iteration 9	48,613	1,948,064	48,643	1,393,811	23,988
Iteration 10	49,628	1,969,758	49,008	1,432,124	24,068

Table 6.2: Corpus statistics for parallel sentences extracted in each iteration of the bootstrapping.

reaches 49K after 8th iteration.

In each iteration we built a translation system using the combined corpus. The same setup that was used for Urdu-English experiments in Chapter 4 was used here. The word-alignment model training and phrase table extraction methods remained the same. The language model was trained using the English side of the baseline corpus as well as a subset of the English Gigaword corpus (50M words). We used the Moses [Koehn et al., 2007] decoder for the translation experiments. A baseline translation system was built using only the initial parallel corpus. The decoder parameters were optimized using MER training on a development set (100 sentences). A separate test set (100 sentences) was used to evaluate the performance. Both sets have only one reference translation. Table 6.3 shows translation results for each iteration of the bootstrapping experiments. The results are given in BLEU, METEOR and TER metrics.

The translation results indicate that the extracted parallel data significantly improves

System	Dev			Test		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline	3.89	13.56	91.67	2.09	11.90	91.21
+ Iteration 1	10.12	16.83	86.00	7.64	16.25	84.12
+ Iteration 2	11.38	18.41	84.60	9.49	16.66	85.02
+ Iteration 3	12.10	17.40	85.81	10.06	18.82	81.81
+ Iteration 4	14.49	18.70	82.70	12.54	17.99	82.21
+ Iteration 5	**14.94**	19.36	82.09	**12.86**	**19.61**	82.26
+ Iteration 6	13.88	18.80	83.70	11.81	17.89	83.90
+ Iteration 7	14.02	19.48	84.93	11.89	18.22	87.47
+ Iteration 8	14.28	**20.05**	**81.39**	12.30	18.51	81.60
+ Iteration 9	13.96	19.83	84.09	10.57	17.76	**81.49**
+ Iteration 10	14.17	19.18	84.79	9.90	17.25	84.47

Table 6.3: Translation results for the bootstrapping experiments for Sinhala-English. Sentences extracted in each iteration are combined with the baseline parallel corpus to build a translation system.

the translation performance of the test sets. The sentence extraction process seems to be robust for smaller sized initial lexica. In the first four iterations of bootstrapping we see a dramatic improvement of translation performance across all metrics. The best performance for the test set is observed in the 5th iteration, both in terms of BLEU and METEOR. In succeeding iterations the performance starts to degrade. The OOV rate, which starts as high as 33% for the test set drops to 21% in the first iteration. After five iterations the rate drops to 6%.

Even though we start with a very small corpus, the extracted sentences in the initial iterations bring in additional data that help improve the lexical coverage. These sentences provide better lexica for the successive iterations to further improve on the performance. However, the extracted sentences may not be fully parallel. They contain non parallel sections, which introduce incorrect lexical choices that will lead to errors in successive iterations.

This is the phenomenon we observe in the results in Table 6.3.

Iteration	**# Phrases**	**# New Phrases**	**# New Src**
Iteration 1	4,477	1,238	1,074
Iteration 2	9,178	3,195	1,215
Iteration 3	9,904	6,099	1,929
Iteration 4	11,598	7,782	2,430
Iteration 5	13,132	8,854	2,766
Iteration 6	13,715	9,160	2,888
Iteration 7	13,859	9,268	2,902
Iteration 8	13,701	9,071	2,902
Iteration 9	13,881	9,213	2,921
Iteration 10	13,769	9,051	2,917

Table 6.4: Statistics on the extracted phrases in each iteration of the bootstrapping.

In the next step, we performed phrase extraction from the extracted comparable corpora. We used the same scheme that was used for Urdu-English in Chapter 5: use top 25% of the extracted sentences as parallel sentences for training the lexicon and the translation system (in Table 6.3), and use the remaining sentences for the phrase extraction task. As we did not have a manually word-aligned corpus to tune the parameters of the phrase extraction model, we used optimized parameters from the Urdu-English experiments. Although this may not be the optimal weights for the models, in a data sparse scenario as ours, this may be a reasonable alternative.

In each iteration of phrase extraction, we used the lexica trained from the extracted sentences in the previous iteration. Phrases were extracted using the Algorithm 4 for both the development set and the test set for all n-grams up to 10 words. Table 6.4 shows the statistics of the extracted phrases. For each iteration, the number of extracted phrases, the number of new phrases compared to the phrase extraction from the data in the sentence selection step, and the number of new source phrases are given in each column, respectively.

We see a similar pattern as in Table 6.2, where the initial iterations show dramatic increase in the number of extracted phrases, and in successive iterations it converging to an upper bound (around 13K phrase pairs). We notice that the number of new phrases among the extracted phrases increases rapidly in the early iterations, but converging to 9K in the latter iterations. The average length an extracted source phrase range between 1.95 and 2.09 words in the successive iterations (compared with the phrases extracted from selected sentences, which ranges between 1.13 and 1.20 words).

We combined the extracted phrases with the phrases from the sentence extraction in the same iteration. The new phrase table was used to decode the development and test sets, with MER tuning of the decoder with the development set. Translation results from the phrase extraction step are shown in Table 6.5. The results are given in BLEU, METEOR and TER metrics.

System	**Dev**			**Test**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Baseline	3.89	13.56	91.67	2.09	11.90	91.32
+ Iteration 1	10.23	17.55	87.30	9.73	17.47	85.50
+ Iteration 2	11.50	17.61	86.74	11.92	18.22	84.42
+ Iteration 3	12.39	17.82	83.95	12.52	18.73	81.22
+ Iteration 4	15.25	19.55	82.42	14.54	19.80	80.85
+ Iteration 5	**16.33**	**21.19**	81.02	**14.56**	**20.63**	81.14
+ Iteration 6	15.70	19.82	82.47	13.61	19.67	**79.54**
+ Iteration 7	15.58	19.88	**80.37**	10.83	18.17	85.97
+ Iteration 8	15.37	20.37	81.44	11.41	19.35	83.99
+ Iteration 9	14.64	19.19	82.51	12.49	18.18	81.75
+ Iteration 10	14.45	20.72	82.14	11.11	18.46	83.39

Table 6.5: Translation results (including phrases) for the bootstrapping experiments for Sinhala-English. Data extracted in each iteration is combined with the baseline parallel corpus to build translation systems.

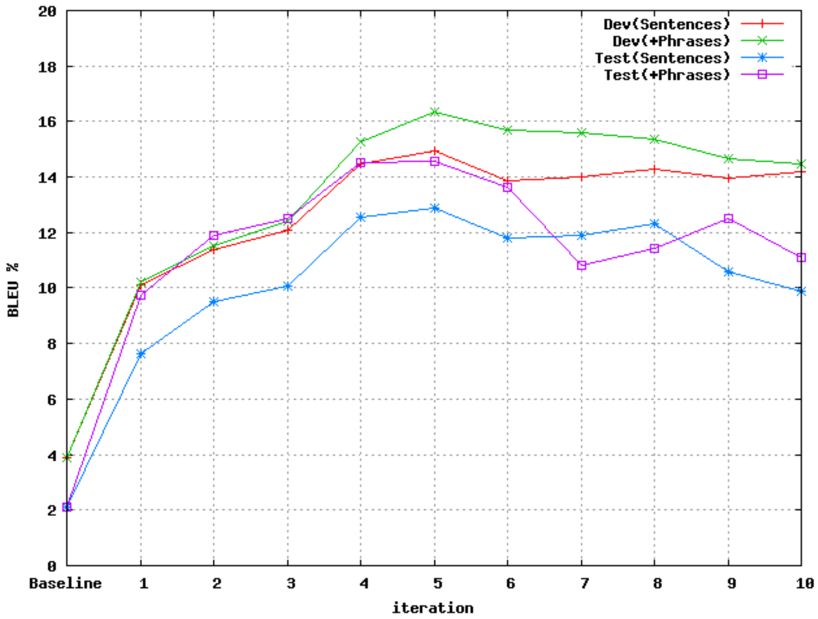


Figure 6.1: Translation performance of bootstrapping experiments with sentences and phrases.

The performance is similar to Table 6.3, where the performance increases up to the 5th iterations, and starts to degrade in the succeeding iterations. The best performance for the test set is seen in the 5th iteration, both in terms of BLEU and METEOR metrics. Figure 6.1 compares the translation performance in BLEU among the two systems: *Sentences* and *Phrases*, for the development set and the test set.

For the test set, we see statistically significant performance improvement (at 95% level) when using the extracted phrases in the initial iterations up to iteration 5. There is a drop in performance for the iterations 7 and 8, but increases again in the later iterations. For the development set there is a consistent improvement in translation performance when the new phrases are used over the extracted sentences only.

From the results it is clear that the benefit of using the extracted data is more pronounced when the initial parallel corpus is smaller where the lexical coverage of the system is very poor. As the data extraction progresses improving the lexical coverage, we see that there comes a point where the quality of the extracted parallel data is more important than improving the lexical coverage. It may be necessary to introduce reliable parallel data to push an MT system beyond this point.

6.3 Active Learning Methods to Obtain a Translation Lexicon

The parallel data detection techniques proposed in previous chapters use two primary resources: a translation lexicon built using some initial parallel corpora, and a manually annotated dataset which contains both parallel and partially parallel sentences that are used to train a classifier. As we have noticed in the previous section, it is a challenge to obtain these resources for low-resource languages. Even if some bilingual data is available, it may not be of good quality. In [Ambati et al., 2011], we investigated if such resources can be elicited at low cost, specifically using active learning techniques.

In active learning, the learner has access to a larger pool of unlabeled data, and a smaller set of labeled data. The task of the learner is to select the most informative instances from the unlabeled data set, and seek annotations from a human expert. The annotated data can then be added to the existing labeled data to re-train the underlying supervised model to improve performance.

Active learning strategies have been applied in SMT for selecting the most informative sentences to train the models. The motivation in such applications is to reduce the cost of data acquisition. Eck et al. [2005] apply active learning as a weighting scheme to select sentences in order to port their MT system to devices with limited resources such as PDAs. As the selection criterion, they use the novelty of the n-grams compared to the already extracted sentences. Gangadharaiah et al. [2009] sample sentences from a larger parallel corpus, based on the expected future usefulness of sentence pairs in improving the translation performance. Haffari et al. [2009] presents a framework for active learning within SMT where sentences are selected based on various strategies to improve both coverage as well as improve model reliability.

In our experiments, we used the task of detecting parallel sentences from a collection of comparable sentence pairs using a maximum entropy classifier, similar to the one we used to detect parallel sentences in Chapter 4. A small seed parallel corpus is used to train the initial translation lexicon that is used to generated the features for the classifier. A small labeled data set is also used to train the classifier. (In contrast to annotating sentence with the percentage of parallel words presented in Chapter 4, here we only label them as being either parallel or non-parallel). The goal of the active learner is to select the optimal set of instances from the comparable sentences so as to maximize the accuracy of the classifier.

We notice that the size of the seed parallel corpus largely affects the accuracy of the classification task. Figure 6.2 shows the size of the seed parallel corpus that was used to train the translation lexicon on the x-axis and the classifier accuracy on the y-axis. The sentences were drawn randomly from an Urdu-English parallel corpus. The plot shows that a larger seed corpus leads to a better classifier performance.

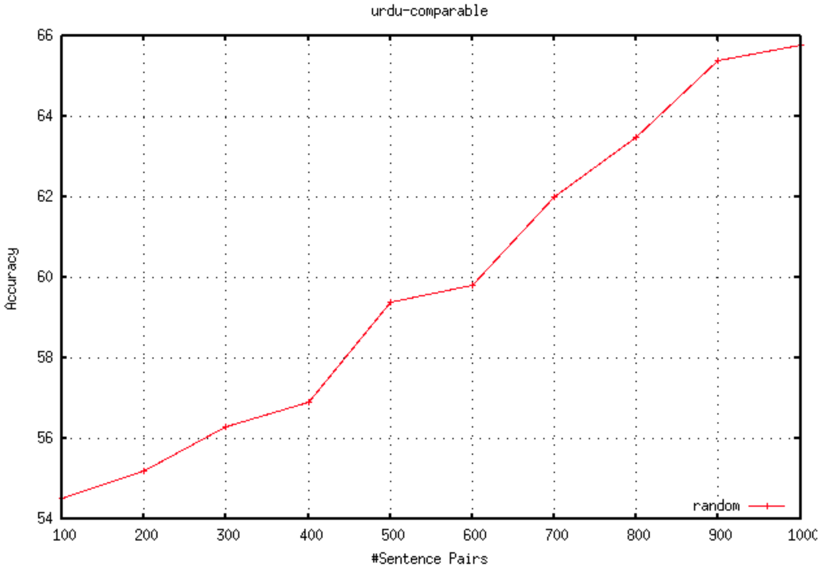


Figure 6.2: Size of the seed parallel corpus vs. classifier performance for Urdu-English.

6.3.1 Active Learning Setup

The active learning process starts with an unlabeled dataset $U_0 = \{x_j =< s_j, t_j >\}$ and a seed labeled dataset $L_0 = \{(< s_j, t_j >, c_i)\}$, where $c \in \{0, 1\}$ are class labels with 0 representing the non-parallel class and 1 representing the parallel class. We also have $T_0 = \{< s_k, t_k >\}$ which corresponds to parallel sentences identified in L_0 that will be used in training the translation lexicon. In our experiments, we started with smaller sets of T_0 and L_0 around 50 to 100 sentences pairs. An iterative active learning loop is then performed for k iterations for acquiring labeled data. We start the active learning loop by first training a translation lexicon using the available parallel data for the iteration T_k. A feature set is then obtained using the lexicon, and we train the classifier θ_k using the labeled dataset L_k. We then score all the sentences in U_k using the model θ_k and apply the selection strategy to retrieve a small set of instances. This set is then annotated by a human expert. The annotated dataset is then added to L_k, which will be used to train the classifier in the next iteration. The parallel sentences in the newly annotated set (i.e. those that were marked with class label 1 by the human expert) are added to T_k, which will then be used to train the translation lexicon in the next iteration. The active learning algorithm is given in Algorithm 5.

Algorithm 5 Active learning setup

```
Given Unlabeled Comparable Corpus: $U_0$
Given Seed Parallel Corpus: $T_0$
Given Classifier Training Set: $L_0$
for $k = 0$ to $k$ do
    Train Lexicon using $T_k$
    $\theta_k$ = Train Classifier using $L_k$
    while $(Cost < B_k)$ do
        $i = Query(U_k, L_k, T_k, \theta_k)$
        $c_i$ = Human Annotation $(s_i, t_i)$
        $L_k = L_k \cup (s_i, t_i, c_i)$
        $T_k = T_k \cup (s_i, t_i, c_1)$
        $U_k = U_k - (s_i, t_i, c_1)$
    end while
end for
```

6.3.2 Sampling Strategies

The selection strategies for obtaining class labels for training the classifier use the model in its current state to decide on the most informative instances for the next round of iterative training. The following two sampling strategies were used for the task:

Certainty Sampling

This strategy selects instances where the current model is highly confident. Certainty sampling is similar to the idea of unsupervised approaches such as boosting or self-training.

However, we make it a semi-supervised approach by having a human in the loop to affirm or correct the selected instances. Given an instance x and the classifier posterior probability P, we select the most informative instance as follows:

$$x* = argmax_x P(c = 1|x)$$

Margin-based Sampling

The margin-based sampling uses the difference between the probabilities assigned by the underlying model to the first best and the second best classes as the sampling criteria. Given an instance x and the classifier with posterior probability over classes $P(c_i = 1|x)$, the margin-based strategy is formulated as:

$$x* = argmax_x [P(c_1|x) - P(c_2|x)]$$

, where c_1 is the best predicted class and c_2 is the second best predicted class under the current model. For binary classification task, the margin-based approach reduces to an uncertainty sampling approach.

6.3.3 Experiments and Results

To have better control over the corpus, we simulate a low-resource scenario by using realistic assumptions of parallelism and noise level at both the corpus level and the sentence level. We start with a sentence-aligned parallel corpus and divide it into three parts. This first part is set aside to draw sentences at random. The second part is used to generate non-parallel sentences. We achieve non-parallelism by randomizing the mapping of the target sentences with the source sentences. This is a slight variation of the strategy used in [Munteanu and

Marcu, 2005] for generating non-parallel sentences to train the classifier. The third part is used to synthesize a comparable corpus at the sentence-level. We perform this by first selecting a parallel sentence pair and then padding either sided by a source and target segment drawn independently at random from the sampling pool. The length of the appended non-parallel part to be lesser than or equal to the original length of the sentence. Therefore, the resulting sentence pairs are guaranteed to contain at least 50% parallelism.

We use this dataset as the unlabeled pool from which the active learner selects instances for labelling. Because the gold-standard labels for this corpus are known, it gives better control over automating the active learning process, which typically requires a human in the loop.

We perform the experiments with data for Urdu-English. The same parallel dataset that was used for the experiments in Chapters 4 and 5 were used here. We started with 50,000 parallel sentence corpus and created a corpus of 25K sentence pair with 12.5K each of comparable and non-parallel sentence pairs. We also used two held-out datasets for training and tuning the classifier, consisting of 1000 sentence pairs with 500 each of parallel and comparable.

The effectiveness of the active learning strategy is tested by the number of queries the learner asks and the resultant improvement in the performance of the classifier. Figure 6.3 shows the classification performance (F-Score) of the classifier on the held-out set for different number of queries. The two sampling strategies are plotted along with a random baseline, where the sentence pair is selected at random. We notice that both active learning strategies: certainty sampling and margin-based sampling perform better than the random baseline. For the same effort (i.e. 2000 queries) the classifier has an increase of 8 absolute points. Another

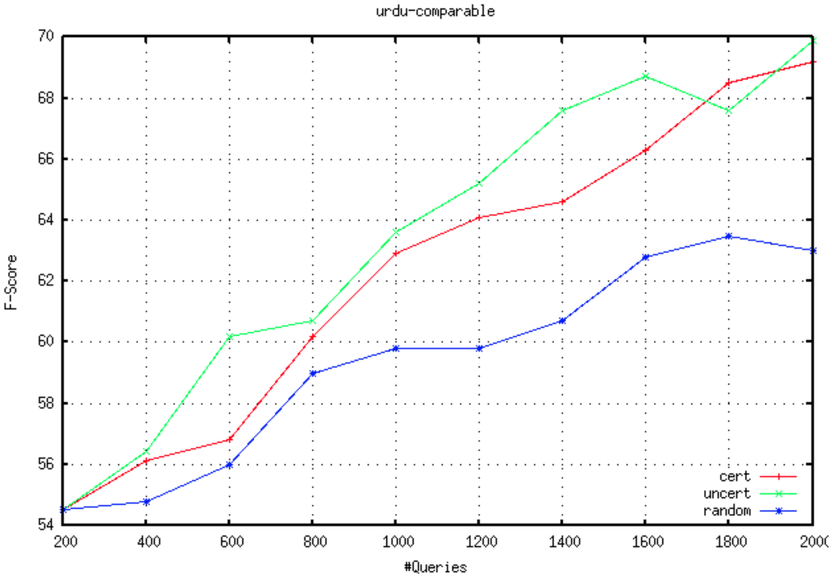


Figure 6.3: Active learning performance for the comparable sentence classification in Urdu-English.

observation is that to reach a fixed accuracy of 63 points, the random sampling method requries 2000 queries whereas the active selection strategies require significantly less effort of 1000 queries.

Chapter 7

Conclusions

7.1 Summary

This thesis explored a hierarchical framework to identify translation equivalences from comparable corpora at different levels of granularity, namely document, sentence, phrase level. In each step, part of the data extracted is used as the input to the next step. Different algorithms were proposed for each of the tasks. We evaluated the effectiveness of the proposed algorithms by applying them to extract data in 3 different data availability scenarios with respect to the available parallel data: *Large Data Scenario*, *Small Data Scenario* and *No Parallel Data Scenario*. One language pair for each of these scenarios was selected for the experimentation: Arabic-English, Urdu-English and Sinhala-English.

The extraction process starts with mining comparable document pairs. We used a cross-lingual information retrieval method for this task. A source document was translated word-by-word into a target language query, which was then matched against a collection of target documents. Different sizes of translation lexica were tested. The experimental results for an Arabic-English known document retrieval task showed that this approach is robust for smaller-sized translation lexica.

Next we extracted comparable sentence pairs from the identified comparable document pairs. For this task, we used a maximum entropy classifier. We experimented with different sets of features, and their effect on the classification task was evaluated for Arabic-English and Urdu-English. We evaluated the effectiveness of the extracted data by combining them with the available parallel data and building a new translation system. The extracted data showed significant improvements (+3.6 Bleu%) for the smallest Arabic-English system (4.5M), but showed no significant improvement for the largest system (120M). For Urdu-English system, the extracted data showed +1 Bleu% improvement. By selecting the top 5% of the extracted data with respect to the classification score, we showed that this can be further improved to +2.6 Bleu%. This also showed that the extracted data is not all parallel, and adding them directly to the training data actually hurts the performance.

To address this issue, we next focused on identifying parallel phrase pairs from comparable sentences. The rationale for this was that by selecting only the parallel phrases from the extracted sentences, we prevent the non-parallel parts from affecting the clean parallel data. We proposed a novel phrase extraction algorithm for comparable corpora, which is an extension of the PESA algorithm designed for parallel sentences. We compared the proposed approach with the standard Viterbi phrase extraction algorithm and a binary classifier that works on the phrase level. Evaluations with Arabic-English phrase extractions showed that the proposed method has a better F-Score than the other two approaches. Translation results with the extracted phrases for Arabic-English did not show a significant improvement.

However, we saw 1.21 Bleu% improvement for Urdu-English.

Finally, we used the extraction methods discussed above in an extreme case of low-resource situation, where there is no previous sentence-aligned parallel corpus available. We built a small 25K word Sinhala-English noisy parallel corpus using various sources such as the titles from Wikipedia articles, Wiktionary entries, bilingual websites, etc. A small translation lexicon was built with this corpus, which was then used to extract parallel data from a comparable corpus. A new translation lexicon was obtained by combining the extracted data with the original data, which was used in the next iteration of extraction. Both sentence extraction and phrase extraction was performed using the new lexicon. Over several iterations, this bootstrapping process showed significant improvements in translation performance.

During our research we have extracted parallel resources in several language pairs that may be useful for MT and in general NLP research community. We will make every effort to share the extracted data after resolving legal and other issues.

7.2 Contributions

This thesis explored a framework to detect and extract translational equivalences from comparable corpora, with a focus on low resource scenarios. This includes languages with low parallel resources and cross-domain translations. Translational equivalences were identified at different levels including document, sentence and sub-sentential phrase level. We summarize the thesis contributions as follows:

- We introduce a hierarchical framework to extract translational equivalences from comparable corpora. Similar frameworks have been used before by previous researchers

[Zhao and Vogel, 2002a; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005], but we are unaware of previous attempts to extract parallel phrases from comparable corpora.

- We proposed an algorithm to extract parallel phrases from comparable sentence pairs. We demonstrated that this algorithm is more accurate as measured in F_1 than the standard approaches and has significant improvement in translation performance.
- We studied the effect of different levels of initial parallel data to the extraction algorithms and demonstrated that the algorithms are robust for low-resource situations.
- We demonstrated that even for languages where no parallel data is available, a translation system can be bootstrapped using parallel data extracted from comparable corpora, and thereby enable machine translation for those languages.

7.3 Future Directions

We are excited that the parallel data extraction methods we explored resulted in improved translation performance, even in really low resource situations such as for Sinhala-English. Nonetheless, there are several directions that the current work can be further extended.

The primary source of comparable data for our experiments was newswire data. We used either previously collected articles, such as the Gigaword collection, or collected articles from specific newswire organizations such as VOA and BBC that publish articles in multiple languages. Our document selection process was limited to the articles that were present in the collected comparable corpus. This may not be an issue when large collections are available in the size of Gigaword corpora, but for languages with limited resources, this poses limitations on the amount of data that can be extracted. Large amounts of comparable data from more heterogeneous sources are available on the Web. The CLIR-based document selection can be extended so that for a given source document, comparable documents are mined from the Web. This may be especially helpful for extracting data from non-news sources, and for language pairs that are not widely covered. We have attempted this in a limited way for Haitian Creole-English [Hewavitharana et al., 2011]. Some recent work explore this idea of mining comparable documents from the Web [Shi et al., 2011; Kumar et al., 2011].

The lack of parallel data for many language pairs is the primary motivation for using comparable corpora to extract translational equivalences. When crawling comparable articles for low resource languages, we observed that these documents are available in multiple languages, giving the possibility of collecting multilingual comparable corpora. It is quite interesting to explore the current framework to extract data from multilingual corpora. Recent research on multilingual SMT has focused on using a language, usually with richer resources (e.g. English), as a pivot language to overcome the resource limitations in certain language pairs. Similar approach can be used to extract data from multilingual comparable corpora to extract data for language pairs that have low resources.

When the available amount of parallel resource are severely limited, as was the case for Sinhala-English, it may not be possible to obtain a clean parallel corpus to build an initial translation lexicon. In such situations, it is necessary to obtain it through human intervention. We investigated using active learning to obtain the initial lexicon. But due to time and other constrains we did not fully exploit this approach in translation experiments. It will be interesting to investigate how this will affect the translation performance.

For the sentence selection step in Chapter 4 we used the ME classifier, which has been successfully used by prior research work. While it is not clear that more advanced inference methods would make much difference, it would nonetheless be helpful to verify this experimentally. The set of features we used in the classifier were fast to compute and required limited resources. This was to to motivated by the fact that the language pairs were were interested in had low resource, which not always had rich linguistic resource such as POS taggers, dependency parsers, etc. However, as the target language in all our language pairs being English, which has a rich set of resources, it is possible to explore additional features based on these resource, and see the effect on the classifier performance.

Bibliography

Sadaf Abdul-Rauf and Holger Schwenk. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL?09)*, pages 16–23, Athens, Greece, 2009.

Vamshi Ambati, Sanjika Hewavitharana, Stephan Vogel, and Jaime Carbonell. Active learning with multiple annotations for comparable data classification task. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 69–77, Portland, Oregon, June 2011.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, USA, June 2005.

Julien Bourdaillet, Stéphane Huet, Philippe Langlais, and Guy Lapalme. TransSearch: from a bilingual concordancer to a translation finder. *Machine Translation*, 24(3-4):241–271, dec 2010.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

Koby Crammer, Ofer Dekel amd Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-agressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

Mark W. Davis and Ted E. Dunning. A TREC evaluation of query translation methods for multi-lingual text retrieval. In *In Proceedings of the Fourth Text Retrieval Conference*, pages 483–498, Gaithersburg, Maryland, USA, November 1995.

Mona Diab and Philip Resnik. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 255–262, Philadelphia, Pennsylvania, USA, July 2002.

Matthias Eck, Stephan Vogel, and Alex Waibel. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of the Machine Translation Summit X*, Phuket, Thailand, September 2005.

Pascale Fung. Compiling bilingual lexical entries from a non-parallel English-Chinese corpus. In *In Proceedings of the 3rd Annual Workshop of Very Large Corpora*, pages 173–183, 1995.

Pascale Fung and Percy Cheung. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 2004a.

Pascale Fung and Percy Cheung. Mining very non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 57–63, Barcelona, Spain, 2004b.

Pascale Fung and Lo Yen Yee. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 414–420, Montreal, Canada, 1998.

William A. Gale and Kenneth Ward Church. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 177–184, Berkeley, California, USA, June 1991.

Rashmi Gangadharaiah, Ralf Brown, , and Jaime Carbonell. Active learning in example-based machine translation. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA)*, 2009.

Gregory Grefenstette, editor. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998.

Mridul Gupta, Sanjika Hewavitharana, and Stephan Vogel. Extending a probabilistic phrase alignment approach for SMT. In *In Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, December 2011.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. Active learning for statistical phrase-based machine translation. In *Proceedings of HLT-NAACL 2009*, pages 415–423, Boulder, CO, USA, May 31 – June 5 2009.

Sanjika Hewavitharana and Stephan Vogel. Enhancing a statistical machine translation system by using an automatically extracted parallel corpus from comparable sources. In *LREC 2008*, Marrakech, Morocco, May 2008.

Sanjika Hewavitharana and Stephan Vogel. Extracting parallel phrases from comparable corpora. In *Proceedings of ACL/HLT 2011 Workshop on Building and Using Comparable Corpora*, Portland OR, USA, June 2011.

Sanjika Hewavitharana and Stephan Vogel. Extracting parallel phrases from comparable corpora. In *Building and Using Comparable Corpora*. Springer, November 2012.

Sanjika Hewavitharana, Nguyen Bach, Qin Gao, Vamshi Ambati, and Stephan Vogel. CMU Haitian Creole-English translation system for WMT 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 386–392, Edinburgh, Scotland, July 2011.

Almut Silja Hildebrand, Kay Rottmann, Mohamed Noamany, Quin Gao, Sanjika Hewavitharana, Nguyen Bach, and Stephan Vogel. Recent improvements in the CMU large scale Chinese-English SMT system. In *Proceedings of ACL-08: HLT, Short Papers*, pages 77–80, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. Creating corpora for speech-to-speech translation. In *In Proc. of EUROSPEECH 2003*, pages 381–384, Geneva, 2003.

Alexandre Klementiev and Dan Roth. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, pages 817–824, Sydney, Australia, 2006.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007.

Phillip Koehn and Kevin Knight. Estimating word translation probabilities from unrelated monolingual corpora using the em algorithm. In *In Proceedings of the National Conference on Artificial Intelligence*, pages 711–715, Austin, Texas USA, 2000.

Robert Krovetz. Viewing morphology as an inference process. In *In Proceedings of 16th ACM SIGIR Conference*, pages 191–202, Pittsburgh PA, June 27 – July 1 1993.

Tadashi Kumano, Hideki Tanaka, and Takenobu Tokunaga. Extracting phrasal alignments from comparable corpora by using joint probability SMT model. In *In Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation*, Skvde, Sweden, September 2007.

Vivek Kumar, Luciano Barbosa, and Srinivas Bangalore. A scalable approach for building a parallel corpus from the web. In *Proc. of Interspeech 2011*, Florence, Italy, August 2011.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Philadelphia, PA, USA, July 2002.

Robert C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas*, pages 135–144, Tiburon, California, USA, 2002.

Dragos Stefan Munteanu and Daniel Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, 2005.

Dragos Stefan Munteanu and Daniel Marcu. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 81–88, Sydney, Australia, 2006.

Douglas W. Oard. Cross-language text retrieval research in the USA. In *In Proceedings of the Third DELOS Workshop on Cross-Language Information Retrieval*, pages 1–10, Zurich, Switzerland, 1997.

Douglas W. Oard and Bonnie J. Dorr. A survey of multilingual text retrieval. Technical Report CS-TR-3615, University of Maryland Computer Science Department, 1996.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, 2003.

Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July 2002.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

Paul Ogilvie and Jamie Callan. Experiments using the Lemur toolkit. In *In Proceedings of the Tenth Text Retrieval Conference (TREC-10*, pages 103–108, 2001.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002.

Chris Quirk, Raghavendra U. Udupa, and Arul Menezes. Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of the Machine Translation Summit XI*, pages 377–384, Copenhagen, Denmark, 2007.

Reinhard Rapp. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics,*, pages 320–322, Cambridge, Massachusetts, 1995.

Reinhard Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 519–526, College Park, Maryland, USA, 1999.

Philip Resnik and Noah Smith. The web as a parallel corpus. *Computational Linguistics*, 29 (3):349–380, 2003.

Lei Shi, Cheng Nie, Ming Zhou, and Jianfeng Gao. A DOM tree alignment model for mining parallel data from the web. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006.

Simon Shi, Pascale Fung, Emmanuel Prochasson, Chi-Kiu Lo, and Dekai Wu. Mining parallel documents using low bandwidth and high precision CLIR from the heterogeneous web. In *International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, November 2011.

Noah A. Smith. From words to corpora: Recognizing translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 95–102, Philadelphia, PA, USA, July 2002.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, 2006.

Richard Sproat, Tao Tao, and ChengXiang Zhai. Named entity transliteration with comparable corpora. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, pages 73–80, Sydney, Australia, 2006.

Trevor Strohman, Donald Metzler, Howard Turtle, and Bruce Croft. Indri: A language model-based search engine for complex queries. In *Proc. of the International conference on Intelligence Analysis*, McLean, VA, May 2005.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. Towards a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of the Third Int. Conf. on Language Resources and Evaluation (LREC)*, pages 147–152, Las Palmas, Canary Islands, Spain, May 2002.

Christoph Tillmann and Sanjika Hewavitharana. A unified alignment algorithm for bilingual data. In *Proc. of Interspeech 2011*, Florence, Italy, August 2011.

Masao Utiyama and Hitoshi Isahara. Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79, Sapporo, Japan, 2003.

Stephan Vogel. SMT decoder dissected: Word reordering. In *Proc. of IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pages 561–566, Beijing, China, October 2003.

Stephan Vogel. PESA: Phrase pair extraction as sentence splitting. In *Proceedings of the Machine Translation Summit X*, Phuket, Thailand, September 2005.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, 1996.

Ruvan Weerasinghe. Bootstrapping the lexicon building process for machine translation between 'new' languages. In *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas*, Tiburon, California, USA, 2002.

Dekai Wu. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics,*, pages 80–87, Las Cruces, New Mexico, USA, 1994.

Ying Zhang and Stephan Vogel. Suffix array and its applications in empirical natural language processing. Technical Report CMU-LTI-06-010, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Dec. 2006.

Ying Zhang, Ke Wu, Jianfeng Gao, and Phile Vines. Automatic acquisition of Chinese-English parallel corpus from the web. In *In Proceedings of the European Conference on Information Retrieval*, Imperial College, London, England, 2006.

Bing Zhao and Stephan Vogel. Adaptive parallel sentence mining from web bilingual news collection. In *In Proceedings of the IEEE International Conference on Data Mining*, pages 745–748, Maebashi City, Japan, 2002a.

Bing Zhao and Stephan Vogel. Full-text story alignment models for Chinese-English bilingual news corpora. In *Proceedings of the ICSLP '02*, September 2002b.