Pattern Matching for Translating Domain-Specific Terms from Large Corpora

Thesis Proposal

CUCS-015-95

Pascale Fung

January 26, 1996

Abstract

Translating domain-specific terms is one significant component of machine translation and machine-aided translation systems. These terms are often not found in standard dictionaries. Human translators, not being experts in every technical or regional domain, cannot produce their translations effectively. Automatic translation of domain-specific terms is therefore highly desirable.

Most other work on automatic term translation uses statistical information of words from parallel corpora. Parallel corpora of clean, translated texts are hard to come by whereas there are more noisy, translated texts and many more monolingual texts in various domains. We propose using noisy parallel texts and same-domain texts of a pair of languages to translate terms.

In our work, we propose using a novel paradigm of pattern matching of statistical signals of word features. These features are robust to the syntactic structure, character sets, language of the text, and to the domain. We obtain statistical information which is related to the lexical properties of a word and its translation in any other language of the same domain. These lexical properties are extracted from the corpora and represented in vector form. We propose using signal processing techniques for matching these features vectors of a word to those of its translation. Another matching technique we propose is applying discriminative analysis of the word features. For each word, the various features are combined into a single vector which is then transformed into a smaller dimension eigenvector for matching.

Since most domain specific terms are nouns and noun phrases, we concentrate on translating English nouns and noun phrases into other languages. We study the relationship between English noun phrases and their translations in Chinese, Japanese and French in parallel corpora. The result of this study is used in our system for translation of English noun phrases into these other languages from noisy parallel and non-parallel corpora.

Contents

List of Figures

Chapter 1

Introduction

The word *Governor* can be translated into different Chinese words such as 總督, 主管(*top manager*), 總裁(*chief*), 州長(*of a State*). However, *The Hong Kong Governor* has only one translation -香港 總督 [1]. Likewise, *house* is normally translated as *maison* in French, but *House of Commerce* should be translated as *Chambre de Commerce*. In addition, domain-specific terms like *Basic Law* or *Green Paper* should not be translated literally word by word into Chinese. Every word of these terms has multiple translations in dictionaries and sometimes the correct translation is not found in any dictionary. Domain terms are often not found in ordinary dictionaries, and sometimes not even in domain-specific dictionaries. This problem is called translation of *unknown terms*.

However, as any bilingual person who has come to Hong Kong would have no trouble translating *Governor* as 總督 just from reading newspapers and journals, many *unknown* domain or regional bilingual terms can be learned from texts produced in the specific domain or region. This is due to the relative *consistency* in the mapping of domain terms.

Human translators of technical materials, not being experts in every technical or regional domain, cannot produce the correct translations effectively. In addition, neither machine-translation systems nor machine-aided translation systems can perform robustly without a domain-specific bilingual lexicon. Meanwhile, human compiled dictionaries require a lot of human expertise in the particular domain, which in technical and scientific areas are ever evolving, producing new terms. There is a huge amount of human power, time and money involved in updating existing dictionaries or in compiling new ones. Therefore, cheap, fast and automated domain term translation algorithms are in great demand.

The consistency of domain term mapping in different languages give rise to certain patterns of usage of these terms and their translations. It is possible to deduce term translations from such patterns by using statistics. With the advance in computational power of machines and the increasing availability of large quantities of electronic texts, much statistical work has been done in the area of translation and lexicon compilation. In our work, we propose a novel statistical learning paradigm, augmented by linguistic knowledge, for translating domain-specific terms.

[1]The translation of *Hong Kong Governor* in Chinese has two forms, the full term 香港 總督 and the acronymic form 港督, which consists of the second character of *Hong Kong* and the second character of *Governor*.

1.1 Problems

There are some key problems to solve in order to achieve the goal of translating domain terms. The main issues we address in our work include different types real world data to use for this task and the robustness of system algorithms. These issues are briefly described as follows.

1.1.1 Finding domain term translations from noisy parallel corpora

Most previous statistical algorithms for compiling bilingual lexicon entries use statistical information derived from clean, sentence-aligned translated documents in a specific domain such as the Canadian Hansard Corpus (Brown *et al.* 1991; Kay & Röscheisen 1993; Gale & Church 1993; Church 1993; Chen 1993; Wu 1994). From a matching list of paired sentences in the two languages, these approaches try to derive co-occurrence patterns of certain words across the language: if word A in the source language occurs in sentences x, y, m, n, etc., and word B in the target language also occurs in the same sentences or mostly in the same sentences, word A and B are extracted as a bilingual word pair in the lexicon. The co-occurrence patterns of words are described by mutual information scores, z-scores, hidden parameter estimation or other kinds of correlation measure. These correlation measures can be interprated either as frequency statistics or likelihood. Word pairs with the highest correlation measure with respect to other words are regarded as translations of each other. This kind of statistic relies on the frequency and coverage of the matching list, and therefore depends on bilingual texts which are sentence-to-sentence translations of each other. However, such bilingual texts are not common and are usually limited to transcriptions of government debates or documents. Moreover, inserting sentence boundaries and matching bilingual texts sentence to sentence are time-consuming, raising the cost of using a large quantity of bilingual texts for statistical bilingual lexicon compilation. Many documents in most other domains are not sentence-to-sentence translations. One example is the technical manuals of various computer programs. The electronic form of the AWK manual(Aho *et al.* 1980) in English and its translation in Japanese, for example, contain many *noisy* translations, such as one sentence to many translations, different programming examples for the same AWK command, pictures files in the English version replaced by a simple link in the Japanese version resulting in non-parallel segments, etc. While such noisy corpora are basically parallel, they cannot be handled by traditional alignment programs which typically work on a sentence by sentence basis and thus cannot ignore a chunk of text in either source or target. In addition, there are many texts available in optical character recognition(OCR) form. This type of text often has visual noise causing missing or unclear sentence boundaries. It clearly would be beneficial to devise algorithms which extract bilingual word pairs from bilingual noisy parallel texts without relying on sentence boundaries.

1.1.2 Finding domain term translations from nonparallel corpora

There is yet a larger source of texts for statistical bilingual work, namely non-parallel, same domain texts in different languages. For example, one can find books on software engineering in English, Chinese, Japanese, French or many other languages written by different authors. These are same domain monolingual texts. These books are often not translations of each other. However, the usage of technical terms is the same - *program* in software engineering texts most likely refers to computer codes than to, say, a TV program if the domain were entertainment.

Turn to the pages of newspapers around the world. Articles about Bosnia contain the same usage of the term *bombarded*, and probably a different usage of the same word when the subject is the U.S. Speaker of the House. Articles on micro-electronics will have consistent usage of the word *chip* which is different from that in any article on agriculture. One can envisage using AP newswire samples, articles and books of the same domain or same topic (e.g., financial news) in different languages to infer term translation. In most domains, there is no translated parallel corpus available in all languages, but there are certainly monolingual texts in these languages. Non-parallel text bilingual processing would tremendously increase the amount of data available to be used as input, and the number of domain-specific bilingual lexicons produced.

Last but not least, the existence of a parallel corpus in a particular domain means *some* translator has translated it. Therefore, the bilingual lexicon compiled from such a corpus is at best a reverse engineering, *re-translation*, of the lexicon this translator used. On the other hand, if we can compile a lexicon of domain-specific terms from non-parallel corpora of monolingual texts, the results would be much more useful.

The challenge of translating domain-specific from non-parallel data lies in the large differences of syntactical, semantic, even discourse structures between two non-parallel texts. Statistical correlation function between word pairs cannot be found using parallel information because there is no matching paragraphs, or sentences in the non-parallel texts.

1.1.3 Cross language group bilingual processing

Another challenge for our work is to develop domain term translation algorithms that perform efficiently on Asian/Indo-European language pairs as well as on Indo-European language pairs. Most other algorithms have been developed and tuned for European language pairs. European languages have somewhat common linguistic heritage leading to closely related morphological, lexical and syntactical structures. Algorithms developed on European language pairs take advantage of such common features between the languages and rely on, for example, common cognates, similar sentence length, similar expressiveness, similar morphology. Such algorithms cannot be applied to language pairs which differ greatly from each other in terms of character/alphabet sets and linguistic structures such as Japanese/English or Chinese/English. In cases where sentence lengths were used as the correlation feature, the performance of the algorithm relies on the one-to-one-ness of clean, parallel texts. In fact, experiments with both cognate-based algorithms (Church *et al.* 1993) and length-based algorithms (Wu 1994) showed lower performance when used on Japanese texts containing English terms and on Chinese texts. This is to be expected since the syntactic structure of Asian languages is quite different from Indo-European languages and translations are also likely to be less literal than across Indo-European pairs.

Important information for bilingual work on European language pairs is the shared morphology. As an example, (Church 1993) uses cognates, or identical character sequences, as points of alignment. Such sequences are quite common in European languages (Simard *et al.* 1992), which often have words with shared root. It is also common that the source word or phrase is left intact in the translation. For similar historical reasons, it might be possible to use the Chinese character sets common to both Chinese and Japanese to align Chinese/Japanese bilingual corpora. However, shared morphological information clearly does not exist for Asian/European language pairs. Our work is to find correlation features between words in any language pairs which are independent of character sets or language groups.

In addition, computational linguistics in many Asian languages, especially in Chinese has a short history. Many monolingual features taken for granted in European languages such as word segments, part-of-speech tags, parsing, grammar, etc are still at the early stage of academic research. Linguistic definitions for these features are not consistent. Very few tools have been developed to process Chinese. However, this is a necessary first step in translating Chinese words into other languages. The various research topics include how to define word boundaries, word part-of-speech and one-to-many mapping of a word and its translations.

1.2 Approach

The prevalence of the statistical approach in research on unknown term translation stems from three main reasons:

First, using pure linguistic knowledge such as thesauri and word net or tree banks requires large amount of time-consuming human labor. Highly specialized linguists and lexicographers are needed for translating domain terms from each pair of languages. Whenever the input pair of languages changes, new linguists/lexicographers would need to be hired. This is very costly. Statistics-based algorithms using real world data as input can be used to propose a rough translation to the human translators cheaply and fast.

Second, the time-changing characteristic of domain specific terms exacerbates the knowledge acquisition bottle-neck. As a result, dictionary updates usually take place only once in many years. On the other hand, the biggest advantages of computers are fast computation and large memory. These advantages allow efficient fast and reliable acquisition of statistical information of large databases. Efficient updating of domain information is possible whenever new data is available.

Third, statistical information can reveal patterns of term usage not apparent at first to humans. The mean, covariance, standard deviation, and other statistics of word usage can be easily computed and manipulated to show certain patterns of the text. This information is not immediately obvious to any human translator. For example, (Mosteller & Wallace 1984) used statistical models to determine the authorship of a much disputed part of the *Federalist Papers* based on frequency patterns of words.

1.2.1 Statistical word features

As demonstrated in all statistical bilingual lexicon compilation algorithms, the foremost task is to identify word features which are similar between a word and its translation, yet different between a word and other irrelevant words in the other language. In most other algorithms with parallel texts as input, the feature used is the positional co-occurrence of a word and its translation in the other language in the same sentences. (Brown *et al.* 1993; Kupiec 1993; Smadja & McKeown 1993; Dagan *et al.* 1993; Wu & Xia 1994). Moreover, feature representation in these other statistical translation algorithms is usually in the form of likelihoods— e.g., likelihood of a word in English given the corresponding French word, according to its occurrence or distribution in the texts. These likelihood scores are reliable only if the occurrence patterns are reliable, such as in a clean parallel corpus with sentence to sentence correspondence. For noisy parallel texts, we need to model the positional co-occurrence of words more robustly than sentence-based algorithms, overcoming the effect of inserted paragraphs, more free translation, and occasional segments of different texts.

For non-parallel texts, since the texts are not translations of each other, neither positional co-occurrence nor occurrence frequency of words can be expected to show any correlation between word pairs. There is no direct sentence-to-sentence or segment-to-segment mapping. We need to go a step beyond to find other kinds of statistical and linguistic features relating pairs of translated words. These features will need to be consistent between words and their translations, domain-dependent, but independent of the language. Likewise, the features should be independent of word order in a sentence, sentence structure in a paragraph, or overall text structure and style.

As explained previously, shared morphological features do not exist between Asian and Indo-European language pairs and thus cannot be used as word features.

On the other hand, we postulate that the *usage* of domain-specific terms is somehow consistent between such texts. For example, the noun *figure* used in scientific and technical articles often refers to a diagram instead of being used as a verb. Consequently, it appears in phrases such as "*This is shown in Figure 1.*" in technical articles in all different languages. In addition, we postulate that the *relationship* between one domain-specific term and another in the same language might form a pattern in these texts. For example, in financial news, the word *market* often refers to *stock market* and is often used in conjunction with other terms related to the stock market. We also think that the immediate context of a pair of words should somehow correlate. If *Hong Kong* is followed by *Governor, government, University,* etc., in some governmental document, then its counterpart in Chinese would also be followed by the Chinese equivalent of *Governor, government, University,* etc.

Our work is to explore these intuitions and represent these usage patterns of domain terms in terms of context and word relation statistics. We would also like to represent these features in vector and signal form for pattern matching. We propose various feature representations such as dynamic recency vectors, position binary vectors, context heterogeneity and context length histograms.

1.2.2 Pattern matching of word features

Once we have found the correlating features between a pair of translated words, we need to develop matching functions between the features. The matching function should give high correlation scores relating the pairs of word features, and low correlation scores otherwise. Mutual information score, t-score, Dice coefficient are some of the most prevalent correlation functions being used by statistical NLP algorithms in applications such as sentence-alignment (), word sense disambiguation (), word and term translation (), etc. However, these correlation scores are based on likelihood or relative frequency of the two words being matched. For example, relative frequency describes how likely *house* is translated into *maison* or *chambre* by counting how many times the two words are mapped to each other in translations. But the likelihoods are derived from *seen* data, namely a clean aligned parallel corpus. If the corpus is noisy, these probabilities cannot be obtained reliably. If the corpus is non-parallel, such co-ocurrence measures would be meaningless.

We propose using non-linear matching functions derived from signal processing techniques such as Dynamic Time Warping(DTW) for correlating pairs of recency vectors from a noisy parallel corpus. DTW is capable of eliminating noise. After we obtain reliable anchor points, ie. having eliminated the noisiness of the corpus, we propose using mutual information and t-score on binary position vectors from the resultant parallel corpus.

For word feature vectors in non-parallel corpora, we propose using Euclidean distance measure and eigenvalue matching for linear mapping when appropriate and other signal matching techniques for non-linear mapping when needed.

1.2.3 Linguistic knowledge for term translation

Although there appears to be no pure linguistic solution to the problems we are addressing, linguistic information can provide some insights to our search for word feature representations and matching functions. In addition, linguistic knowledge can be applied as a filter to the statistical engine at various stages of our lexicon compilation system.

For example, part-of-speech tags can provide constraints to matching. Given a tagged corpus, we can concentrate on translating only the nouns and noun phrases, increasing algorithmic efficiency.

Since there is a great deal of nominalization in domain specific terminology, one goal of our work is to translate domain specific noun phrases. Linguistic knowledge about noun phrases, bracketing as well as statistical parsing in English and other languages will be used in our work. We will use part of the English grammar for noun phrases for extraction. Linguistic knowledge about simple Chinese and Japanese noun phrases or compound nouns can also be used in our work.

Moreover, if we find out the relationship between English noun phrases and their translations, not necessarily noun phrases in the other language, we can constrain the matching by filtering out bilingual pairs of words whose POS tags do not entitle them to be part of a translation pair.

1.3 Contributions

Overall, the contributions of our work will be:

- We propose new algorithms for term translation from **noisy parallel corpora** of languages which do not share common linguistic roots. Most parallel corpora are not clean, sentence-alignable texts. Our algorithm expands the input domain for bilingual research.

- We propose new algorithms for finding domain term translation from same domain, **non-parallel corpora**. This will relax the constraint of parallel corpora, greatly increasing the type of domain data we can use for term translation. Our new algorithms will include finding statistical word features from context to represent domain words and non-linear matching functions derived from signal processing techniques.

- We have identified the **non-linear recency vector** and the **position binary vector** as novel word features for domain dependent bilingual term translation from noisy parallel corpora. These features are more robust than conventional position co-occurence and word frequencies as features. We have also identified context heterogeneity, context length histograms as word features in non-parallel corpora. These features can be combined to a single **vector representation**, enabling efficient matching by standard signal processing and pattern matching techniques.

- Usage of **signal processing techniques** for translation opens up a new toolbox for pattern matching in the translation problem. Dynamic Time Warping and space-frequency transformation are examples of common techniques for matching signals in speech and image processing. We transform our term translation problem into the signal domain, enabling efficient matching of word features.

- Our algorithms address the problem of translation of domain terms **across language groups** instead of between Indo-European languages. Our algorithms are more language-robust. We do not relay on common cognates or similar morphological and syntactical information frequently used by other translation algorithms.

- We study the relationship between English **noun phrases** and their translations, not necessarily noun phrases themselves, in Chinese and other languages from aligned, parallel corpora. The result of our findings can be used to improve translations of English noun phrases in non-parallel corpora.

1.4 Organization of this proposal

In Chapter 2, we describe previous and current work in the area of statistical NLP related to our work. We survey on algorithms for sentence alignment, word and term translation, various correlation functions as well as Chinese unknown word extractors. We also discuss the relevance to our work in each case. We then describe our algorithm for noisy parallel corpora processing for bilingual lexicon compilation in Chapter 3, and our word feature extraction and matching algorithms for this application. Chapter 4 describes our previous and current work on extracting bilingual word pairs from same domain, non-parallel texts. We also describe the statistical part of feature extraction and how to use linguistic knowledge to help translation. In Chapter 5, various steps of the remaining work will be presented. We also describe in this chapter how our algorithms for statistical word feature extraction and matching functions can be integrated with linguistic knowledge into a domain term translation system for non-parallel corpora. Chapter 6 is a summary of the various contributions of this thesis work. Finally Chapter 7 describes the limitations of this thesis work and points out possible future directions.

Chapter 2

Related Work

With the advent of large corpora, there has been a surge of work on parallel text. Most of this work has focused on European language pairs, especially English-French. It has been found that when extended to languages such as English-Japanese and English-Chinese, many of the algorithms are heavily constrained or have limited application. In this chapter, we will describe related work in sentence alignment, word translation, Asian language processing and noun phrase translation.

2.1 Sentence alignment

Much work has been done using bilingual parallel texts for matchine translation. Since this work is based on the premise that sentences in these texts are aligned first, there have been quite a number of papers on sentence alignment. Although our work focuses on bilingual processing *without* sentence alignment, we feel that the methodology of the alignment work is still relevant here. There are two main approaches for sentence alignment, namely text-based and length-based alignment. The former makes use of lexical information in the sentences and the latter makes use of the total number of characters or words in a sentence.

- Warwick-Armstrong & Russell (1990) propose using a bilingual dictionary to select word pairs in sentences from a parallel corpus, and then align the sentence pairs containing such word pairs.

- Brown *et al.* (1991) use a hidden Markov model for the generation of aligned pairs of corpora. The parameters of the model are estimated from a large parallel corpus. The implementation made use of text-specific comments and annotations. The comments and annotations are used in a preprocessing step to find anchors for sentence alignment. On average, there are only ten sentences in between the known anchors points. This method is clearly text-specific and would be difficult to generalize.

- Chen (1993) uses lexical information to align sentences by an EM-based parameter estimation method. It produces a small lexicon on the fly while performing alignment. The main disadvantages are, first, it needs manual alignment of 100 sentences for bootstrapping and second, it is tens of times slower than length-based alignment algorithm. In addition, it is also unclear how much the algorithm depends on the similarity in word order between two languages or how it can be extended to Asian/Indo-European language pairs.

- Gale & Church (1993) use character-based sentence lengths as a basis for alignment. Their algorithm is based on a probabilistic model of the distance between two sentences, and a dynamic programming algorithm is used to minimize the total distance between aligned units. Their algorithm assumes that each character in one language give rise to, on average, some relatively constant number of characters in the other language. This assumption is later shown to be inaccurate for Chinese-English sentence pairs by Wu (1994).

- Simard *et al.* (1992) suggest that language-specific knowledge would be helpful for alignment, and propose using common cognates between language pairs as alignment anchors. In European language pairs, cognates often share character sequences, e.g., *government* and *gouvernement*. Such anchors are more robust to insertion and deletion than length-based algorithms. But they only show that this method could be used in combination with length-based algorithm, not alone.

- Church (1993) implement a cognate-based tool, *char_align*, to align texts. This method has the advantage of not requiring clear and well-defined sentence boundaries in the texts. Its disadvantage is that it cannot be extended to language pairs which do not share identical cognates.

- Church *et al.* (1993) reports some preliminary success in aligning the English and Japanese versions of the AWK manual (Aho, Kernighan, Weinberger (1980)), using *char_align* (Church, 1993), a method that looks for character sequences that are the same in both the source and target. The *char_align* method is designed for European language pairs. In general, this approach does not work between languages such as English and Japanese which are written in different alphabets. The AWK manual happens to contain a large number of examples and technical words that are the same in the English source and target Japanese.

- Wu (1994) uses a similar algorithm to that of Gale & Church (1993) for aligning Chinese-English sentence pairs. In addition, the algorithm also makes use of a small lexicon as anchor points (e.g. translations of numerals, weekdays). The algorithm requires clear sentence boundaries and assumes one-to-one sentence translation.

- Kay & Röscheisen (1993) starts with a partial alignment of words to induce a maximum likelihood alignment of the sentence level, which is in turn used to refine the word level estimation in the next iteration. The algorithm appears to converge to the correct sentence alignment in a few iterations. The similarity of two sentences is estimated by the number of correlated constituent words. The correlation of constituent words is estimated on the basis of the similarity of word distributions and the total number of occurrences. Some morphology information is used to map words to their root form. An article from Scientific American and its German translation Spektrum der Wissenschaft is used as test data. This algorithm also requires clear sentence boundaries.

2.2 Word and term translation

Some of the algorithms used for alignment[1] produce a small bilingual lexicon as a by product (Kay & Röscheisen 1993; Chen 1993). Some other algorithms use sentence-aligned parallel texts to further compile a bilingual lexicon (Gale & Church 1991; Dagan *et al.* 1993; Kupiec 1993; Wu & Xia 1994; Dagan & Church 1994; Smadja & McKeown 1993). Note that all of the following algorithms, with the exception of (Dagan *et al.* 1993; Dagan & Church 1994), require clean, sentence-aligned parallel text input and therefore are limited in their applications.

- Brown *et al.* (1990); Brown *et al.* (1993) are the first to use a stochastic sentence translation model. Estimation-Maximization is used to estimate the parameters for the model. Their model produces word alignment from clean, sentence-aligned parallel corpora during EM estimation.

- Gale & Church (1991) propose to using mutual information and t-scores to find word correspondences as an alternative to the IBM model. They note that there is an explosive number of parameters to be estimated in the EM model when the vocabulary size is large. They use a progressive deepening method, starting with a small region of the training corpus and find the best pairs of words according to their mutual information and t-score. At subsequent iterations, the training sample is increased and more word pairs are found. This work laid the basis for other translation algorithms using correlation scores.

- Kay & Röscheisen (1993) partially align words to their translations as part of the sentence-alignment task. The result is a smal bilingual lexicon of English and German.

- Kupiec (1993) uses as input a sentence-aligned, tagged French-English corpus, and outputs a list of translated noun phrases. Simple noun phrases are extracted from the corpus and grouped as single terms. The noun phrases which appear in the same sentences are considered to have strong correlations and thus translations of each other.

- Dagan *et al.* (1993) describes a tool, *word_align*, which uses the aligned output from *char_align* to further refine the alignment at the word level. Words are matched across the two languages within a window size around the anchor points found by *char_align*. The final matching between words is found by an EM-based learning method. This tool is an improvement on the IBM word alignment model because it only requires a rough initial alignment and but it also assumes that the texts within a window size are translations of each other.

- Wu & Xia (1994) computed a bilingual Chinese-English lexicon from a strictly one-to-one, sentence-aligned and selected bilingual corpus processed by the algorithm in (Wu 1994). The estimation of the word matching is an EM-based model. Various filtering techniques are used to improve the matching.

- Smadja & McKeown (1993) propose a collocation translation algorithm using sentence-aligned, parallel texts. Individual words in an English collocation are translated into French

[1] Some of the work cited actually finds the correct word ordering in the translation, while others do not. It was argued that word alignment should only refer to those which deal with word ordering in the translation. However, we decide to follow the convention in the literature to include word correspondence work in the alignment work.

words by Dice coefficients based on cooccurrence information in aligned sentence pairs. The contribution of this algorithm is its ability to translate collocations of words, its disadvantage being the reliance on clean, sentence-aligned parallel texts of European languages. Moreover, whereas the sentence-alignment step can be replaced by other kinds of alignment in order for their algorithm to perform on noisy parallel corpora, there is no immediate way to extend their system to deal with non-parallel corpora where cooccurence information is missing and hence would render Dice coefficients inappropriate.

- Dagan & Church (1994) use the output from *word_align* and a part-of-speech tagger() to develop a tool for aiding human translators. *Termight* finds noun phrases in English and then uses *word_align* to align the head and end nouns of the noun phrases to the words in the other language. The word sequence in the other language starting and ending with the aligned words of the English head and tail nouns is considered as the translated noun phrase. This work emphasizes *Termight*'s practicality and efficiency as translator aid. Its advantage over (Smadja & McKeown 1993) is that it would also find translations for infrequent noun phrases as long as the head noun of that noun phrase is frequent. Its disadvantage is that it cannot find collocations with flexible distances. The accuracy of *Termight* is also relatively low.

It is possible to align a noisy parallel corpus, and feed the output into the above-mentioned systems and obtain word or collocation translation. Our work on noisy parallel corpora can be used to replace the initial alignment step of these systems. However, none of the above systems can be used on non-parallel corpora because of the missing coocurrence statistics in those corpora. Our work on non-parallel corpora will be an alternative to these systems because we will use correlation measures other than coocurrence statistics.

2.3 Asian language word extraction and segmentation

The prerequisite for translating terms is to extract terms from the texts. To translate words, we need to identify words first. Words are roughly delimited by spaces in European languages and therefore easy to identify as strings of characters between spaces and punctuations. In Chinese and Japanese, however, there is no space as delimiters. To tokenize words, various Japanese and Chinese *segmenters* have been used. We have developed a domain word extractor called CXtract to augment a statistical segmenter. In this section, we will describe relevant work in Chinese segmentation and domain word extraction as well as a tool for Japanese tokenization:

2.3.1 Chinese word segmentation

We co-developed with HKUST a dictionary-based Chinese segmenter and tagger (Fung & Wu 1994; Wu & Fung 1994). The dictionary can be augmented statistically by a domain word extractor, CXtract. The segmenter looks for maximum length characters sets which match the dictionary items, and the tagger tags the lexical items by a Viterbi search based on a statistically trained model. CXtract was developed by first extending and modifying Xtract(Smadja 1993), and then applying Chinese linguistic filters.

We used statistical frequency and distribution information to find character strings which are likely to be a single word or a single term. These terms are in general domain specific and not found in common dictionaries. Most Chinese and Japanese segmenters rely on dictionary look-up and therefore can be greatly improved by a domain term extractor such as CXtract. Language-specific linguistic filters can further improve the performance of such an extractor by eliminating non-words. This work is described in more detail in (Fung & Wu 1994; Wu & Fung 1994).

In general, Chinese word segmentation approaches fall into two major categories, rule-based and statistical. In addition, some approaches attempt merely to segment character sequences into words that match known lexical entries, whereas others attempt to handle unknown words as well. We briefly describe other Chinese segmenters with unknown word extraction.

- For unknown word resolution, there are other methods which use syntactic rules only, such as that of Wang *et al.* (1991), which used parsing rules to identify unknown words according to (1) replication of character patterns, (2) numbers, or (3) prefix and suffix. Since the identification method is restricted to depend entirely on the rules, it cannot be applied to find words which do not follow the parsing rules.

- Others (Chiang *et al.* 1992; Lin *et al.* 1993; Nie *et al.* 1994) use a combination of morphological rules and statistical methods to identify unknown words. Chiang *et al.* and Lin *et al.* take the approach of using morphological rules to identify regular unknown words, combined with a probabilistic method for identifying irregular unknown words. The morphological rules they employed might complement those we have used; however, they do not give their rule sets. Their irregular unknown word identification method counts the statistics of characters considered to be part of an unknown word in a training set, and uses these counts to estimate the probabilities used for the test set. The main problem of this approach, as noted by Nie *et al.*, is that these probabilities are based on single characters whereas most Chinese words are character bigrams or longer.

- Among currently existing algorithms, Nie *et al.*'s (1994) approach[2] is the most similar to our CXtract approach. They also use statistical methods to find unknown words though their statistical measures are based only on frequency of n-grams and therefore extract many freely grouped n-grams which are not words. In contrast, CXtract uses significant measures and filters out many non-words (Fung & Wu 1994). They also used some linguistic filters to process the statistically obtained words. Their rules overlap some with the linguistic filters we used in CXtract, but also include some we didn't use. One advantage of their method over CXtract is that their algorithm also count a shorter legitimate string as a word even if it is embedded in a longer legitimate string whereas CXtract takes the longest match approach. In Chinese, it might be useful to record the shorter string as a word also.

- Sproat *et al.* (1994) use a finite-state transducer with costs to estimate the probabilities of a sequence of characters being a word. Handling of unknown words is restricted to several predetermined forms, namely morphological affixes, Chinese personal names, and transliterations of foreign words in Chinese. To our knowledge, they have not reported on how other types of unknown words might be handled.

[2] Which was developed independently and first reported concurrently with CXtract.

2.3.2 Japanese morphological word segmenter

The most well-known and probably the most developed Japanese word segmentation tool is JUMAN (Matsumoto & Nagao 1994). JUMAN is a morphological analyzer which uses various part-of-speech, inflection type, connectivity and morpheme dictionaries to decide word boundaries. Japanese is written in three different character sets - Chinese characters, Hirakana and Katagana. The later is used mostly to transcribe foreign terms (e.g., computer → con puyou ta). The boundaries between the character sets correspond fairly well to word boundaries. Japanese, unlike Chinese, is also a inflectional language with temporal and person conjugations. This greatly facilitates a morphological analyzer. However, part of some words can be written by different types of Chinese characters or can be replaced by Kana characters. There are other cases where some Kana characters could be omitted. Matsumoto & Nagao (1994) developed a variation dictionary of all possible variations on Japanese words to be used for JUMAN.

One feature of JUMAN is to allow customization by the user on the various part-of-speech classification, inflection types, costs of searches on the dictionaries etc. Since part of CXtract without Chinese morpho-syntactic filters is purely statistical, it is conceivable that this part of CXtract can be used to customize JUMAN to produce a domain-specific Japanese dictionary for better performance.

We have been using JUMAN as a Japanese word extractor for all our experiments with Japanese texts, and its performance is highly reliable.

2.4 Noun phrase extraction

We have chosen to concentrate on the translation of noun phrases in our task of domain term translation. The reason is that most domain terms are nouns and noun phrases. Traditional noun phrase extraction involved parsing a sentence completely and then identifies the noun phrase part. This was considered to be a rather complex and tedious task. Instead, recent approaches have been focusing on partial parsers or regular expression extraction of noun phrases. Such parsers and extractors can also be probabilistic, using statistics from hand-marked corpora.

There are quite a number of tools for English and other European languages:

- Church (1988) reports a non-recursive simple noun phrase extractor. This bracketer finds the minimal-length noun phrases non-recursively from a tagged English text. It is a probabilistic bracketer based on starting noun phrase matrix and ending noun phrase matrix.

- Rausch *et al.* (1992) propose a nuclear noun phrase extractor which inserts brackets around noun phrases consisting of sequences of determiners, premodifiers and nominal phrase in tagged Swedish texts.

- hua Chen & Chen (1994) propose a English NP extractor combining statistical method and rule-based method. A probabilistic parser is used to find out the best chunk sequence. Linguistic knowledge is then used to assign a syntactic head and a semantic head to each chunk. Finally, a finite-state mechanism is employed to extract the most likely noun phrases according to the statistical and linguistic information.

- Bourigault (1992) reports a statistical tool, LECTEUR for extracting French terminologies. It can be used to extract French noun phrases with an 95% recall rate. However, the precision rate was not reported.

- NPtool is a detector of English noun phrases by Voutilainen (1993) NPtool extracts noun phrases and uses lexical, POS tag, and head noun information. It is also a rule-based tool.

There is no existing tool specifically for extracting noun phrases in Japanese and only one known to exist for Chinese(Li *et al.* 1995). Noun phrases can be very complex in these languages. For example, they sometimes overlap with verb phrases (Li & Thompson 1989). However, it is possible to use a simple pattern matcher to get the most simple form of noun phrases/compound nouns. Texts in Chinese or Japanese need to be tagged first, and the pattern matcher will select simple noun phrases/compound nouns from the tagged texts.

- JUMAN is a Japanese tagger and segmenter suitable for this application. As described in the previous section, JUMAN is a dictionary-based morphological analyzer.

- Li *et al.* (1995) presents a Chinese noun phrase extractor, NPext, which is a probabilistic partial parser. NPext is trained on hand-marked sentence sets. Probabilities of various POS starting or ending an NP are obtained from training. The test set of their algorithm was quite small and the recall and precision was 69.4% amd 71.3% respectively. They argue that the poor performance shows that a pure statistical approach cannot solve the complex problem of Chinese noun phrase finder, and propose to use additional linguistic rules in their future work.

Chapter 3

Previous work: Word translation from noisy parallel corpora

As seen from the description of related work in bilingual word translation algorithms, the main weakness of these approaches is lack of robustness - against structural noise in parallel corpora, and against language pairs which do not share etymological roots.

There are many bilingual texts which are translations of each other but are not translated sentence by sentence. Some texts such as the AWK manual in English and its translation in Japanese do not contain the same programming examples or postscript figures. Their online versions have many mismatched sentences and paragraphs, suggesting a discontinuous mapping between the parallel texts. Most algorithms cannot deal with deletion and insertions in parallel corpora efficiently and therefore cannot be applied to such noisy corpora. In addition, texts scanned in by optical character recognition(OCR) tools are often visually noisy, with missing sentence boundaries or additional ink marks (Church 1993). Algorithms relying on clean sentence boundary cannot be applied to OCR inputs.

Another issue is language robustness. Tools like *char_align* (Church 1993) do not rely on sentence boundaries. Instead, *char_align* aligns segments of English texts to French texts by using shared character sequences between an English word and a French word as anchor points. Such tools can be applied to any pairs of languages sharing a linguistic common root and therefore have the common cognate phenomena. However, such phenomena do not exist between, say English and Chinese, which are from different language groups and have completely different character sets.

In this chapter, we describe an algorithm we developed for bilingual lexicon acquisition which addresses the above-mentioned robustness problems simultaneously(Fung 1995). This algorithm extracts a bilingual noun and proper noun lexicon with minimal alignment from a noisy parallel corpus of English and Chinese texts(Wu 1994).

This algorithm bootstraps off of corpus alignment procedures we developed previously (Fung & Church 1994; Fung & McKeown 1994). Our previous alignment procedures attempted to align texts by finding matching word pairs and we have demonstrated their effectiveness for Chinese/English and Japanese/English. The main focus then was accurate alignment, but the procedure produced a small number of word translations as a by-product. In contrast, our new algorithm performs a rough alignment, to facilitate compiling a much larger bilingual lexicon.

The paradigm for Fung & Church (1994); Fung & McKeown (1994) is based on two main steps - find a small bilingual *primary lexicon*, use the text segments which contain some of the word pairs in the lexicon as anchor points for alignment, align the text, and compute a better *secondary lexicon* from these partially aligned texts. This paradigm can be seen as analogous to the Estimation-Maximization step in Brown *et al.* (1991); Dagan *et al.* (1993); Wu & Xia (1994).

For a noisy corpus without sentence boundaries, the primary lexicon accuracy depends on the robustness of the algorithm for finding word translations given no *a priori* information. The reliability of the anchor points will determine the accuracy of the secondary lexicon. We also want an algorithm that bypasses a long, tedious sentence or text alignment step. So the purposes of our new algorithm are to compile a reliable primary lexicon and to use minimal anchor points for secondary lexicon estimation.

3.1 A noisy parallel corpus of Chinese and English

We use parts of the HKUST English-Chinese Bilingual Corpora for our experiments on noisy-parallel corpus. This corpus consists of transcriptions of the Hong Kong Legislative Council debates in both English and Chinese (Wu 1994). The topic of these debates varies though is to some extent confined to the same domain, namely the political and social issues of Hong Kong. While the corpus contains many formal and legal phrases, it also contain many slang and idiomatic expressions which are typical of transcribed, rather than written, texts. This to some extent is a test of the algorithmic robustness since slang is usually translated non-literally. We ran the algorithm on a small part of the parallel corpus which consists of approximately 5760 unique English words.

3.2 Algorithm overview

We treat the domain word translation problem as a pattern matching problem—each word shares some common features with its counterpart in the translated text. We try to find the best representations of these features and the best ways to match them.

The outline of the algorithm is as follows:

1. **Tag the English half of the parallel text**. In the first stage of the algorithm, only English words which are tagged as nouns or proper nouns are used to match words in the Chinese text.

2. **Compute the dynamic recency vector of each word**. Each of these nouns or proper nouns is converted from their positions in the text into a vector.

3. **Match pairs of recency vectors, giving scores**. All vectors from English and Chinese are matched against each other by Dynamic Time Warping (DTW).

4. **Select a primary lexicon using the scores**. A threshold is applied to the DTW score of each pair, selecting the most correlated pairs as the first bilingual lexicon.

5. **Find anchor points using the primary lexicon**. The algorithm reconstructs the DTW paths of these positional vector pairs, giving us a set of word position points which are filtered to yield anchor points. These anchor points are used for compiling a secondary lexicon.

6. **Compute a non-linear segment binary vector for each word using the anchor points**. The remaining nouns and proper nouns in English and all words in Chinese are represented in a non-linear segment binary vector form from their positions in the text.

7. **Match binary vectors with correlation measure to yield a secondary lexicon**. These vectors are matched against each other by mutual information. A confidence score is used to threshold these pairs. We obtain the secondary bilingual lexicon from this stage.

In Section 3.4, we describe the first four stages in our algorithm, cumulating in a primary lexicon. Section 3.5 describes the next anchor point finding stage. Section 3.6 contains the procedure for compiling the secondary lexicon.

3.3 Tagging to identify nouns

Since the recency vector representation relies on the fact that words which are similar in meaning appear fairly consistently in a parallel text, this representation is best for nouns or proper nouns because these are the kind of words which have consistent translations over the entire text.

As ultimately we will be interested in finding domain-specific terms, we can concentrate our effort on those words which are nouns or proper nouns first. For this purpose, we tagged the English part of the corpus by a modified POS tagger, and apply our algorithm to find the translations for words which are tagged as nouns, plural nouns or proper nouns only. This produced a more useful list of lexicon and again improved the speed of our program.

3.4 Finding high frequency bilingual word pairs

3.4.1 Dynamic recency vectors

We treat translation as a pattern matching task where words in one language are the templates which words in the other language are matched against. The word pairs which are considered to be most similar by some measurement are taken to be translations of each other and form anchor points for the alignment. Our task, therefore, is to find a similarity measurement which can find words to serve as anchor points.

In looking at the text shown in Figure 3.1, we can see that words like *Governor* and 總督, which are translations of each other, may not necessarily occur within the same segments. To use an extreme case example, word A and its translation word B may occur in Chapters 1, 2 and 10 but they might not be in linearly corresponding byte segments. Chapters in the two parallel documents do not necessarily start or end at the same byte positions.

However, if we define arrival interval to be the difference between successive byte positions of a word in the text, then Figure 3.1 shows that the arrival intervals are very similar between word pairs. Alternatively, in our previous example, although the exact byte positions of word A in chapters 1, 2 and 10 is not similar to those of word B, the fact that they each occur 3 times first at a distance of 1 chapter and then at a distance of 8 chapters is significant. This arrival interval can be regarded as the recency information of words and can be used as another feature in the pattern matching.

More concretely, the word positions of *Governor* form a vector $\langle 2380, 2390, 2463, 2565, ... \rangle$ of length 212. We compute the recency vector for *Governor* to be $\langle 10, 73, 102, 102, 91, 923, 998, ... \rangle$ with

Figure 3.1: Part of the concordances of the word *Governor* in English and Chinese

position	arrival interval	sentence		
2380:		MISS EMILY LAU :	Governor	- - I think I would
2390:	10	would address you as	Governor	rather than Mr President
2463:	73	life will be protected .	Governor	, after reading these
2565:	102	protection of human rights ,	Governor	, I could not find
2667:	102	about other laws ,	Governor	? So will you please
2758:	91	better to address me as	Governor	rather than Mr President
3681:	923	MR PETER WONG : Mr	Governor	, you have adopted a
4679:	998	I do not think , as	Governor	of Hong Kong trying
5144:	465	IK CHI - YUEN (in Cantonese) : Mr	Governor	, Meeting Point welcome
5439:	295	proposals because the	Governor	of Hong Kong and
90:		席者 :	總督	彭 定 康 先
2021:	1931	定 和 以往 歷任	總督	與 本 局 所
2150:	129	問 (譯 文) :	總督	先生 (我 想
2158:	8	(我 想 稱呼 你 為	總督	先生 較
2238:	80	的 社會 。	總督	先生 , 讀 畢
2329:	91	滿 。 同 時 ,	總督	先生 , 在 保
2416:	87	歡迎 。 但	總督	先生 , 其他
2490:	74	未來 ?	總督	答 (譯 文) :
2510:	20	為 你 稱呼 我 為	總督	先生 較 主
3238:	728	這 是 前任	總督	領導 下 的

length 211. The position vector of 總督 is $\langle 90, 2021, 2150, 2158, 2238, ...\rangle$ of length 254. Its recency vector is $\langle 1931, 129, 8, 80, 91, 87, 74, 20, 728, ...\rangle$ with 253 as its length. The recency vectors of these two words have different lengths because their frequencies are different.

We can compute the recency vector for any word in the bilingual texts in this way. The values of the vector are integers with lower bound of 1, and upper bound being the length of the text.

Figuratively, for each word, we have a signal such as is shown in Figure 3.2 with the horizontal axis the word position in the text and the vertical axis the values of the recency vectors. Note the striking similarity of the two signals which represent *Governor* in English and Chinese. The word *Bill* in Chinese is not a translation of *Governor* and its signal is clearly uncorrelated with that of *Governor*. The signal for *President* is also very different. These signal patterns of the words can be used as the features for matching to give us most similar pairs.

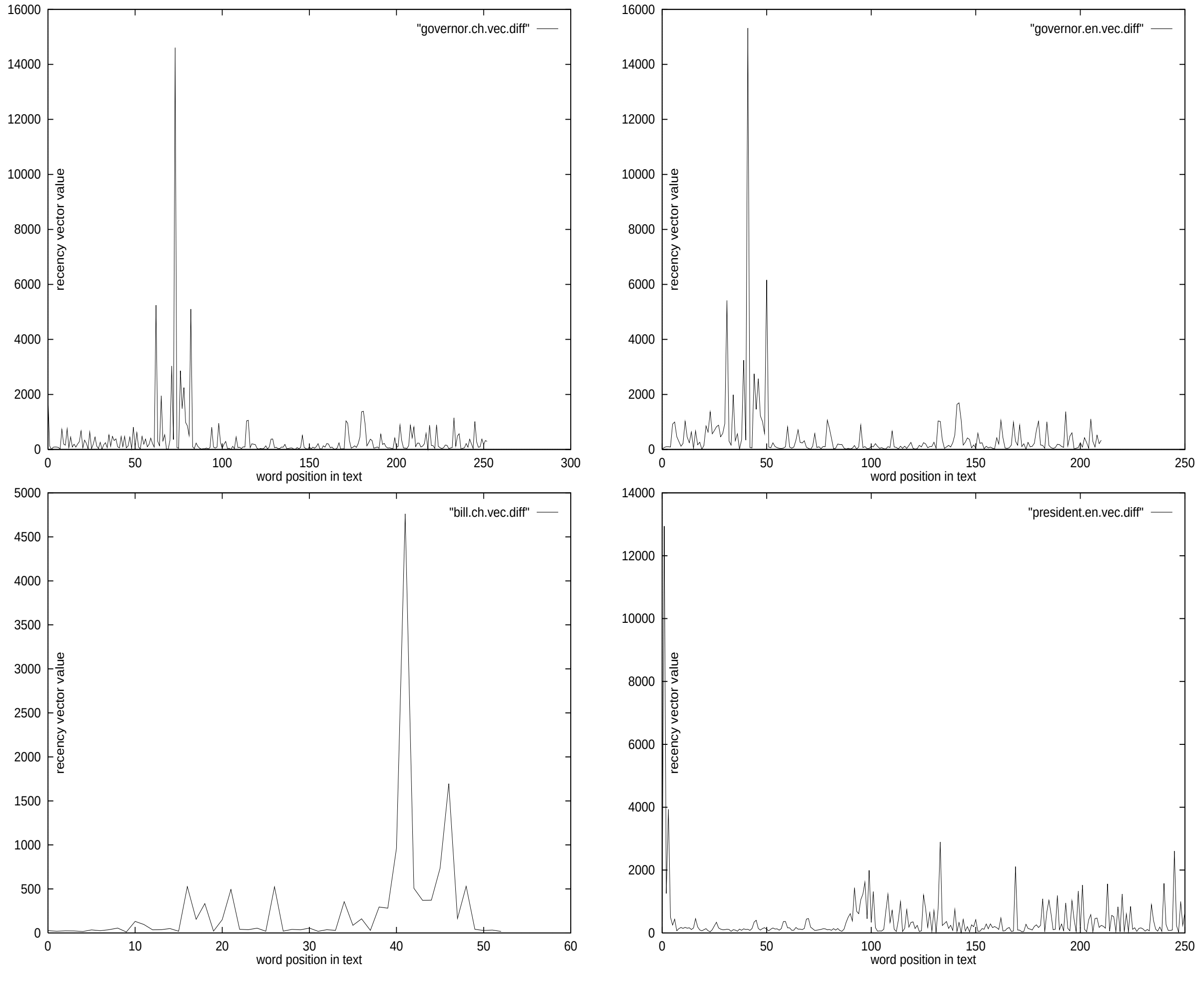


Figure 3.2: DK-vec signals showing similarity between *Governor* in English and Chinese, contrasting with *Bill* and *President* in English

Looking at the values of individual recency vectors, it is hard for us to tell which pairs are indeed similar to each other. However, the signals show that there is a characteristic shape of each vector which can be used for pattern matching. Just as it is hard for people tell if the spectral signals of two utterances represent the same word, so it is hard for humans to match the patterns of the recency vectors. But pattern matching has been successfully used in signal processing to match the spectral patterns for two words; we use a similar techinque to "recognize" translation of words.

Thus, DK-vec assumes that if two words are translation of each other, they are more likely to occur similar number of times at similar arrival intervals. Our task is thus to determine what distance metric to use to measure the arrival intervals and do pattern matching.

3.4.2 Matching recency vectors

Many pattern recognition techniques could be used to compare vectors of variable lengths in one language against vectors in the other language. We propose using Dynamic Time Warping (DTW), which has been used extensively for matching spectral signals in speech recognition tasks.

Consider the two vectors for *Governor*/總督, vector1 of length 212 and vector2 of length 254. If we plot vector2 against vector1, we get a trellis of length 212 and height 254. If the two texts were perfectly aligned translations of each other, then the interword arrival differences would be linearly proportional to the slope of the trellis, so that we could draw a straight line from the origin to the point (212, 254) to represent this correspondence between the two vectors. Since they are not, however, we need to measure the distortion of the one vector relative to the other. The way we do this is by incrementally computing the optimal path from the origin to point (212,254) as close as possible to the idealized diagonal. The divergence from the ideal gives a measure of the distortion. DTW traces the correspondences between all points in $V1$ and $V2$ (with no penalty for deletions or insertions). Moreover, DTW paths can be reconstructed during a backtracking step. The algorithm can then automatically choose the best points on these DTW paths as anchor points.

Our DTW algorithm is as follows:

- **1 Initialization**

$$\begin{aligned}
\varphi_1(1,1) &= \zeta(1,1) \\
\varphi_1(i,1) &= \zeta(i,1) + \varphi(i-1,1]) \\
\varphi_1(1,j) &= \zeta(1,j) + \varphi(1,j-1) \\
\text{where } \varphi(a,b) &= \text{minimum cost of moving} \\
& \quad \text{from } a \text{ to } b \\
\zeta(c,d) &= |V1[c] - V2[d]| \\
\text{for } i &= 1,2,\ldots,N \\
j &= 1,2,\ldots,M \\
N &= \dim(V1) \\
M &= \dim(V2)
\end{aligned}$$

- **2 Recursion**

$$
\begin{aligned}
\varphi_{n+1}(i,m) &= \min_{1\le l\le 3}[\zeta(l,m)+\varphi_n(i,l)] \\
\xi_{n+1}(m) &= \operatorname*{argmin}_{1\le l\le 3}[\zeta(l,m)+\varphi_n(i,l)] \\
\text{for } n &= 1,2,\ldots,N-2 \\
\text{and } m &= 1,2,\ldots,M
\end{aligned}
$$

- **3 Termination**

$$
\begin{aligned}
\varphi_N(i,j) &= \min_{1\le l\le 3}[\zeta(l,m)+\varphi_{N-1}(i,l)] \\
\xi_N(j) &= \operatorname*{argmin}_{1\le l\le 3}[\zeta(l,m)+\varphi_{N-1}(i,j)]
\end{aligned}
$$

- **4 Path reconstruction**

  In our algorithm, we reconstruct the DTW path and obtain the points on the path for later use. The DTW path for *Governor*/總督 is as shown in Figure 3.3.

$$
\begin{aligned}
\text{optimal path} &= (i, i_1, i_2, \ldots, i_{m-2}, j) \\
\text{where } i_n &= \zeta_{n+1}(i_{n+1}), \\
& \quad n = N-1, N-2, \ldots, 1 \\
\text{with } i_N &= j
\end{aligned}
$$

In step (2), to choose among the three directions, a distortion or distance function is computed. So to decide whether the path at (i,j) should go to (i+1,j), (i,j+1) or (i+1, j+1), we compute the absolute difference between V1[i+1] and V2[j], V1[i] and V2[j+1], and between V1[i+1] and V2[j+1]. The direction is determined by the smallest absolute difference among the three pairs.

As the path is constructed, the absolute differences at each step are accumulated into a running score which is finalized when the path reaches the point (212, 254). This is the score for the pair of vectors we compared. For every word vector in language A, the word vector in language B which gives it the highest DTW score ($D(X[i:N], Y[i:N])$) is taken to be its translation.

We thresholded the bilingual word pairs obtained from above stages in the algorithm and stored the more reliable pairs as our primary bilingual lexicon.

3.4.3 Statistical filters

If we have to exhaustively match all English nouns and proper nouns against all Chinese words in the corpus, the matching would be very expensive since it involves computing all possible paths between two vectors, and then backtracking to find the optimal path, and doing this for all English/Chinese word pairs in the texts. The complexity of DTW is $\Theta(NM)$ and the complexity of the matching is $\Theta(IJNM)$ where I is the number of nouns and proper nouns in the English text, J is the number of unique words in the Chinese text, N is the occurrence count of one English word and M the occurrence count of one Chinese word.

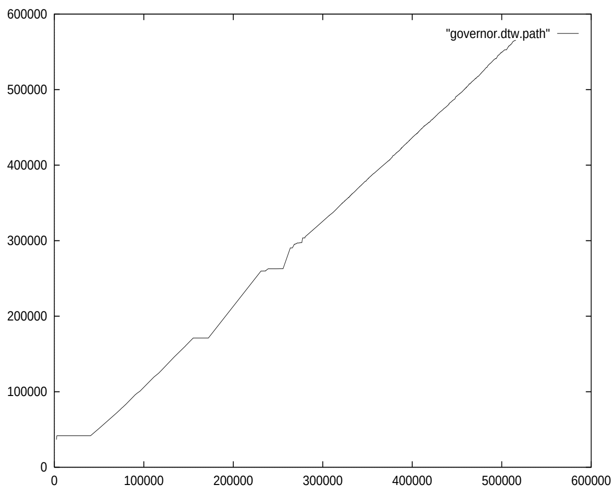


Figure 3.3: Dynamic Time Warping path for *Governor* in English and Chinese

- A starting point constraint - the value of the first dimension in each vector is its starting word position (minus zero). So vectors whose first value is at least half the text length apart are not considered in DTW since it means they start to occur at least half a text apart and cannot be translations of each other. This is a position constraint.
- Length constraint - the length of a vector is actually the frequency of that word minus 1. So vectors whose lengths vary by at least half the length of the text represent cases where one word occurs at least twice as often as the other word; thus they cannot be translations of each other. They are not considered for DTW either. This is a frequency constraint.
- To improve the computation speed, we constrain the vector pairs further by looking at the Euclidean distance $\mathcal{E}$ of their means and standard deviations:

$$\mathcal{E} = \sqrt{(m_1 - m_2)^2 + (\sigma_1 - \sigma_2)^2}$$

  If their Euclidean distance is higher than a certain threshold, we filter the pair out and do not use DTW matching on them. This process eliminated most word pairs. Note that this Euclidean distance function helps to filter out word pairs which are very different from each other, but it is not discriminative enough to pick out the best translation of a word. So for word pairs whose Euclidean distance is below the threshold, we still need to use DTW matching to find the best translation. However, this Euclidean distance filtering greatly improved the speed of this stage of bilingual lexicon compilation.

3.5 Finding anchor points and eliminating noise

Since the primary lexicon after thresholding is relatively small, we would like to compute a secondary lexicon including some words which were not found by DTW. At stage 5 of our algorithm, we try to find anchor points on the DTW paths which divide the texts into multiple aligned segments for compiling the secondary lexicon. We believe these anchor points are more reliable than those obtained by tracing all the words in the texts.

For every word pair from this lexicon, we had obtained a DTW score and a DTW path. If we plot the points on the DTW paths of all word pairs from the lexicon, we get a graph as in the left hand side of Figure 3.4. Each point (i, j) on this graph is on the DTW $path(v1, v2)$ where $v1$ is from English words in the lexicon and $v2$ is from the Chinese words in the lexicon. The union effect of all these DTW paths shows a salient line approximating the diagonal. This line can be thought of the text alignment path. Its departure from the diagonal illustrates that the texts of this corpus are not identical nor linearly aligned.

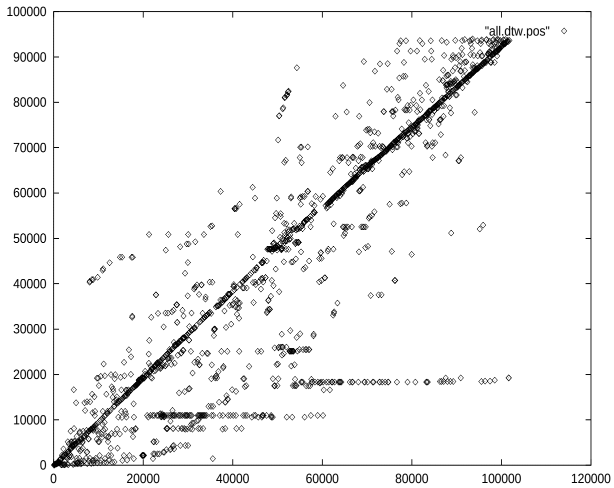
There is some noise in this graph due to the rough lexical alignment in the first stage. Previous alignment methods we used such as Church (1993); Fung & Church (1994); Fung & McKeown (1994) would bin the anchor points into continuous blocks for a rough alignment. This would have a smoothing effect. However, we later found that these blocks of anchor points are not precise enough for our Chinese/English corpus. We found that it is more advantageous to increase the overall reliability of anchor points by keeping the highly reliable points and discarding the rest.

From all the points on the union of the DTW paths, we filter out the points by the following conditions: If the point (i, j) satisfies

$$\begin{array}{ll} \textit{(slope constraint)} & j/i > 600 * N[0] \\ \textit{(window size constraint)} & i >= 25 + i_{previous} \\ \textit{(continuity constraint)} & j >= j_{previous} \\ \textit{(offset constraint)} & j - j_{previous} > 500 \end{array}$$

then the point (i, j) is noise and is discarded.

After filtering, we get points such as shown in the right hand side of Figure 3.4. There are 388 highly reliable anchor points. They divide the texts into 388 segments. The total length of the texts is around 100000, so each segment has an average window size of 257 words which is considerably longer than a sentence length; thus this is a much rougher alignment than sentence alignment, but nonetheless we still get a bilingual lexicon out of it.



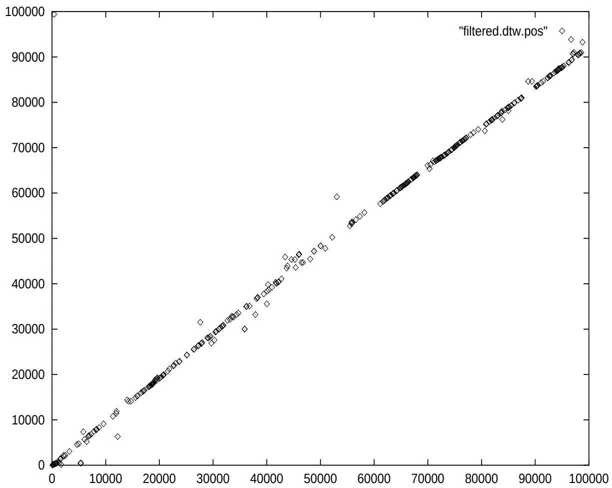


Figure 3.4: DTW path reconstruction output and the anchor points obtained after filtering

The constants in the above conditions are chosen roughly in proportion to the corpus size so that the filtered picture looks close to a clean, diagonal line. This ensures that our development stage is still unsupervised. We would like to emphasize that if they were chosen by looking at the lexicon output as would be in a supervised training scenario, then one should evaluate the output on an independent test corpus.

Note that if one chunk of noisy data appeared in text1 but not in text2, this part would be segmented between two anchor points (i, j) and (u, v). We know point i is matched to point j, and point u to point v, the texts between these two points are matched but we do not make any assumption about how this segment of texts are matched. In the extreme case where $i = u$, we know that the text between j and v is noise. We have at this point a segment-aligned parallel corpus with noise elimination.

3.6 Finding low frequency bilingual word pairs

Many nouns and proper nouns were not translated in the previous stages of our algorithm. They were not in the first lexicon because their frequencies were too low to be well represented by recency vectors.

3.6.1 Non-linear segment binary vectors

In stage 6, we represent the positional and frequency information of low frequency words by a binary vector for fast matching. We developed binary vector representation previously for alignment(Fung & Church 1994). In a parallel corpus, texts in both languages are divided into equal number of segments by a set of segment delimiters. A binary vector for a certain English word has its *ith* bit set to one if that word appears in the *ith* segment, zero otherwise. For every given *pair* of English and Chinese words to be matched, another co-occurrence binary vector is computed where the i-th bit is set to one if *both* words are found in the i-th segment.

The binary vector notation can be illustrated by the following contingency tables using the English word *prosperity* and its Chinese translation 繁榮 as an example. In the table, (a) represents the number of segments where both English and Chinese word were found, (b) shows the number of segments where just the English word was found, (c) is the number of segments where just the Chinese word was found, and (d) is the number of segments where neither word was found:

Figure 3.5: A contingency matrix

	Chinese	
English	a	b
	c	d

In general, if the English and Chinese words are good translations of one another, then a should be large, and b and c should be small. In contrast, if the two words are not good translations of one another, then a should be small, and b and c should be large.

The segment delimiters in our algorithm were set by the anchor points obtained from previous stage 5. The 388 anchor points $(95, 10)$, $(139, 131), \ldots, (98809, 93251)$ divide the two texts into 388 non-linear segments. Text1 is segmented by the points $(95, 139, \ldots, 98586, 98809)$ and text2 is segmented by the points $(10, 131, \ldots, 90957, 93251)$.

For the nouns we are interested in finding the translations for, we again look at the position vectors. For example, the word *prosperity* occurred seven times in the English text. Its position vector is $\langle 2178, 5322, \ldots, 86521, 95341\rangle$. We convert this position vector into a binary vector $V1$ of 388 dimensions where $V1[i] = 1$ if *prosperity* occured within the *ith* segment, $V1[i] = 0$ otherwise. For *prosperity*, $V1[i] = 1$ where $i = 20, 27, 41, 47, 193, 321, 360$. The Chinese translation for prosperity is 繁榮. Its position vector is $\langle 1955, 5050, \ldots, 88048\rangle$. Its binary vector is $V2[i] = 1$ where $i = 14, 29, 41, 47, 193, 275, 321, 360$. We can see that these two vectors share five segments in common. Their contingency table is as follows:

Figure 3.6: *prosperity* in English and Chinese

	繁榮	
prosperity	5	2
	3	378

We compute the segment vector for all English nouns and proper nouns not found in the first lexicon and whose frequency is above two. Words occurring only once are extremely hard to translate although our algorithm was able to find some pairs which occurred only once.

3.6.2 Binary vector correlation measure

To match these binary vectors $V1$ with their counterparts in Chinese $V2$, we use a mutual information score m to describe the probability of finding both words in the same segment.

The occurrence probability of $V1$ is $\Pr(V1) = \frac{b}{a+b+c+d}$. The probability of $V2$ is $\Pr(V2) = \frac{c}{a+b+c+d}$. The joint probability, assuming $V1$ and $V2$ are independent of each other, is $\Pr(V1, V2) = \frac{a}{a+b+c+d}$. We use a correlation score, the mutual information score:

$$\begin{aligned} m &= \log_2 \frac{\Pr(V1, V2)}{\Pr(V1)\Pr(V2)} \\ &= \log_2 \frac{a * (a + b + c + d)}{b * c} \end{aligned}$$

If *prosperity* and 繁榮 occurred in the same eight segments, their mutual information score would be 5.6. If they never occur in the same segments, their m would be negative infinity. Here, for *prosperity*/繁榮, $m = 5.077$ which shows that these two words are indeed highly correlated.

Unfortunately, mutual information is often unreliable when the counts are small. Such is the case when we look for infrequent words. If we pick a pair of these words at random, there is a very large chance that they would receive a large mutual information value by chance. For example, let e be an English word that appeared just once and let c be a French word that appeared just once. Then, there is a non-trivial chance ($\frac{1}{K}$) that e and f will appear in the same piece. If this should happen, the mutual information estimate would be very large, i.e., $\log K$, and probably misleading.

In order to avoid this problem, we use a t-score to filter out insignificant mutual information values. We keep pairs of words if their $t > 1.65$ where

$$t \approx \frac{\Pr(V1, V2) - \Pr(V1)\Pr(V2)}{\sqrt{\frac{1}{a+b+c+d}\Pr(V1, V2)}}$$

Note that t score is low when $a+b+c+d$ is low. So the segment numbers need to be significant for the co-occurrence measure to be reliable. For *prosperity*/繁榮, $t = 2.33$ which shows that their correlation is indeed reliable.

3.7 Results

The English half of the corpus has 5760 unique words containing 2779 nouns and proper nouns. Most of these words occurred only once. We carried out two sets of evaluations, first counting only the best matched pairs, then counting top three Chinese translations for an English word. The top N candidate evaluation is useful because in a machine-aided translation system, we could propose a list of up to, say, ten candidate translations to help the translator. We obtained the evaluations of three human judges (E1-E3). Evaluator E1 is a native Cantonese speaker, E2 a Mandarin speaker, and E3 a speaker of both languages. The results are shown in Figure 3.7.

lexicons	total word pairs	correct pairs			accuracy		
		E1	E2	E3	E1	E2	E3
primary(1)	128	101	107	90	78.9%	83.6%	70.3%
secondary(1)	533	352	388	382	66.0%	72.8%	71.7%
total(1)	661	453	495	472	68.5%	74.9%	71.4%
primary(3)	128	112	101	99	87.5%	78.9%	77.3%
secondary(3)	533	401	368	398	75.2%	69.0%	74.7%
total(3)	661	513	469	497	77.6%	71.0%	75.2%

Figure 3.7: Bilingual lexicon compilation results

The average accuracy for all evaluators for both sets is 73.1%. We found that many of the mistaken translations resulted from insufficient data suggesting that we should use a larger size corpus in our future work. Tagging errors also caused some translation mistakes. English words with multiple senses also tend to be wrongly translated at least in part (e.g., *means*). There is no difference between capital letters and small letters in Chinese, and no difference between singular and plural forms of the same term. This also led to some error in the vector representation. The evaluators' knowledge of the language and familiarity with the domain also influenced the results.

Apart from single word to single word translation such as *Governor*/總督 and *prosperity*/繁榮 , we also found many single word translations which show potential towards being translated as compound domain-specific terms such as follows:

- **finding Chinese words**: Chinese texts do not have word boundaries such as space in English, therefore our text was tokenized into words by a statistical Chinese tokenizer (Fung & Wu 1994). Tokenizer error caused some Chinese characters to be not grouped together as one word. Our program located some of these words. For example, *Green* was aligned to 綠, 皮 and 書 which suggests that 綠皮書 could be a single Chinese word. It indeed is the name for Green Paper – a government document.

- **compound noun translations**: *carbon* could be translated as 碳, and *monoxide* as 一氧化. If *carbon monoxide* were translated separately, we would get 碳 一氧化 . However, our algorithm found both *carbon* and *monoxide* to be most likely translated to the single Chinese word 一氧化碳 which is the correct translation for *carbon monoxide*.

The words *Legislative* and *Council* were both matched to 立法 and similarly we can deduce that Legislative Council is a compound noun/collocation. The interesting fact here is, *Council* is also matched to 局. So we can deduce that 立法局 should be a single Chinese word corresponding to *Legislative Council*.

- **slang**: Some word pairs seem unlikely to be translations of each other, such as *collusion* and its first three candidates 扯(*pull*), 貓(*cat*), 尾(*tail*). Actually *pulling the cat's tail* is Cantonese slang for *collusion*.

  The word *gweilo* is not a conventional English word and cannot be found in any dictionary but it appeared eleven times in the text. It was matched to the Cantonese characters 俗, 稱, 鬼, and 佬 which separately mean *vulgar/folk*, *name/title*, *ghost* and *male*. 俗稱鬼佬 means *the colloquial term gweilo*. *Gweilo* in Cantonese is actually an idiom referring to a male westerner that originally had pejorative implications. This word reflects a certain cultural context and cannot be simply replaced by a word to word translation.

- **collocations**: Some word pairs such as *projects* and 房屋(*houses*) are not direct translations. However, they are found to be constituent words of collocations – the *Housing Projects* (by the Hong Kong Government).Both *Cross* and *Harbour* are translated to 海底(*sea bottom*), and then to 隧道(*tunnel*), not a very literal translation. Yet, the correct translation for 海底隧道 is indeed *the Cross Harbor Tunnel* and not *the Sea Bottom Tunnel*.

  The words *Hong* and *Kong* are both translated into 香港, indicating *Hong Kong* is a compound name.

  Basic and *Law* are both matched to 基本法, so we know the correct translation for 基本法 is *Basic Law* which is a compound noun.

- **proper names** In Hong Kong, there is a specific system for the transliteration of Chinese family names into English. Our algorithm found a handful of these such as *Fung*/馮, *Wong*/黃, *Poon*/潘, *Hui*/ *Lam*/林, *Tam*/譚, etc.

3.8 Discussion

Dynamic word features used in this algorithm capture the frequency, position and recency information of words in noisy parallel texts. These features are more robust than conventional co-occurrence word features in that they can be obtained from mismatching sentences and segments in noisy parallel corpora. Since our algorithm does not assume sentence boundaries, it can be applied to OCR input. Because of these properties, the choice of source and target input languages can be arbitrary and no amount of cleaning or determination of sentence boundaries in a corpus is necessary.

In addition, our algorithm bypasses the sentence alignment step to find a bilingual lexicon of nouns and proper nouns. It has shown effectiveness in compiling a bilingual word lexicon from texts with no sentence boundary information and with noise; fine-grain sentence alignment is not necessary for lexicon compilation as long as we have highly reliable anchor points. Compared to other word alignment algorithms, it does not need *a priori* information. Since EM-based word alignment algorithms using random initialization can fall into local maxima, our output can also be used to provide a better initializing basis for EM methods.

Its output shows promise for translating domain-specific, technical and regional compounds terms. We need to experiment on how to extend the algorithm for this purpose.

Chapter 4

Previous work: Word translation from non-parallel corpora

Although bilingual parallel corpora have been available in recent years, they are still relatively few in comparison to the large amount of monolingual text. Acquiring and processing of parallel corpora are usually labour-intensive and time-consuming. It is also unlikely that one can find parallel corpora in any given domain, in any given pair of languages. If a machine translation system, or a translator needs the help of a domain-specific term translation, it could be a time consuming and probably fruitless effort to look for translated parallel texts in that domain in those languages. However, given any domain, it is very possible that there are texts about that domain in any given language. The translation process would not be constrained by the texts available. More importantly, the existence of a parallel corpus in a particular domain means *some* translator has translated it, therefore, the bilingual lexicon compiled from such a corpus is at best a reverse engineering of the lexicon this translator used. This is called *re-translation*. On the other hand, if we can compile a dictionary of domain-specific words from non-parallel corpora of monolingual texts, the results would be much more useful. Moreover, a language-independent algorithm for non-parallel corpopra could be run on *multiple* languages in the same domain at the same time. For example, if domain term translation is needed for English into Chinese, Japanese, and French, three sets of parallel corpora consisting of six texts in the same domain would be needed; whereas only four texts are needed for non-parallel corpora translation algorithm [1].

In the following sections, we describe our various algorithms in extracting word correspondence information from non-parallel texts in order to compile bilingual lexicon.

4.1 A non-parallel corpus of Chinese and English

We again select different parts of the HKUST English-Chinese Bilingual Corpora for our experiments on non-parallel corpus. We use the data from 1988-1992, taking the first 73618 sentences from the years 1988-90 in the English text, and the next 73618 sentences from the years 1990-92 in the Chinese text. There are no overlapping sentences between the texts. The corpus contains many micro-domains. Various topics discussed during the debates include government-sponsored

[1] parallel texts of multiple languages do exist, such as the United Nations document in its five official languages, but they are limited to a few domains.

projects such as the Public Housing Project, the new Airport Core Project, various public transportation issues, income tax, education budgets, etc, etc. This might be a less than ideal choice of a domain specific corpus. But we are using it at the moment for testing purposes.

Note that although we select the same number of sentences from each language, there are 22147 unique words from English, and only 7942 unique words from Chinese. This demonstrates the non-parallelness of the corpus.

4.2 Context heterogeneity

In this section, we describe a novel **context heterogeneity** similarity measure between words and their translations in helping to compile bilingual lexicon entries from a non-parallel English-Chinese corpus. Current algorithms for bilingual lexicon compilation rely on occurrence frequencies, length or positional statistics derived from parallel texts. There is little correlation between such statistics of a word and its translation in non-parallel corpora. On the other hand, we suggest that words with productive context in one language translate to words with productive context in another language, and words with rigid context translate into words with rigid context. Context heterogeneity measures how productive the context of a word is in a given domain, independent of its absolute occurrence frequency in the text. Based on this information, we derive statistics of bilingual word pairs from a non-parallel corpus. These statistics can be used to bootstrap a bilingual dictionary compilation algorithm.

As demonstrated in all the bilingual lexicon compilation algorithms, the foremost task is to identify word features which are similar between a word and its translation, yet different between a word and other words which are not its translations. In parallel corpora, this feature could be the positional co-occurrence of a word and its translation in the other language in the same sentences (Kupiec 1993; Smadja & McKeown 1993; Dagan *et al.* 1993; Wu & Xia 1994) or in the same segments (Fung & Church 1994; Fung 1995). In a non-parallel corpus, there is no corresponding sentence or segment pairs, so the co-occurrence feature is not applicable. In Fung & McKeown (1994); Fung (1995), the word feature used was the recency vector. Whereas this is more robust than sentence co-occurrence features, the matching between two positional difference vectors presumes the two texts are rough translations of one another. Moreover, whereas the occurrence frequency of a word and that of its translation are relatively similar in a parallel corpus, they have little correlation in non-parallel texts. Our task is, therefore, to identify a word feature correlating a pair of words even if they appear in texts which are not translations of each other. This feature should also be language and character set independent, i.e. it should be applicable to pairs of languages very different from each other. We propose that **context heterogeneity** is such a feature.

4.2.1 Context heterogeneity of a word

In a non-parallel corpus, a domain-specific term and its translation are used in different sentences in the two texts. Take the example of the word *air* in the English text. Its concordance is shown partly in Figure 4.1. It occurred 176 times. Its translation 空氣 occurred 37 times in the Chinese text and part of its concordance is shown in Figure 4.2. They are used in totally different sentences. Thus, we cannot hope that their occurrence frequencies would correspond to each other in any significant way.

On the other hand, *air*/空氣 are domain-specific words in the text, meaning something we breathe, as opposed to of some kind of ambiance or attitude. They are used *mostly* in similar *contexts*, as shown in the concordances. If we look at the content word preceding *air* in the concordance, and the content word following it, we notice that *air* is not randomly paired with other words. There are a limited number of word bigrams (x, W) and a limited number of word bigrams (W, y) where W is the word *air*; likewise for 空氣. The number of such unique bigrams indicate a degree of heterogeneity of this word in a text in terms of its neighbors.

We define the context heterogeneity vector of a word W to be an ordered pair[2] (x, y) where:

$$
\begin{aligned}
\text{left heterogeneity } x &= \frac{a}{c}; \\
\text{right heterogeneity } y &= \frac{b}{c}; \\
a &= \text{number of different types of tokens} \\
&\quad\ \text{immediately preceding W in the text;} \\
b &= \text{number of different types of tokens} \\
&\quad\ \text{immediately following W in the text;} \\
c &= \text{number of occurrences of W in the text;}
\end{aligned}
$$

The context heterogeneity of any function word, such as *the*, would have x and y values very close to one, since it can be preceded or followed by many different words. On the other hand, the x value of the word *am* is small because it always follows the word *I*.

We postulate that the context heterogeneity of a given domain-specific word is more similar to that of its translation in another language than that of an unrelated word in the other language, and that this is a more salient feature than their occurrence frequencies in the two texts.

For example, the context heterogeneity of *air* is $(119/176, 47/176) = (0.676, 0.267)$ and the context heterogeneity of its translation in Chinese, 空氣 is $(29/37, 17/37) = (0.784, 0.459)$. The context heterogeneity of the word 休會/*adjournment*, on the other hand, is $(37/175, 16/175) = (0.211, 0.091)$. Notice that although *air* and 休會 have similar occurrence frequencies, their context heterogeneities have very different values, indicating that *air* has much more productive context than 休會. On the other hand, 空氣 has more similar context heterogeneity values as those of *air* even though its occurrence frequency in the Chinese text is much lower.

4.2.2 Distance measure between two context heterogeneity vectors

To measure the similarity between two context heterogeneity vectors, we use simple Euclidean distance $\mathcal{E}$ where :

$$\mathcal{E} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

.

The Euclidean distance between *air* and 空氣 is 0.2205 whereas the distance between *air* and 休會 is 0.497. We use the ordered pair based on the assumption that the word order for nouns in English and Chinese are similar most of the times. For example, *air pollution* is translated into 空氣污染.

[2]The ordered pair can be extended into ordered n tuples

Figure 4.1: Part of the concordance for *air*

Word position in text 1	concordance	
8754	people to enjoy fresh	air , exercise , and a complete change of
14329	, is it possible for room	air - conditioners to be provided
14431	houses and institutions . I believe that	air - conditioners
20294	Chicago Expo told people all about	air - conditioning and the 1 9 3 9 Expo in
31780	likely to be attracted to visit Expo by	air would only aggravate the problem .
86604	overnment needs to come out of its old	air - tight armour suit which might serve
102837	the problems of refuse , sewage , polluted	air , noise and chemical
118017	ociety marching parallel with decline our	air and water and general
118113	. It will cover whole spectrum pollution :	air , noise , water and wastes.
119421	KMB is now experimenting with	air - conditioned double - deckers

Figure 4.2: Part of the concordance for *air* in Chinese

Word position in text 2	concordance	
32978	上 沒有 免費 東西 , 即使 我 們 呼 吸	空氣 , 由於 需要 解決 污染 問題 , 也 絕 非 免費 的
65488	減低 了 燃 油 含 硫 量 , 從而 大大 提高	空氣 質素 。 這項 措施 旨在 解決 影響 民 居 最
153687	下列 各項 新 措施 : (a) 推 出 兩 條 新	空氣 調節 特 快 巴士 線 , 來 往 九龍
202338	及 公布 有關 本港 使用 無 鉛 汽 油 後	空氣 含 苯 量 資料 , 以及 會否 採取 管制 措施 ,
202594	環境保護署 目 前 正 進行 測 量 ” 周 圍	空氣 每月 含 苯 量 ” , 作為 空氣污染 監察 程序
240355	一些 令人 鼓舞 成績 : — 大大 減輕 了	空氣 污染 程度 ; — 在 實施 新 廢物
261651	電 工程 師 設計 輸 電 管 , 排 水 , 通 風 及	空氣 調節 等 系統 。 完成 此等 工程
284517	服 務 建議 。 我 提出 建議 包括 改善	空氣 調節 系統 , 以及 與 小 輪 公司 加強 合作 , 推
284547	鼓勵 乘客 使用 渡輪 和 輕鐵 服 務 。 (1)	空氣 調節 不足 , 尤 其 是 在 夏 天
293127	國際間 所 採用 規定 來 立例 規定	空氣 中 危險 化 學 品 含 量 標準 ?

4.2.3 Filtering out function words in English

There are many function words in English which do not translate into Chinese. This is because in most Asian languages, there are very few function words compared to Indo-European languages. Function words in Chinese or Japanese are frequently omitted. This partly contributes to the fact that there are far fewer Chinese words than English words in two texts of similar lengths.

Since these functions words such as *the*, *a*, *of* will affect the context heterogeneity of most nouns in English while giving very little information, we filter them out from the English text. This heuristic greatly increased the context heterogeneity values of many nouns. The list of function words filtered out are *the*, *a*, *an*, *this*, *that*, *of*, *by*, *for*, *in*, *to*. This is by no means a complete list of English function words. More vigorous statistical training methods could probably be developed to find out which function words in English have no Chinese correspondences. However, if one uses context heterogeneity in languages having more function words such as French, it is advisable that filtering be carried out on both texts.

4.2.4 Experiment 1: Finding word translation candidates

Given the simplicity of our current context heterogeneity measures and the complexity of finding translations from a non-parallel text in which many words will not find their translations, we propose to use context heterogeneity only as a bootstrapping feature in finding a candidate list of translations for a word.

In our first experiment, we hand-compiled a list of 58 word pairs as in Figures 4.4 and 4.5 in English and Chinese, and then used 58 by 58 context heterogeneity measures to match them against each other. Note that this list consists of many single character words which have ambiguities in Chinese, English words which should have been part of a compound word, multiple translations of a single word in English, etc. The initial results are revealing as shown by the histograms in Figure 4.3.

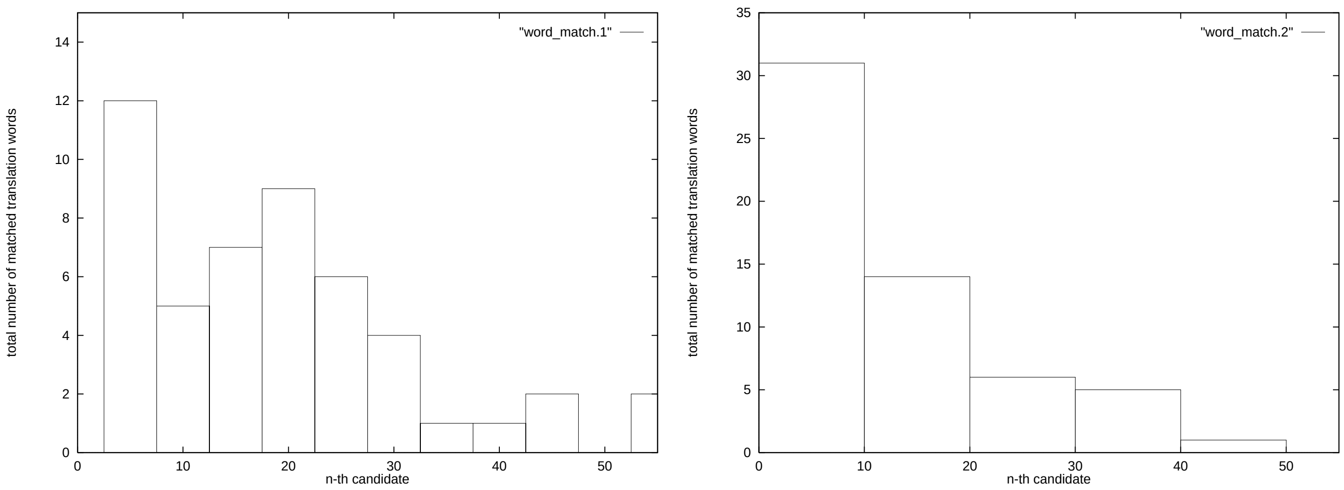


Figure 4.3: Results of word matching using context heterogeneity

In the left figure, we show that 12 words have their translations among the top 5 candidates in the first column, 5 words have their translations among the top 10 candidates, and so on.

In the right figure, we show the result of filtering out the Chinese genitive 的 from the Chinese texts. In this case, we can see that over 50% of the words found their translation in the top 10 candidates, although fewer words found their translations among the top 5 candidates. These histograms show that *most* words, by the context heterogeneity measure, are *more* correlated with their translations than average.

In Sections 4.2.4 to 4.2.4, we will discuss the effects of various factors on our results.

Figure 4.4: Test set words - part one

English word	Chinese word	possible Chinese POS
Basic	基本法	noun
British	英國	noun-adj
CHIM	詹	ambiguous
CHOW	周	ambiguous
CHOW	淑	ambiguous
China	中國	noun-adj
Committee	委員會	noun
Council	局	ambiguous
Declaration	聲明	noun-verb
Financial	財政	noun-adj
Government	政府	noun-adj
Governor	總督	noun
Hong	香港	proper noun
Kong	香港	proper noun
LAM	林	ambiguous
LAU	劉	proper noun
Law	基本法	noun
Ltd	有限公司	noun
McGREGOR	覺	ambiguous
Mr	議員	noun
October	十月	noun
SECURITY	保安	noun-verb
Second	二讀	noun
TAM	譚	proper noun
TU	杜	ambiguous
WONG	黃	ambiguous
YIU	耀	ambiguous

Effect of Chinese tokenization

We used a statistically augmented Chinese tokenizer for finding word boundaries in the Chinese text (Fung & Wu 1994; Wu & Fung 1994). Chinese tokenization is a difficult problem and tokenizers always have errors. Most single Chinese characters can be joined with other character(s) to form different words. So the translation of a single Chinese character is ill-defined. Moreover,

Figure 4.5: Test set words - part two

English word	Chinese word	possible Chinese POS
address	施政報告	noun
air	空氣	noun
colleagues	同事	noun
debate	辯論	noun-verb
decisions	領導	noun-verb
development	發展	noun-verb
employers	僱主	noun
employment	僱主	noun
expenditure	開支	noun-verb
figures	數字	noun
growth	增長	noun-verb
incidents	事件	noun
land	公頃	quantifier
land	土地	noun
laws	法例	noun
majority	大多數	noun-adj
proposals	建議	noun-verb
prosperity	繁榮	noun-adj
quality	素	ambiguous
rate	率	ambiguous
relationship	關係	noun
rights	人權(human rights)	noun
risk	險	ambiguous
safety	安全	noun-adj
services	服務	noun-verb
simple	簡單	adj
step	步	ambiguous
targets	目標	noun
tunnels	隧道	noun
vessels	船隻	noun
welfare	社會福利	noun
yesterday	昨天	noun

in some cases, our Chinese tokenizer groups frequently co-occurring characters into a single word that does not have independent semantic meanings. For example, 條第/*-th item, number*. In the above cases, the context heterogeneity values of the Chinese translation is not reliable. However, translators would recognize this error readily and would not consider it as a translation candidate.

Effect of English compound words

As we have mentioned, our Chinese text has many acronyms and idioms which were identified by our tokenizer and grouped into a single word. However, the English text did not under go a collocation extraction process. Therefore, there are far more English words than Chinese words. There is also a mismatching between English words and Chinese phrases/words. For example, *Cross Harbour Tunnel* is counted three in English, but as one in Chinese - 海底隧道. Since the capitalized *Harbour* is always surrounded by *Cross ... Tunnel*, its context heterogeneity would be very low. However, we still want to reflect the fact that *Harbour* is closely correlated to 海底隧道. We can use the following heuristic to achieve this:

For a given word W_i in a trigram of (W_{i-1}, W_i, W_{i+1}) with context heterogeneity (x, y):

if $W_i(x) = 1$	if the left heterogeneity of a word equals to one
$\quad W_i(x) \leftarrow W_{i-1}(x)$;	back off to one previous word
if $W_i(y) = 1$	if its right heterogeneity equals to one
$\quad W_i(y) \leftarrow W_{i+1}(y)$;	shift to the following word
return $(W_i(x), W_i(y))$;	

For *Harbour*, since its left heterogeneity is one, we back off to look at the left heterogeneity of *Cross*. Its right heterogeneity is also one, so we shift to the right heterogeneity of *Tunnel*. As a result, *Harbour* has the same heterogeneity as *Cross Harbour Tunnel* which is closely correlated to that of 海底隧道.

Using this method, we have improved the context heterogeneity scores of 人權/*human rights*, 基本法/*Basic Law*, ,二讀/*Second Reading* and 香港/*Hong Kong*.

Effect of words with multiple functions

As mentioned earlier, many Chinese words have multiple part-of-speech tags such as the Chinese for *declaration/declare, development/developing, adjourned/adjournment*, or *expenditure/spend*. Therefore these words have one-to-many mappings with English words.

We could use part-of-speech taggers to label these words with different classes, effectively treating them as different words.

Another way to reduce one-to-many mapping between Chinese and English words could be to use a morphological analyzer in English to map all English words of the same roots with different case, gender, tense, number, capitalization to a single word type.

Effect of word order

We had assumed that the trigram word order in Chinese and English are similar. Yet in a non-parallel text, nouns can appear either before a verb or after, as a subject or an object and thus, it is conceivable that we should relax the distance measure to be:

$$\mathcal{E} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_1 - y_2)^2 + (y_1 - x_2)^2}$$

.

We applied this measure and indeed improved on the scores for nouns such as *vessels*, *Government*, *employers*, *debate*, *prosperity*. In some other languages such as French and English, word order for trigrams containing nouns could be reversed most of the time. For example, *air pollution* would be translated into *pollution d'air*. For adjective-noun pairs, Chinese, English and even Japanese share similar orders, whereas French has adjective-noun pairs in the reverse order most of the time. So when we apply context heterogeneity measures to word pairs in English and French, we might map the left heterogeneity in English to the right heterogeneity in French, and vice versa.

4.2.5 Experiment 2: Finding the word translation among a cluster of words

The above experiment showed to some extent the clustering ability of context heterogeneity. To test the discriminative ability of this feature, we choose two clusters of known English and Chinese word pairs *debate*/辯論. We obtained a cluster of Chinese words centered around 辯論 by applying the Kvec segment co-occurrence score (Fung & Church 1994) on the Chinese text with itself. The Kvec algorithm was previously used to find co-occurring bilingual word pairs with many candidates. In our experiment, the co-occurrence happens within the same text, and therefore we got a candidate list for 辯論 that is a cluster of words similar to it in terms of occurrence measure. This cluster was proposed as a candidate translation list for *debate*. We applied context heterogeneity measures between *debate* and the Chinese word list, with the result shown in Figure 4.6 with the best translation at the top.

The asterisks in Figure 4.6 indicate tokenizer error. The correct translation is the third candidate. Although we cannot say at this point that this result is significant due to the small size of test set, it is to some extent encouraging. Experiments on a larger test set will need to be carried out at a later stage.

4.2.6 Non-parallel corpora need to be larger than parallel corpora

Among the 58 words we selected, there is one word *service* which occurred 926 times in the English text, but failed to appear even once in the Chinese text (presumably the Legco debate focused more on the issue of various public and legal *services* in Hong Kong during the 1988-90 time frame than later during 1991-92. And in English they frequently accuse each other of paying lip *service* to various issues). We expect there would be a great number of words which simply do not have their translations in the other text. Words which occur very few times also have unreliable context heterogeneity. A logical way to cope with this sparse data problem is to use *larger* non-parallel corpora. Our texts each have about 3 million words, which is much smaller than the parallel Canadian Hansard used for the same purposes. Because it was divided into two parts to form a non-parallel corpus, it is also half in size to the parallel corpus used for word alignment (Wu & Xia 1994). With a larger corpus, there will be more *source* words in the vocabulary for us to translate, and more *target* candidates to choose from.

Figure 4.6: Sorted candidate list for *debate*

0.117371	debate	觸/*
0.149207	debate	月十/*
0.155897	debate	辯論/debate
0.158305	debate	恢復/resumption
0.185699	debate	休會/adjournment
0.200486	debate	委員會審議階段/Amendment stage of the Council
0.233063	debate	月二十/*
0.246826	debate	條第/*
0.255721	debate	於一/*
0.268771	debate	二讀/Second Reading
0.284134	debate	條例草案二讀/Second Reading of the Bill
0.312637	debate	九九/*
0.315210	debate	條例草案二讀動議/moved to Second Reading of the Bill
0.349608	debate	委員會審議/Council Amendment
0.367539	debate	今午/this afternoon
0.376238	debate	這次/this time
0.389296	debate	全局/Council
0.389693	debate	照會議常規第/*
0.403140	debate	獲按/*
0.404000	debate	條例草案經過二讀/Second Reading of the Bill passed

4.2.7 Discussion

Context heterogeneity is a feature showing the existence of statistical correlations between words and their translations even in a non-parallel corpus. We have shown initial results of matching words with their translations in a English-Chinese non-parallel corpus by using context heterogeneity measures. We need to further explore the power of context heterogeneity as a clustering measure and as a discrimination measure. Given two corresponding clusters of words from the corpus, context heterogeneity could be used to further divide and refine the clusters into few candidate translation words for a given word. We need to explore how the context heterogeneity feature can be integrated into a domain term translation with other statistical and linguistic features. We will also extend the binary vector representation to find clusters of monolingual words as one step further towards finding term translations.

4.3 Word context length histogram

We have shown in the previous section that context heterogeneity is a correlation feature between pairs of translated words in a non-parallel corpus. We have proposed to use it as *part of* a translation system for compilation of bilingual lexicon from non-parallel texts. This is because we found that it is largely a clustering feature which groups similar words together. It is not powerful enough to pick out the best translation for a word. Further work led to the discovery of another statistical correlation feature of word pairs in non-parallel corpus - the **context length histogram**.

In this section, we demonstrate a pattern matching method by using the **context length histogram** (Fung 1996), to correlate pairs of translated words. We also show how **space-frequency analysis** is used for matching such word pair signals for translation. We again use the compiled non-parallel version of the HKUST bilingual corpus for our experiments.

4.3.1 Algorithm overview

The procedure for our experiments is as follows:

1. **Segment both the English and the Chinese texts** by a class of delimiters.
2. **Compute segment lengths** of both texts and record them.
3. **Compute the context length histogram** of each English and Chinese word.
4. **Transform the histograms using wavelet basis functions** into space-frequency domain.
5. **Dynamic Time Warping to match** the wavelet transformations of all pairs of English and Chinese words.
6. **Obtain a bilingual lexicon** from matching results.

4.3.2 Segments of texts in English and Chinese

Segmental information had been found to be useful in providing statistics for word pair matching. In parallel corpora, a long sentence in one language would correspond to a long sentence in its translation to another language. Such information could be used to align sentences and word pair matching could be carried out from aligned sentence pairs (Kupiec 1993; Smadja & McKeown 1993; Dagan *et al.* 1993; Wu & Xia 1994). In noisy parallel corpora, sentence boundaries are often unreliable. Texts in noisy parallel corpora could be segmented by word pair anchor points (Fung 1995) and aligned. However, such sentence or segment information is text-dependent. Given a non-parallel bilingual text, there is no such sentence or segmental mapping - given any sentence in one language, its translation does not even appear in the other text. We need to find segmental correspondence which are text-independent.

It is generally found that English sentences, delimited by a dot full-stop are shorter than Chinese sentences, delimited by a round circle. (English full stops are syntactic marks whereas Chinese full-stops correspond roughly to the end of some semantic meaning.) Very often, Chinese would use commas or semi-colons instead where in English a full-stop would have been used.

Therefore, full-stops in English and Chinese are not good corresponding delimiters for segments. On the other hand, we believe punctuations in general are still good delimiters. So we divided both the English and the Chinese texts into segments delimited by one of the following punctuations: an English full-stop, a Chinese full-stop, a comma, a question mark, a semi-colon or an exclamation mark.

We hypothesize that if a word appears frequently in short segments, then its translation would also tend to appear more frequently in short segments. For example, the word *figure* is often seen in segments like *"We will show this in figure 1"*, *"The ... is shown as follows in figure 1"*, *"in the left figure"*, etc. It rarely appears in long segments. Its translation is used in the same way in Chinese. We define the length of an English segment to be the number of words in that segment. However, the length of a Chinese segment is defined as the number of characters in that segment. This is used for two reasons: (1) Chinese texts have no space delimiters between words, we would need a tokenizer to insert spaces. Such tokenizers are not perfect and therefore can cause mismatching between English and Chinese words. For example, *air pollution* is counted as two words in a English segment, but its Chinese translation would be counted as one word by a Chinese tokenizer. If we used characters, then there would be four characters in Chinese. (2) Over all, Chinese texts are more concise than English. For example, Chinese has very few if any function words compared to English. In general, there would be fewer words used in Chinese than English to express the same meaning. (2) compensating for (1), the number of characters in a Chinese segment correspond roughly to the number of words in an English segment for the same meaning.

4.3.3 Histograms of context segment lengths

Next, we compute the histogram of context segment lengths for each word in English and Chinese. Assuming the maximum segment length is 100[3] and the minimum is one. This is also the range for the x-axis for the histogram plot. For *Government*, part of its concordance in the English text is shown in Figure 4.7, one segment per line. The first field indicates the length of each segment. The y value of the histogram indicates how many times *Government* occurs in a segment of length x.

We plot the histogram of the values in the length field from the concordance list. The histogram for *Government* is shown in Figure 4.8. Since this information would be used to match words in non-parallel texts with very different occurrence frequencies, we normalize this graph so that the total area under the graph would be one. We obtain a graph of the same shape but with different y values. This is shown in Figure 4.9.

As another example, the same procedure is applied to the word 政府 *Government* in Chinese. Its normalized histogram is shown in Figure 4.10. Note the visual similarity between the two graphs in Figures 4.9 and 4.10.

4.3.4 Wavelet transformation for matching histograms

From the above-mentioned plots, we can see that in general, the histogram of a word in English and that of its translation in Chinese have similarities perceivable to the human eyes, i.e. we can see they have similar *shapes* [4]. To match these shapes algorithmically, however, is much more

[3] In actual case, the maximum is usually around 70

[4] It might be helpful for us to carry out this perception test on human subjects.

Figure 4.7: Part of the concordance for *Government*

length	concordance
20	council has brought with it a greater diversity of views and a closer scrutiny of the work of the Government
23	The policies of the Government which I shall put before you this afternoon will require a great deal of work from the Administration
18	The Government have already taken a number of measures to try to reduce the size of the problem
20	There have been calls for the Government to change its policy to allow contractors to import workers for specific projects
30	And it continues actively to look for opportunities to provide services through bodies outside the Government where there are clear advantages in terms of cost - effectiveness and management flexibility
12	A number of major facilities are currently being built by the Government
28	The Government have taken a number of steps in the past year towards achieving our objective of providing adequate accommodation for all by the turn of the century
24	The Government 's policy is to provide a legal framework which will give owners of private buildings the opportunity to manage their buildings effectively
18	It will advise the Government on what further measures are needed to improve the management of private buildings
17	These demands stem both from the growing range and complexity of the services provided by the Government
12	Many cut across the boundaries of several different Government branches and departments
6	from within and outside the Government
18	The Government will also propose legislation to curb illegal gambling and reduce the problems caused by vice establishments
13	the Government have since May prosecuted groups arrested at their places of employment
15	Starting such a programme is the main objective of our discussions with the Vietnamese Government

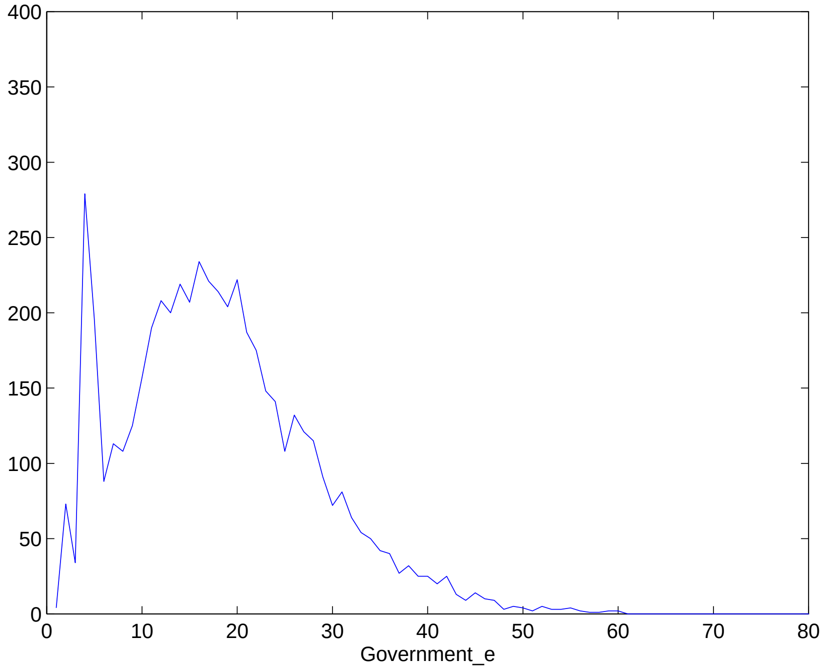


Figure 4.8: Histogram of *Government* in English

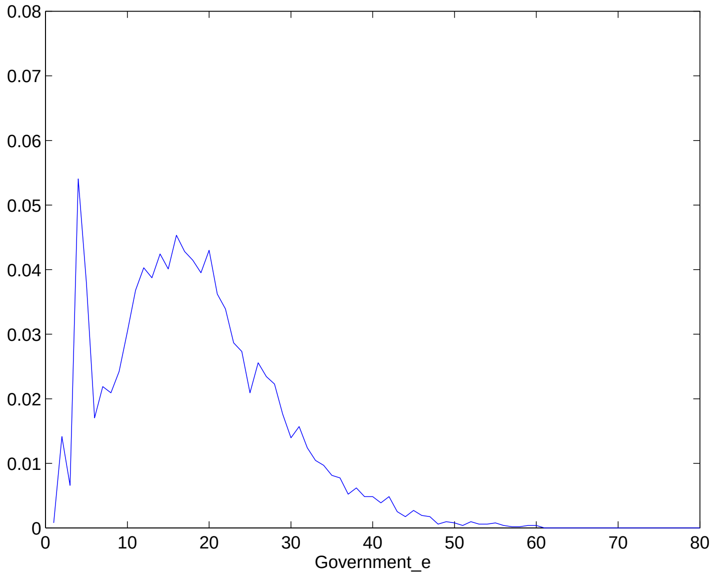


Figure 4.9: Normalized histogram of *Government* in English

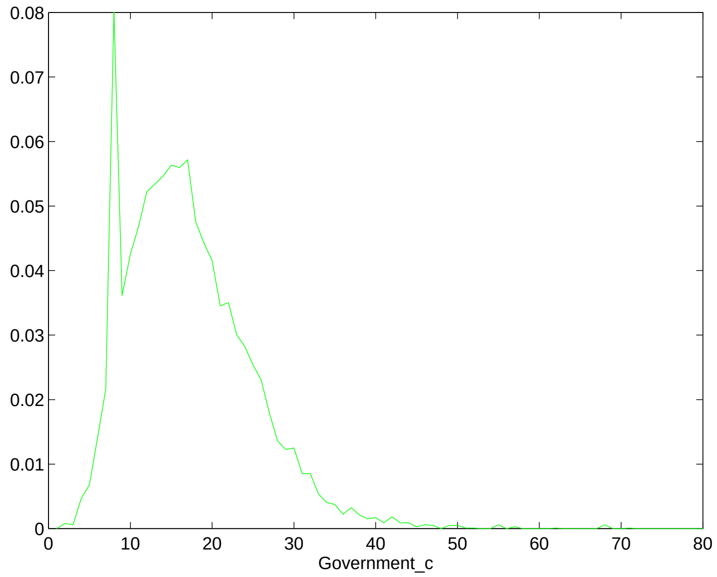


Figure 4.10: Normalized histogram of *Government* in Chinese

difficult. The plot pairs are non-linearly scaled, warped versions of each other. They have the following properties:

- The plots of a corresponding word pair are not linearly scaled in the x-axis. So cross-correlation-like matching methods cannot be applied.
- In addition, they are not linearly scaled in the y-axis, which means a simple warped matching cannot find the most correlated pairs.
- Yet, they do have general matching shapes. i.e. if the graphs were smoothed, the general *humps* correlate closely between the plots of a word pair.
- The little peaks and valleys also have some correlations, if there is a small peak followed by a big peak in one plot, they will appear in the same order in the plot of the translated word.
- The graph is a combination of a general shape, and the salient peaks and valleys

Thus, we need a way to analyze the general hump, as well as the peaks and valleys of the plots, preserving the order in which they appear.

We would like to transform the original plots into a different domain to emphasize the characteristics of a plot and to reduce superfluous information.

If we treat the x-axis in the plots as the time axis, it becomes evident that we need signal processing by time-frequency analysis. Wavelet transformation is one type of such analysis (Rioul & Vetterli 1991; Strang 1989; Chui 1992). We use a wavelet transformation to analyze our signals[5]

[5] Wavelet transformation is two directional, i.e. one can analyze the signal in time-frequency domain by transformation into wavelets; and synthesize the original signal from the transformed wavelets. However, we only concern ourselves with the analysis stage here.

After transformation, the x-axis is still the same as the original one, but the y-axis would denote frequency. The value at the point (x, y) would denote the intensity i where i is the value of the wavelet at frequency y at the point x in "time". At any given x, the plot is a weighted combination of different wavelets at frequencies marked by the y-axis. The weight is shown by i.

Wavelet transformation can be thought of as a *magnifying glass.* If one is looking at high frequency parts of the original signal, one uses a higher resolution for the magnifying glass. If one is looking at the part of the signal with low frequencies, we use a lower resolution magnifying glass. The magnifying glass itself is a wavelet basis function, with different *magnifying power* corresponding to different scaling of the basis function. We use an approximation of a wavelet basis function as shown in Figure 4.11. It is the difference of two Gaussians:

$$\begin{aligned}
h &= \left(\frac{1}{\sqrt{2\pi a2}} e^{-0.5u^2/a2^2}\right) \\
&\quad -\left(\frac{1}{\sqrt{2\pi a}} e^{-0.5u^2/a^2}\right) \\
\text{where } a &= 1, 5, 10, \ldots, N \\
a2 &= 0.5a \\
\text{and } u &= -5a : 5a
\end{aligned}$$

The total area of the basis function is zero. Different a would contract or dilate the basis function, thereby changing the magnifying power.

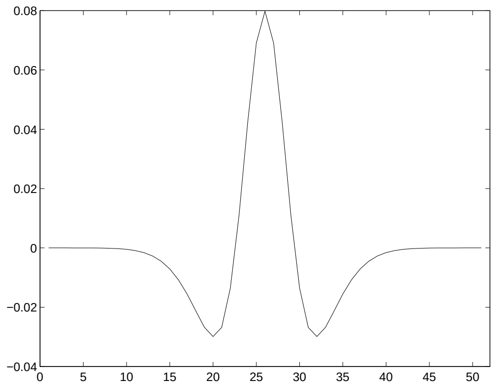


Figure 4.11: Difference of two Gaussians as the basis function

To look through the magnifying glass, we convolve the wavelet functions with the interpolated graph of the original signal V at all positions on the x-axis[6]:

$$\text{val}(V1_i[m]) = \int_0^{100} h \cdot V'$$

The result of the convolution is a matrix with the values $\text{val}(V_i[m])$ quantized into the color on the wavelet transformation plots. The colors of the plots range from black-red-yellow to white,

[6]The Matlab tool *interp* is used to re-sample the histogram sequence with 20 times the original data points.

with increasing values, quantized into 256 levels between the minimum and the maximum of the wavelet graph. High intensity (white color) at point (x, y) corresponds to high convolution value which is an indication that at the point x, there is a peak of frequency y.

The wavelet transformation plots of word pair *Government* are shown in Figures 4.3.4 and 4.12. The x-axis in these graphs indicate the interpolated value which is 20 times the original sample points. The y-axis indicates the different frequency bands, ranging from 0 to 50.

Looking at Figure 4.3.4, where the wavelet transformation plots of the word *Government* in English is superimposed with its original histogram signal in blue, we see that there is a small white patch at around $y = 5$, high frequency. This corresponds to the sharp peak, a local maxima in the original signal at around $x = 180/20 = 9$. The bigger white patch at lower frequencies and at around $x = 300/20 = 15$ corresponds to the general shape of the original signal having a gentle hump there. We see corresponding sharp peaks and gentle hump in the signals for the Chinese word, and its wavelet transformation in Figure 4.12.

The wavelet transformation provides us an analytical way of looking at the histogram signals. By quantizing the signals, the *relative* peaks and valleys on the signals become more salient.

4.3.5 Dynamic time warping on frequency-variant delta vectors

Now we can see that the transformed signals of word pairs are more or less warped signals of each other in the x-axis. To match the transformed graphs, we use Dynamic Time Warping (DTW) on the *difference* of the intensity at each frequency. At each y, the $x-1$-dimensional row vector is the *delta encoder* of the original x-dimensional vector. We compare the row vector of a word $V1$ to that of another word $V2$ at the same y_i value, giving a score $DTW(V1, V2, y_i)$. The total correlation score between two graphs is $\sum_{i=1}^{N} DTW(V1, V2, y_i)$ where $DTW(V1, V2, y_i)$ is:

- **Initialization**

$$
\begin{aligned}
\varphi_1(1,1) &= \zeta(1,1) \\
\varphi_1(i,1) &= \zeta(i,1) + \varphi(i-1,1]) \\
\varphi_1(1,j) &= \zeta(1,j) + \varphi(1,j-1) \\
\text{where } \varphi(a,b) &= \text{minimum cost of moving} \\
& \text{from } a \text{ to } b \\
\zeta(c,d) &= \mathcal{E}(V1[c], V2[d]) \\
& \text{Euclidean distance between} \\
& V1[c] \text{ and } V2[d] \\
\text{for } i &= 1, 2, \ldots, N \\
j &= 1, 2, \ldots, M \\
N &= \dim(V1) \\
M &= \dim(V2)
\end{aligned}
$$

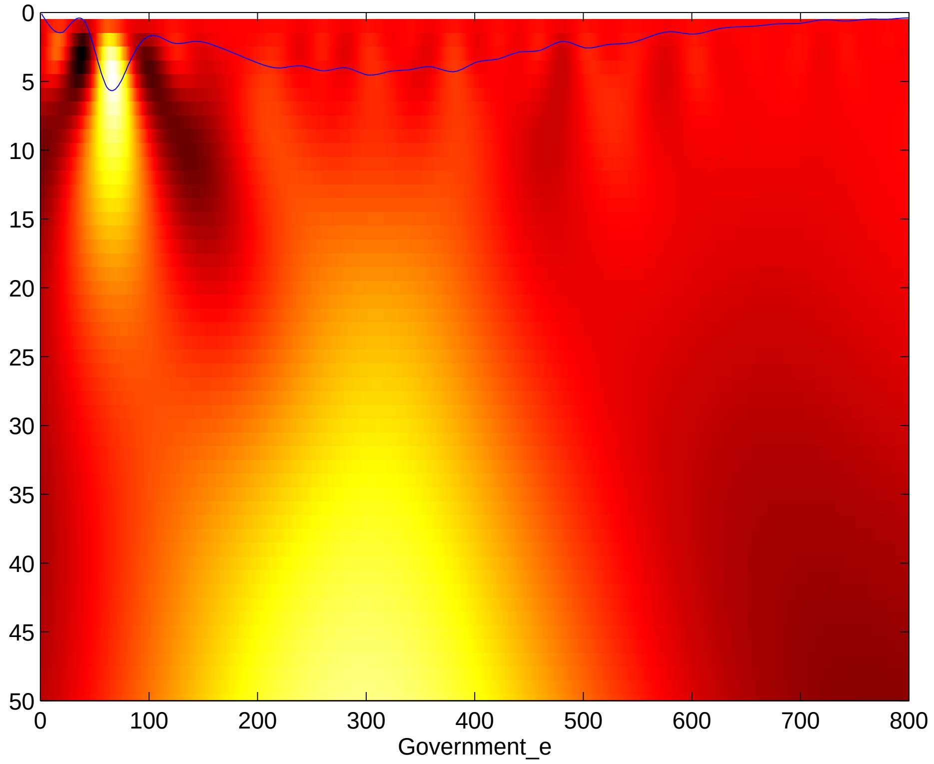


Figure 4.12: Histogram and wavelet plots of *Government* in English

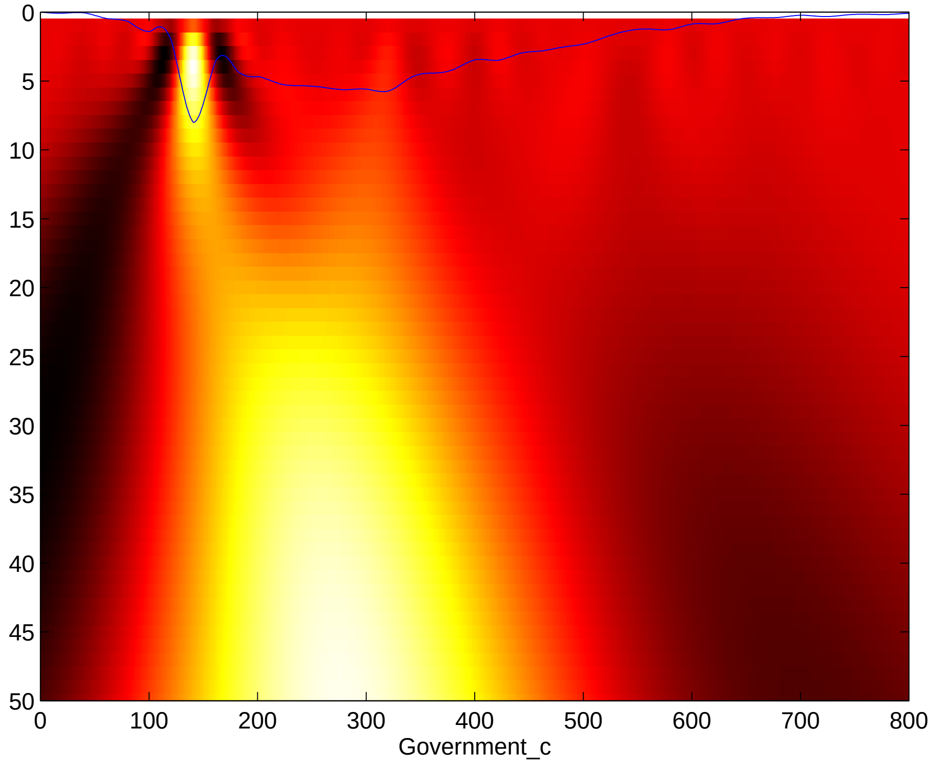


Figure 4.13: Histogram and wavelet plots of *Government* in Chinese

- **Recursion**

$$
\begin{aligned}
\varphi_{n+1}(i,m) &= \min_{1\leq l\leq 3}[\zeta(l,m)+\varphi_n(i,l)] \\
\xi_{n+1}(m) &= \operatorname*{argmin}_{1\leq l\leq 3}[\zeta(l,m)+\varphi_n(i,l)] \\
\text{for} \;\; n &= 1,2,\ldots,N-2 \\
\text{and} \;\; m &= 1,2,\ldots,M
\end{aligned}
$$

- **Termination**

$$
\begin{aligned}
\varphi_N(i,j) &= \min_{1\leq l\leq 3}[\zeta(l,m)+\varphi_{N-1}(i,l)] \\
\xi_N(j) &= \operatorname*{argmin}_{1\leq l\leq 3}[\zeta(l,m)+\varphi_{N-1}(i,j)]
\end{aligned}
$$

4.3.6 Discussion

We have tested this algorithm on more than 50 word pairs, the result shows that about 40 of the words match most closely to their translations in the other language. Evidently, we still need to experiment with the context length histogram feature on a larger set of words. We will combine the wavelet transformation of context segment length histogram information with other statistical and linguistic features of words for word translation. The other features will include context heterogeneity, word relations etc. They could be combined as weighted sums or cascaded filters in a system. They can also be combined into a single vector for eigenvalue transformation and matching. We will also investigate the linguistic justification of this feature as well as analyse its domain-specificity.

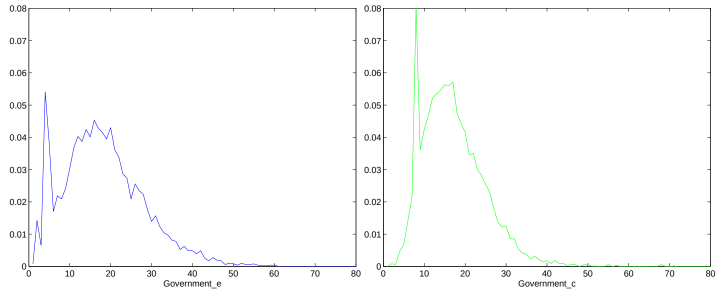


Figure 4.14: Normalized histogram of *Government* in English and Chinese

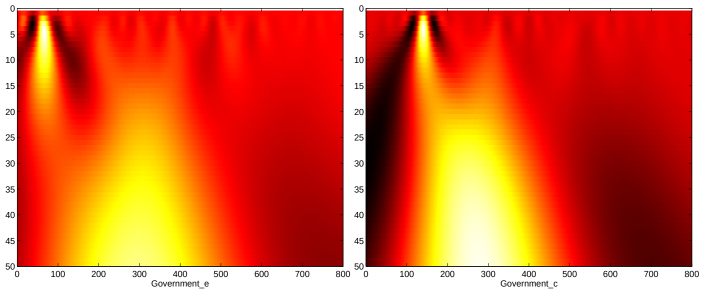


Figure 4.15: Space-frequency plots of *Government* in English and Chinese

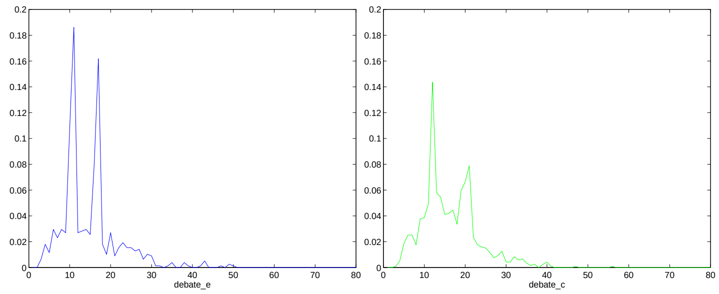


Figure 4.16: Normalized histogram of *debate* in English and Chinese

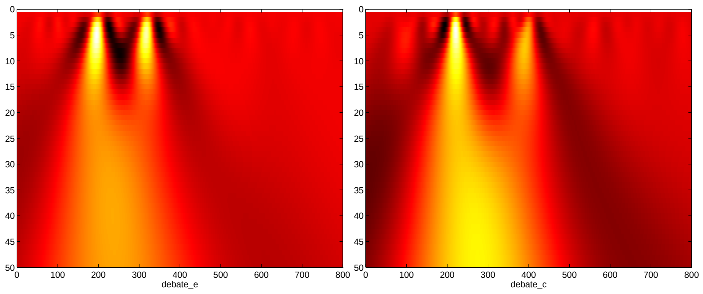


Figure 4.17: Space-frequency plots of *debate* in English and Chinese

Chapter 5

Proposal of remaining work

We have devised an algorithm for translating domain words from noisy parallel corpora without sentence alignment. We have started experimenting with finding bilingual word pairs from non-parallel, same domain, monolingual texts and have shown some initial results. There remains some work to be done in order to achieve our final goal of developing a noun phrase translation system from noisy parallel and non-parallel texts.

To further evaluate and improve on our initial results with non-parallel corpora, we need to collect more same-domain, non-parallel corpora for experiments. To achieve better precision in lexicon compilation, we need to carry out more experiments on finding new word features for bilingual lexical entries in monolingual texts. We will experiment on combining various statistical features in a single vector representation and use eigenvalue estimation of basis vectors for classification. To translate English noun phrases, we will use pattern matching tools to extract simple noun phrases from these corpora and use our algorithm to translate them. Finally, we will port different parts of our work to an integrated system of domain-specific term translation from non-parallel corpora. Further evaluations will be carried out at different stages of our experiments to check the individual and combined performance of various features.

5.1 Collecting more domain-specific texts

We have previously compiled a non-parallel corpus using documents from different years. However, this is not necessarily a good choice of non-parallel texts. The Hong Kong Legislative Council members' debate on issues varied from year to year. The overall domain remains in the social and political issues of Hong Kong, but the micro-domains changed from year to year. There was more concentration on transportation issue one year, more on the airport construction the next; proposals of new education projects would lead to new issues being discussed.

In real applications of domain term translation, one might wish to choose from pairs of texts with closer domain resemblance. We plan to collect and use more domain-specific texts in further experiments. Some possibilities include the following:

- Same domain texts from MULTEXT (Multilingual text tools and corpora), a project funded in the Commission of European Communities Linguistic Research and Engineering Program. This corpus consists of 2 million words per language from six languages (English, French, Germain, Italian, Spanish, Dutch), composed of comparable types of texts from two or three different domains.

- Part of the ECI/MCI Corpus 1 (European Corpus Initiative Multilingual Corpus 1) which contains approximately 97 million words in 27 (mainly European) languages. We plan to use newspaper texts from the same time period in different languages, assuming that the newspapers report on similar topics in the same time period.
- Wall Street Journal articles from various time periods, and part of the Nihon Kezai Shimbun texts consisting of 30 million words from the largest Japanese financial news daily newspaper. These two corpora can be used in conjunction as a non-parallel corpus. The latter is available from the Linguistic Data Consortium.
- The AP Newswire material in English and French from the same period, to form a non-parallel news domain corpus.

5.2 Statistical work

We have used a dynamic recency vector representation and a binary position vector representation for translating words in a noisy parallel corpus. We will experiment on extending the algorithm to noun phrase translation. We have also explored using context heterogeneity and context length histogram for word correlation in a non-parallel corpus. We need to develop further both these features. In addition, we will investigate other statistical word features for non-parallel corpus. Finally, we will combine all the statistical features into a single vector and apply standard pattern classification techniques on it.

5.2.1 Context heterogeneity

We used ordered pairs to represent context heterogeneity, looking at only the immediate neighboring one word. Further tests will be carried out on varying this neighbor constraint. We will extend the algorithm to test using two and then three immediate neighboring words. The clustering power of context heterogeneity will be tested on larger sets of words. Once it is established that context heterogeneity is a good clustering feature, we will use it as part of extension to term translation.

5.2.2 Context length histogram and non-linear matching

The first step for extending this algorithm is to try it on a larger set of words. We will extend the algorithm to automatically take all English nouns and proper nouns in our corpus and find the corresponding Chinese words. This will automate the evaluation process. It will also become one component of the final system.

We will then experiment on different non-linear matching on the wavelet transformations. Currently, all points in all frequency ranges are matched by Dynamic Time Warping. It is worth investigating how the complexity of matching can be reduced. We will investigate using eigenvalue analysis to obtain the essential features in the wavelet transformation.

5.2.3 Word relation matrix

We can also derive the relationship between a word and its translation with respect to other *known* pairs of words. For example, if we know the translation for *Commerce* is *Commerce* in French, than we can use co-occurrence information to hypothesize that *House* in *House of Commerce* should be translated into *Chambre* in *Chambre de Commerce.* In this case, *Commerce* and *House* are collocations. On the other end of the spectrum, we can also measure the dissimilarity between a word and a known word. For example, *House* and *market* are unlikely to co-occur, they might appear in the same segment very rarely. Likewise, *Chambre* and *marcheé* would rarely co-occur. These type of co-occurrence measures can be represented in the linear positional binary vector we previously used for word matching in noisy parallel corpora. Given an *unknown* word, its co-occurence and dissimilarity in occurrence with various n known seed words will be represented in a $n \times k$ binary matrix, where k represents the dimension of each binary vector.

A general algorithm is as follows:

- **input** a word list in English e, a word list in Chinese c;
- **given** a small bilingual dictionary of known words $(e1, c1), (e2, c2), (e3, c3) \ldots, (en, cn)$;
- compute the relational vector v of e to ei where $1 \leq i \leq n$ such that $v[i]$ = correlation1(e,ei), where correlation1() could be collocation information;
- compute the relational vector w of c to ci where $1 \leq i \leq n$ such that $w[i]$ = correlation1(c,ci); and
- compute correlation2(v,w); if it is high, e and c are translated word pairs, otherwise they not.

correlation1 and *correlation2* are two different correlation functions. They can be collocation mutual information or other matching functions. When we compute v and w for all domain words in English and Chinese, the word pairs with highest *correlation2* scores are selected as translations of each other. We need to experiment on this and if successful, use it as a filter or in combination with other word features as part of the bilingual lexicon compilation program.

5.2.4 Eigenvalue matching of combined statistical feature vector

We will experiment on combining the above mentioned statistical features into a single vector representation with n dimensions, say, $\vec{X}$. We will then carry out discriminant analysis on all such vectors. Discriminant analysis will emphasize the differences between vectors of unrelated words, and the closeness of vectors of correlated words.

$\vec{X}$ will then be transformed into a compressed vector $\vec{Y}$ with m dimensions where $m < n$. This transformation is expressed by

$$\vec{Y} = A^T \vec{X}$$

where A is an $n \times m$ matrix and its column vectors are linearly independent. For discriminant analysis to classify words according to their vector representations, we need to find the A that optimizes the class separability criteria formulated from $\vec{X}$ (Fukunaga 1990). These criteria are as follows:

- The **within-class scatter matrix** S_w, shows the scatter of samples around their respective class expected vectors.

- The **between-class scatter matrix** S_b, is the scatter of the expected vectors around mixture mean.
- The **mixture scatter matrix** S_m, is the covariance matrix of all samples regardless of their class assignments.

The S matrices characterizes the distribution of various words in the vector space. The optimization of A is done by optimizing one of the various combinations of the S matrices. We will choose the optimized A represented by $S_{2\vec{X}}^{-1} S_{1\vec{X}}$, where S_2 and S_1 are two of the three S matrices.

5.3 Linguistic knowledge

Our statistical algorithms include two main steps: finding word features and matching word pairs. There are many ways where these steps can be improved by using linguistic knowledge. First, since we are concentrating on translating noun phrases, only words with POS belonging to an English noun phrase should be selected for translation. Second, even though our system is designed to be language-independent, linguistic knowledge about the relationship between noun phrases and other phrasal structures in the source and target languages can be obtained in a preprocessing stage.

5.3.1 Taggers and NP finder

We will use the Church POS tagger (Church 1988) to tag the English corpora. We will use JUMAN for tagging Japanese.We will also use a Chinese statistical tagger from HKUST to tag the Chinese corpus.

We are currently implementing a simple noun phrase finder based on Lex, for extracting simple noun phrases from tagged texts of any language. This tool will be written in Perl and consists of a series of filters to CREP. All texts including Chinese and Japanese would be tagged and then converted into ASCII format, as input to this tool. Given a simple template of noun phrase form, this tool will either output a list of noun phrases found or the list of sentences containing them with the noun phrase bracketed.

These tagged texts will be used first in our experiments to find the percentage correspondence in the language to English noun phrases. Later, the POS information of a corpus will be used in helping our algorithm as described in the following section.

In a first scenario, we plan to extract only noun phrases in English, and then match words in the other language to the constituent words of the English noun phrases. In the second scenario, we will use our NP finder to extract noun phrases from both languages and try to match them using our algorithm. The advantage of the first approach is that we can circumvent the noun phrase extraction step of the other language. In addition, this approach can produce the Chinese translation to an English noun phrase even if the translation is not itself a noun phrase, which is a common case for translation. The disadvantage is that the translated constituent words in another language, especially in Chinese, may or may not form a full noun phrase. The system will use both strategies in the development stage and then undergo evaluation. The winning strategy will be the one the system will use in the future.

5.3.2 Relation between English noun phrases and their translations

It is not clear that English noun phrases would translate exclusively into noun phrases in any other language. So we would like to find out the statistics of what English NPs actually map into in Chinese (Japanese and/or French).

We plan to use parallel corpora for this experiment. For English/Chinese for example, we will use a word-aligned corpus of tagged English and Chinese (Wu & Xia 1994). Noun phrase bracketing will be carried out on the English part. The Chinese will be tagged without bracketing. We will implement a program which counts what Chinese word groups English noun phrases align to. Given an English sentence with NP bracketing and its aligned Chinese word groups, the program counts the POS of all the Chinese words aligned to the English NP. The accumulative counts of all English and Chinese sentence pairs in the corpus will yield the statistics of how English noun phrase map into Chinese POS groups. The statistics will be expressed in an probabilistic Markovian model similar to the Coerced Markov Model for translation-driven tagging (Fung & Wu 1995).

The Markovian model will be used as part of our bilingual domain term translation system. Given a noun phrase in English, and a possible set of words in Chinese and their POS, found by other word features, our system will use the Markovian model to predict how likely this set of words is a translation of the English NP.

5.3.3 Chinese NLP knowledge

We will attempt to use some linguistic knowledge available about Chinese to analyse the language and help the performance of our final Chinese/English bilingual lexicon compilation system. We intend to refer to literature in Chinese noun phrase construction, compound noun construction, nominalization in domain specific texts, scientific translation (Li 1985; Li & Thompson 1989; Huang 1989) etc.

5.3.4 Linguistic justification of statistical features

It was quite straightforward to see the rationale behind the dynamic recency vector, the position binary vector, and their related statistical features. For noisy parallel corpora, we made use of the fact that words/noun phrases and their translations occurr in *similar* positions at *similar frequencies*. This fact was represented by the above-mentioned features. On the other hand, there is no immediate linguistic justification for why context heterogeneity and especially context length histograms would relate words to their translations in a non-parallel corpus. Yet, they have proven to be effective to some extent for our purposes. While our immediate next step is to evaluate these features further on larger sets of words, we believe it will be helpful to understand the linguistic justification of these features. Our understanding can lead to the discovery of their strengths and weaknesses, as well as provide insight to other possible knowledge and features we can incorporate into our system.

5.4 A noun phrase translation system

Our algorithm for finding word translations in noisy parallel corpora shows promise for finding translations of compound nouns or noun phrases. Similarly, the algorithms for finding word features in non-parallel corpora can be augmented to a noun phrase translation algorithm.

We will implement a system integrating our noun phrase finder and the statistical algorithms to translate English noun phrases into other languages from non-parallel corpus. This system will use minimal linguistic knowledge pertinent to the language we use as input so that it can be easily ported to other languages using the corpora mentioned in Section 5.1 as input.

So far, our code for different statistical features and matching functions are separate. We will need to port these codes into a single program in C, driven by UNIX shell scripts, to form a single system.

The overall algorithm of the system will be:

- **input** a text in English and a text in another language of the same domain;
- **tag** both texts with individual taggers and *extract* noun phrases in English;
- **form two word lists** of all constituent words of all noun phrases in English and all words in the other text;
- **compute the correlation** between words in the English word list and all words in the other list according to various word features. The various word features can be combined either in cascade, with the most clustering feature as the first filter, the most discriminative feature the last filter; or into a single vector. The combined single vector will be transformed into eigenvalues for matching; and
- **select the highest correlation pairs** as translated words and append them as entries to a bilingual noun phrase lexicon.

5.5 Evaluation

Since the ultimate goal is to develop a translation tool which finds translations of English noun phrases from either noisy parallel or non-parallel but domain-specific data, all our evaluations will be carried out on noisy parallel or non-parallel test corpora. It should be noted that we need to bear in mind what the application of our system will be in order to do appropriate evaluations accordingly. We are developing domain term translation systems primarily as translator aids, and then as part of an MT system. For the first application, the system only needs to suggest a list of terms for the translator, assuming the translator is able to pick out the best translation. Therefore, evaluation of **top N candidate** translations is appropriate. For the second application, the **precision and recall** of the system in choosing the best translation need to be evaluated. Moreover, our evaluations will be two types: **analytical** evaluation and **black-box** evaluation. Analytical evaluations will assess the clustering and discriminative power of various statistical and linguistic features we integrate into the system individually. Black-box evaluation will evaluate the system performance in its applications, without looking into individual features. Analytical evaluations can guide us in the development of the system and will be carried out throughout the implementation stage, whereas black-box evaluations provide an indicator of the system performance and will be carried out when the implementation is finished.

We plan to evaluate our algorithms and system in the following different stages:

1. In the first stage, we need to analytically evaluate the efficiency of each individual word feature such as context heterogeneity, context histogram spectrum, and word correlation.

We will implement a single program in C integrating all these features. The program will have flags representing each feature. When we turn on a flag, the program will choose that particular feature as correlation feature between *single* words. We will omit the noun phrase extraction step in the evaluation of statistical word feature efficiency because the relative result of translating single words will give the relative efficiency of each feature.

The evaluation will be done on a test corpus. The corpus will be tagged first. Given the tagged corpus as input and a flag indicating which feature to evaluate, the program will find bilingual word correspondences with respect to the correlation value of this feature. The output will be a list of bilingual words with their correlation scores.

For **context heterogeneity**, we will evaluate on a list of randomly selected words with known translations. For each word in the list, the program with context heterogeneity as the sole correlation feature will output a list of translation candidates sorted by their scores. Since we already know the correct translation, we will look for its position in the translation candidate list. From our previous tests, it is unlikely for most of the correct translations to be the top candidate in a list. However, we will compute the number of correct translations in the first ten percent of candidates, second ten percent, third ten percent etc. The results will be plotted in a histogram.

To evaluate the **context length histogram** feature, we will implement a program which match all English nouns and proper nouns in a corpus to all words in Chinese using this feature. For each English word, a top one candidate and then a top 10 candidate list will be compiled, sorted according to the matching scores. Percentage precision of both lists will be computed.

To evaluate the **word relation matrix**, same test will be carried out as that for context length histogram. However, we also need to test the effect of the seed word list.

2. In the second stage, we will evaluate the clustering and discriminative power of these features relative to each other so as to decide the weight of each feature in the system. Each feature will be assigned a weight from 0 to 1, with increasing significance in the system. If a feature has zero weight, it is not used by the system at all. The correlation score will be normalized by the weights of the features. For example, if we have features $\langle a, b, c, \ldots, \rangle$, with weights $\langle i, j, k, \ldots \rangle$. The correlation score s will be the raw weighted sum $\frac{W_a + W_b + W_c \ldots}{i + j + k \ldots}$.

3. Third, we will evaluate the system precision and recall in compiling a bilingual lexicon from a same domain, non-parallel corpus. This will be black-box evaluation.

   The output will be evaluated in four ways:

   - The percentage correct of the first 1000 or so pairs of bilingual words sorted by their correlation scores. This will require human evaluators. We plan to use three or more human evaluators who are native speakers of languages other than English. This part evaluates the recall and precision of the system to output a bilingual domain term dictionary.
   - Percentage correct of the translation of a randomly selected list of around 50 English words. The correct translation would be detected by human from the corpus. We will use 50 test pairs instead of 1000 Since this translation detection process is very labor-intensive. This part will also require human evaluators since a single word could

have multiple translations such as in the case of a collocation. At the word translation stage, we would like to keep the word pairs belonging to a collocation as part of the translation list, to be piped into a noun phrase translation stage later. For this purpose, the human evaluation will need to be more guided by the corpus, reducing human errors. So for example, even though *Cross* and the Chinese word for *Tunnel* would not seem to be likely translations of each other, the human evaluator would see that they are both part of a collocation, namely the *Cross Harbor Tunnel.* We would like to keep *Cross, Tunnel* in the same set. Therefore they are acceptable as a matched pair. This part evaluates the efficiency of the system as a translator aid.

- In each of the above two cases, we will evaluate percentage correct of the translation candidate with the best correlation score, as well as the percentage correct of the candidates with the 10 highest score. In the latter scenario, a translation is *correct* if one of the three candidates is a correct translation.
- Another output list will include all mappings between all English words and Chinese words, sorted according to their scores. We will then count from the top pairs down, until a threshold. Percentage precision of this list will indicate the system performance in compiling a domain term dictionary.

4. Fourth, we will evaluate the performance of the system in translating noun phrases from English to Chinese/Japanese. The English part of the corpus will be tagged. The NP extractor will extract from the tagged corpus a list of noun phrases. We then apply our word translation program to constituents of each noun phrase to obtain groups of words from the other language. For each constituent word, the program will output ? top candidates of translated words. The translated words will be re-grouped in the order of their appearance in the corpus. We will perform two sets of evaluations: first by evaluating the precision correct of all candidate *open class words* compared to the English noun phrase, then by evaluating the recall of candidate words compared to the correct noun phrase translation in the other language.

   For example, if *House of Commerce* is translated into *La Chambre de Commerce*, both precision and recall are 100%. If it is translated into *Commerce*, the precision is 100% but recall is 50%. On the other hand if it is translated into *Etats-Unis gouvernment Chambre Commerce*, its precision would be 50% and its recall 100%.

5. Finally, we will divide different words and word groups into separate classes according to their domain-specification. This division will be done either manually or automatically. We will test our system on each class and show its performance accordingly. This will show how useful our system will be in translating domain terms relative to other more common nouns and noun phrases.

5.6 Timetable for the remaining work

The timetable for the remaining work is as follows. Some of the work items can be done in parallel with others.

Nov95 More evaluation of the context histogram length feature on complete noun and proper noun list in the non-parallel corpus. About two weeks.

Nov95 Study possible linguistic justifications for context length histogram. About one week.

Nov95 Obtain one more non-parallel corpus from LDC. This will be done in parallel with obtaining a parallel, tagged English/Chinese corpus for studying relations between English noun phrases and their translations in Chinese. This part will be mostly carried out by the project students.

Dec95 Implementation of word relation matrix. About two weeks.

Dec95 Project students finish finding bilingual phrasal relation statistical work.

Jan95 Evaluation of word relation matrix and other statistical correlation features and matching functions. About one week.

Jan96 ACL deadline, possible submission.

Jan96 Implementation of eigenvalue discriminant analysis. About one month.

Feb96 Integration of linguistic constraints from analysis of noun phrase translations into system. Start implementation of an integrated system. Individual feature (except context length histogram) analysis evaluation. System black-box evaluation. Submission of papers.

Mar96 Continue implementation, system evaluation. In parallel, start writing the thesis.

Apr96 Evaluation. System integration. Thesis writing.

May96 Evaluation. System integration. Thesis writing.

Jun96 Thesis writing. Defense.

July96 Thesis writing. Defense.

Aug96 Thesis writing. Defense.

Chapter 6

Summary of contributions

The main contributions of our work are in two areas, namely in the application issue of automatically translating domain terms from large corpora; and in the methodological issue of achieving this purpose through integrating techniques from statistics, information theory, pattern matching, signal processing with linguistic knowledge. One byproduct is that we have developed a statistical tool for tokenizing Chinese texts.

6.1 Domain word translation

We have introduced techniques for extracting bilingual lexicons of technical terms, domain-dependent terms and regional terms from large corpora. Most of these terms cannot be found in dictionaries. Human translators, not being experts in most technical fields, cannot produce their translations readily. Using automatic methods online to obtain such lexicons from large texts saves time and manual labor in hand coding. It can help human translators to translate technical materials faster. It is also an essential part of the lexicon for a fully automatic machine translation system. It allows portability of such machine translation systems into different domains.

6.1.1 Domain word translation from noisy parallel corpora

We have shown techniques to align noisy parallel corpora by segments, and to extract bilingual word lexicon from it.

Our algorithm bypasses the sentence alignment step to find a bilingual lexicon of nouns and proper nouns. Its output shows promise for compilation of domain-specific, technical and regional compounds terms. It has shown effectiveness in computing such a lexicon from texts with no sentence boundary information and with noise; fine-grain sentence alignment is not necessary for lexicon compilation as long as we have highly reliable anchor points. Compared to other word alignment algorithms, it does not need *a priori* information. It has also shown promise for finding noun phrases in English and compound nouns in Chinese.

Our algorithms eliminate the need to manually clean up large parallel corpora by deleting unparallel paragraphs and segments, as well as the need for precise, clear sentence boundary markers. This saves large amount of manual labor and time. Many OCR texts can therefore be used as inputs, increasing the amount of useful bilingual corpora for our application.

Part of our algorithm can also be used as a text-based alignment step. Using the initial anchor points and primary bilingual lexicon, large bilingual texts can be segment-aligned. Further EM-based word-alignment method can be used on this bootstrapping step to obtain a larger bilingual lexicon.

6.1.2 Domain word translation from non-parallel corpora

We have developed algorithms to find the correlation between a word and its translation in another language in non-parallel bilingual texts.

We have shown initial results of matching words with their translations in a English-Chinese non-parallel corpus by using context heterogeneity measures and context length histogram measures. We can use the result of matchings between such word features in non-parallel corpora for extracting bilingual word pairs.

Our algorithms get rid of the need to rely on parallel, translated texts in different domains for domain word translation. There are many more monolingual texts in various domains available. Therefore our techniques greatly increases the amount of text data usable as input. We plan to run the algorithms further on various domain-dependent texts in different languages, such as the AP newswire data in English and French, the Japanese financial news daily newspaper Nihon Kezai Shimbun with English Wall Street journal from approximately the same period.

6.2 Pattern matching and signal processing of word features

Another major contribution of our work is in using pattern matching and signal processing techniques for translation. Most other learning algorithms treat a word pair in translation as two statistical variables with some sort of probability conditioned on the context. We have found *lexical signature* of words which are independent of languages, consistent over texts and domain. Our work describe such lexical signature in terms of signals dynamically over text lengths. Each signature of the words has a particular physical *shape* or *space-frequency transformation*. Matching word shapes becomes a task not unlike that of matching a rotated or distorted object in image processing, or that of processing waveforms of different pronunciations of the same word in speech recognition. We have employed pattern matching and signal analysis methods from image and speech processing to lexical processing. The advantages of using this approach are described in the following sections.

6.2.1 Dynamic word signals

Dynamic word signals are discovered which are more robust than conventional word position statistics. In other statistical lexicon processing algorithms, the characteristic of a word is usually described by a single conditional probability ascribing the general likelihood of co-occurrence of a pair of words. The dynamic local behaviors of a word and its translation in their contexts are not described. We have derived word features which can be plotted on a two dimensional plan. These plots have signatural shapes which have characteristic space-frequency properties depending on the lexical property of the word. In a noisy parallel bilingual corpus, the *arrival distance* of a particular word is a warped version of that of its translation. The arrival distance is a dynamic feature along the length of the texts and is represented as recency vector. The recency vector of a word and its translation are warped, distorted versions of each other.

In a non-parallel corpus, we found that the histogram information of context segments of a word and that of its translation are closely matched. The lexical characteristics of a word can be described in a histogram signal. Furthermore, the salient features of the histogram signals can be emphasized and analyzed by space-frequency transformations. The histogram signals are not simply warped signals in one dimension; therefore we use their space-frequency transformations for pattern matching. The histogram signals of a word are consistent over texts in different languages in the same domain, even if they are not translated versions of each other.

6.2.2 Word feature vector

Word features are represented in vector forms for pattern matching. We have derived word features such as context heterogeneity and word relation matrix, in addition to two dimensional signals. These features are represented in individual vector form. They describe the lexical properties of words in terms of weighted frequencies or conditional probabilities. They were not previously found by other statistical lexical processing algorithms. Our discovery of these features add to our knowledge of lexical properties of words in different contexts, domains and languages. We use them in our domain term translation algorithm but they can also be used for other applications such as text classification, word sense disambiguation, collocation extraction, etc. For example, context heterogeneity describes the productivity of a word in a context. If a word is productive, it is found in context of many different types of words. The word 'the' can be followed by practical any noun. 'The' is a *right productive* word. On the other hand, in the Hansard corpus, *Chambre* is almost always followed by *de* as part of a collocation. The context heterogeneity of words are consistent with that of their translation in the same domain. These features can be used both as a clustering measure and a discrimination measure. Given two corresponding clusters of words from the corpus, these features could be used to further divide and refine the clusters into few candidate translation words for a given word. They can also be combined with the signal vector into a single vector for each word for eigenvalue classification.

6.2.3 Signal processing of word features

We have developed a new signal processing paradigm to match pairs of words which are translations of each other. Word features are expressed as signals in two dimensions. Dynamic Time Warping, a signal matching method commonly used for processing speech signals, is used to match these word signals. Moreover, space-frequency analysis, often employed in image processing, was used to transform the original word signal to extract salient visual features from the words. Dynamic Time Warping was again applied to the transformed signals for matching. This is the first time such a paradigm is introduced to statistical lexical processing where words are treated more as a continuous signal in the text-space, rather than discrete symbols with various attributes. The advantage of such an approach is that it treats the words *in context*, relative to other words in the context. We believe this method captures more robustly the domain-dependency of words. Transforming the word signals to a different space allow us to *discover* some linguistic features of these words which would otherwise take human a long time to speculate from looking at the texts. The signal processing and pattern matching techniques we used are rather straightforward. Once we can describe word features as signals, more signal processing and pattern matching tools become available—we have opened a new toolbox for lexical processing. Another major advantage of this transformation is described in the following section.

6.2.4 Combined statistical feature classification

Since we represent all the statistical features in vector forms, we are able to use standard discriminative analysis techniques in pattern recognition to automatically match the words according to the optimal classification criteria of the vectors, to their translations. The advantage of these techniques is they transform the statistical features into a smaller subspace, emphasizing the significant eigenvalues, for faster matching.

6.3 Cross language group bilingual processing

6.3.1 Language-robust algorithms

Our algorithms can be applied to languages with different language groups, character sets, syntactic structures. Most other bilingual lexicon compilation work has been done with languages of Indo-European origins. Since these languages share a common root to some extent, the various algorithms commonly take advantage of their morphological, syntactic and even semantic similarities. These algorithms cannot be applied to the problem of extracting bilingual knowledge between, say, Chinese and English because these two languages are too different. They differ greatly in character sets, sentence lengths, syntactic structures, expressiveness features, idioms, metaphors, etc.

We have successfully demonstrated the common features which exist between words and their translations *even* in a language pair like Chinese and English. We will test our algorithms further on Japanese and French to show their robustness and extendibility.

6.3.2 English noun phrases and their translations

In English, most technical terms are nouns and noun phrases. Our work concentrates on translating these terms into other languages. However, many English noun phrases do not translate into noun phrases in the other language. We systematically study the relationship between English noun phrases and their translations in Chinese, Japanese and French by using parallel corpora. This finding can be used to help English noun phrase translation from large corpora, including non-parallel corpora.

6.4 Linguistic knowledge

We use various linguistic knowledge as filter, preprocessor and augmenter to our statistical algorithms.

6.4.1 Noun phrase finder

We have studied the grammar of English noun phrases and implemented an English noun phrase bracketer, which is extendable to bracket other phrasal groups and to other languages. This tool has the option of taking tagged or untagged input corpus, and outputs noun phrase or other brackets according to a grammar template given.

6.4.2 Chinese segmentation

We have successfully used collocation techniques previously developed to apply to our problem of Chinese language processing. This is a prerequisite to our work on Asian/Indo-European domain term translation, since we have to define word boundaries in Chinese texts before any word translation can be considered.

We have demonstrated that statistically-based lexical acquisition on the same corpus being tokenized can significantly reduce error rates due to unknown words not found in a Chinese dictionary. Many domain-specific and regional words, names, titles, compounds, and idioms that were not found in a machine-readable dictionary were automatically extracted by our tool. We have also shown the effectiveness of simple linguistic filters to improve the precision of a statistical method for generating new Chinese lexical entries. Using linguistic knowledge to construct filters rather than generators has the advantage that applicability conditions do not need to be closely checked, since the training corpus presumably already adheres to any applicability conditions.

Chapter 7

Limitations and future directions

There are clearly limitations to our work. Many problems remain to be solved. Some are beyond the scope of this thesis work. We will discuss some of these in this chapter. Possible extension of applications are also proposed here as post-thesis work.

7.1 Limitations

The most important limitation is that our work does not concentrate on building a dictionary of terms, nor do we study the **semantics** of terms. We do not use the class of words according to dictionary definition, nor the relation between words according to thesaraus definitions. The domain term translation our system compiles is one particular mapping of bilingual terms among other possible mappings. The output of our system can be used as a translator-aid, or as part of a machine translation system, but not as a bilingual dictionary for semantic discrimination of any kind.

It is possible that more linguistically inclined research could be done using the information provided by dictionaries and thesauraus, either in the form of decision tree or as an additional part of the statistical system. Such work is, however, beyond the scope of this thesis work.

Another obvious limitation to our statistical algorithms is they cannot handle many **low frequency terms**. Low frequency terms give unreliable statistics and therefore unreliable word features. This is in fact the disadvantage of most statistical approaches. Linguistic knowledge can help increase the recall of our systems but will be too general to help many individual terms.

Another limitation is that the algorithms do not *output* all domain specific translations. The system does not extract only domain terms for translation in the first place. A more efficient system could use a domain term extraction and classifier to preprocess the corpora and then use our system to find matchings for these terms.

We have assumed that most domain specific terms are **noun phrases**. There are still others which are not nouns or noun phrases. For example, the word *programming* is a computer science term whose translation in Chinese or Japanese is quite consistent. If we do not filter non-nouns out, our system would have a higher recall, although it might also have a lower precision.

We also assumed that all domain specific terms have consistent, predominately one-to-one **mapping** to the same terms in another language. We have not studied the percentage of domain specific terms for which this is valid. How many of them have one-to-one mapping? How many one-to-two, one-to-many? Are there more one-to-many mappings in non-parallel corpora? Which class of words tend to have one-to-many mappings? How is domain-specification related to one-to-one-ness in mapping?

There are some domain specific **collocations** consisting of words separated by other words. Our system does not find these terms. It is conceivable to preprocess a text by a collocation extraction tool, such as Xtract (Smadja 1993) and apply our algorithms on the constituent words of these collocations. However, such collocations are even more complicated than simple noun phrases and if the language pairs do not have consistent one-to-one collocation mappings, more complex algorithms would need to be devised.

The **non-parallelness** of monolingual corpora can greatly affect the performance of our system. It would be useful in the future to compute the asymptotic behavior of the system performance with respect to the degree of mismatch between two same-domain monolingual texts.

We have used various tools to preprocess our corpora before applying the translation algorithms. The precision of these tools, including part-of-speech taggers, noun phrase finders, and Chinese and Japanese segmenters all affect the output of our algorithms. Further improvement on these tools, especially Chinese segmenters and noun phrase identifiers, is a research topic remain to be explored by others.

7.2 Other application: profiling text and domain characteristics for spoken language understanding

As a byproduct of finding text-dependent or domain-dependent word translations, we have developed a paradigm for profiling/describing text characteristics and domain characteristics in terms of their constituent words. It is conceivable that word features which are descriptive of their context and domain such as those used in our work can be applied to describe the text and domain they represent. Such domain profiling can be used in other natural language processing applications such as in an natural language understanding system for spoken language processing. Nowadays all spoken language understanding systems are trained on large corpora. The language modeling component of such systems relies heavily on statistical n-grams. However, whenever a new domain is encountered by the system, total re-training is needed by using large amount of data in the new domain. The old n-grams are discarded. By using domain profiling, useful part of the language model from the old domain can be kept, only a small amount of new data would be needed to adapt the system to the new domain according to the new domain characteristics. This would considerably reduce the amount of re-training needed, and add to the robustness of the spoken language understanding system.

Bibliography

Aho, A., B. Kernighan, & P. Weinberger. 1980. *The AWK programming language*. Addison-Wesley, Reading, Massachusetts, USA.

Bourigault, Didier. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of COLING 92*, 977–981.

Brown, P., J. Lai, & R. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*.

Brown, P.F., J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, & P. Roosin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85.

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

Chen, Stanley. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, 9–16, Columbus, Ohio.

Chiang, Tung-Hui, Jing-Shin Chang, Ming-Yu Lin, & Keh-Yih Su. 1992. Statistical models for word segmentation and unknown resolution. In *Proceedings of ROCLING-92*, 121–146.

Chui, Charles K. 1992. *Wavelet analysis and its applications*. Academic Press, Inc.

Church, K., I. Dagan, W. Gale, P. Fung, J. Helfman, & B. Satish. 1993. Aligning parallel texts: Do methods developed for English-French generalize to Asian languages? In *Proceedings of Pacific Asia Conference on Formal and Computational Linguistics*.

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, 136–143, Austin, Texas.

Church, Kenneth. 1993. Char_align: A program for aligning parallel texts at the character level. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, 1–8, Columbus, Ohio.

Dagan, Ido & Kenneth W. Church. 1994. Termight: Identifying and translating technical terminology. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, 34–40, Stuttgart, Germany.

Dagan, Ido, Kenneth W. Church, & William A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, 1–8, Columbus, Ohio.

Fukunaga, Keinosuke. 1990. *Statistical pattern recognition.* Compute Science and Scientific Computing. San Diego, California: Academic Press.

Fung, Pascale. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, 236–233, Boston, Massachusettes.

Fung, Pascale. 1996. Space-frequency analysis for domain word translation. In *Proceedings of ICASSP 96*, Atlanta, Georgia. To appear.

Fung, Pascale & Kenneth Church. 1994. Kvec: A new approach for aligning parallel texts. In *Proceedings of COLING 94*, 1096–1102, Kyoto, Japan.

Fung, Pascale & Kathleen McKeown. 1994. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, 81–88, Columbia, Maryland.

Fung, Pascale & Dekai Wu. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, 69–85, Kyoto, Japan.

Fung, Pascale & Dekai Wu. 1995. Coerced Markov Models for cross-lingual lexical-tag relations. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, 240–255, Leuven, Belgium.

Gale, William & Kenneth Church. 1991. Identifying word correspondences in parallel text. In *Proceedings of the Fourth Darpa Workshop on Speech and Natural Language*, Asilomar.

Gale, William A. & Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

hua Chen, Kuang & Hsin-Hsi Chen. 1994. Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation.

Huang, Chu Ren, 1989. *Mandarin chinese NP de - a comparative study of current grammatical theories.* Taipei, Taiwan: Institute of History and Philology Academia Sinica dissertation.

Kay, Martin & Martin Röscheisen. 1993. Text-Translation alignment. *Computational Linguistics*, 19(1):121–142.

Kupiec, Julian. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, 17–22, Columbus, Ohio.

Li, Charles & Sandra Thompson. 1989. *Mandarin Chinese - a functional reference grammar.* University of California Press.

Li, Wenjie, Haihua Pan, Ming Zhou, Kam-Fai Wong, & Vincent Lum. 1995. Corpus-based maximal-length Chinese noun phrase extraction. In *Proceedings of natural language processing pacific rim symposium 95.*

LI, YU DE. 1985. *Ke Ji Han Yu Yu Fa (Technical Chinese Grammar).* Xin Hua Shu Dian Publishers.

LIN, MING-YU, TUNG-HUI CHIANG, & KEH-YIH SU. 1993. A preliminary study on unknown word problem in Chinese word segmentation. In *Proceedings of ROCLING-93*, 119–141.

MATSUMOTO, YUJI & MAKOTO NAGAO. 1994. Improvements of japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, 22–28.

MOSTELLER, FREDERICK & DAVID L. WALLACE. 1984. *Applied bayesian and classical inference - the case of the federalist papers.* Springer Series in Satistics, Springer-Verlag.

NIE, JIAN-YUN, WANYING JIN, & MARIE-LOUISE HANNAN. 1994. A hybrid approach to unknown word detection and segmentation in chinese. In *Proceedings of the International Conference on Chinese Computing*, 326–335, Singapore.

RAUSCH, NORRBACK, & SVENSSON. 1992. Excerperting av nominalfraser ur löpande text. In *Ms., Stockhomlms Universiet, Institutionen för linfvistik.*

RIOUL, OLIVIER & MARTIN VETTERLI. 1991. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 14–38.

SIMARD, M., G FOSTER, & P. ISABELLE. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of the Forth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada.

SMADJA, FRANK. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

SMADJA, FRANK & KATHLEEN MCKEOWN. 1993. Translating collocations for use in bilingual lexicons. In *Proceedings of the ARPA Human Language Technology Workshop 94*, Plainsboro, New Jersey.

SPROAT, RICHARD, CHILIN SHIH, WILLIAM GALE, & N. CHANG, 1994. A stochastic word segmentation algorithm for a Mandarin text-to-speech system. Submitted to ACL-94.

STRANG, GILBERT. 1989. Wavelets and dilation equations: A brief introduction. *SIAM Review*, 31(4):614–627.

VOUTILAINEN, ATRO. 1993. NPtool: a detector of English noun phrases. In *Proceedings of Workshop on Very Large Corpora*, 48–57, Columbus, Ohio.

WANG, LIANG-JYH, TZUSHENG PEI, WEI-CHUAN LI, & LIH-CHING R. HUANG. 1991. A parsing method for identifying words in Mandarin Chinese sentences. In *Proceedings of IJCAI-91, Twelfth International Joint Conference on Artificial Intelligence*, volume 2, 1018–1023, Sydney.

WARWICK-ARMSTRONG, S. & G. RUSSELL. 1990. Bilingual concordancing and bilingual lexicography. *Euralex.*

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

WU, DEKAI & PASCALE FUNG. 1994. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, 180–181, Stuttgart, Germany.

WU, DEKAI & XUANYIN XIA. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, 206–213, Columbia, Maryland.