Pure

Bangor University

DOCTOR OF PHILOSOPHY

Evaluating Parallel Corpora and Translation Quality for Chinese and English

Liu, Wei

Award date:
2016

Awarding institution:
Bangor University

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. Apr. 2025

Evaluating Parallel Corpora and Translation Quality for Chinese and English

Wei Liu [1]

School of Computer Science, Bangor University

June, 2016

[1]Supervised by Dr. William J. Teahan

Acknowledgements

Sometimes it is the people who no one imagines anything of who do the things that no one can imagine.

—Alan Turing

Sometimes it is hard to express the gratitude felt for another, especially for my supervisor Dr. William J. Teahan. I acknowledge, with gratitude, my debt of thanks to him for encouraging my research and allowing me to grow as a research scientist. His advice on both research as well as on my career have been invaluable and very much appreciated.

I would also like to thank my loving wife Xuejiao Li for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. Without her wholehearted support, I would not be able to finish my PhD studies.

My parents in law—Yuxiang Tao and Xinfu Li and my parents—Junzhi Wu and Bingzeng Liu, receive my deepest gratitude and love for their dedication and the many years of support during my undergraduate, master and PhD studies. I am so grateful to have such kind parents in my life.

Last, but not least, I would express my very special thanks to my best friend Mr. David Jones. He encouraged me a lot to live optimistically and improved my English very much.

Abstract

Parallel bilingual corpora are important basic resources for statistical machine translation. Accurate alignment of textual elements (e.g. documents, paragraphs, sentences) in a parallel bilingual corpus is a crucial step for statistical machine translation. Rather than using sentence length, word co-occurrence, cognates, dictionaries or parts of speech, this thesis uses compression code lengths based on the Prediction by Partial Matching (PPM) compression algorithm to measure when two sentences are aligned for parallel Chinese-English corpora. PPM has been found to be an effective method as a measure of whether the information conveyed by the texts is similar at estimating the entropy of the text.

Evaluation of the quality of sentence alignment is a way to measure the quality of a corpus. Evaluating parallel bilingual corpora is also an important process and usually the last step for parallel bilingual corpus creation. However, most statistics of parallel bilingual corpora are based on counts of characters, words, tokens, sentences or files. As there is a lack of advanced parallel bilingual corpus evaluation methods, this thesis adopts a new PPM-based method for parallel bilingual corpus evaluation. The method has been used to evaluate the quality of three existing parallel bilingual corpora—the DC Corpus, the Hong Kong Yearbook Corpus and the UN Corpus.

The compression-based method has also been applied to the problem of the automatic creation of new parallel corpora. The quality of sentence alignment for automatically created parallel bilingual corpora is always lower than manually checked corpora. This thesis processed the Corpus of United Nations by using the PPM-based metric and sought the best code length

threshold value that can be used for automatically determining satisfactory or unsatisfactory sentence alignment in terms of translation quality in the corpus. The thesis also collected bilingual textual elements from the web and improved the quality based on the threshold code length ratio of 1.5.

The approach has also been adapted to use as a method to perform translation system evaluation by comparing the compression code lengths of back translations at the sentence level. Compared to Bilingual Evaluation Understudy (BLEU) scores, the back translation-based evaluation method was able to present differences at the sentence level between original sentences and their back translations more accurately when used to evaluate some common Chinese-English translation systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background & Motivation

A corpus is a collection of electronic texts, which is usually well-sampled and widely used for computational linguistics. Corpora as a basic resource for Corpus Linguistics have been developing for over five decades since the 1960s. The analysis of corpora in early research concentrated on word frequency statistics along with grammatical annotation (e.g. part-of-speech tagging, etc.).

Currently, there are abundant monolingual corpora due to building monolingual corpora has made brilliant achievements. However, obtaining a satisfactory bilingual parallel corpus is more difficult as they are usually not readily available. The main reasons are that they are more difficult to build and process than monolingual corpora and that it is also not easy to evaluate the quality of a parallel corpus before use. There is still a lack of high quality parallel corpora in terms of sentence alignment and/or they are expensive to purchase for small research teams or individual researchers. Parallel corpora is important for language learning, translation, statistical machine transla-

tion and many other fields.

Aligning is an important step for the creation of parallel corpora, which significantly improves the usability. A parallel corpus has usually been manually aligned as documents and a number of them have also been manually aligned at the paragraph level. However, due to the large amount of sentences, depending on the source, aligning parallel corpora at the sentence level becomes more expensive. Along with the increasing need of high quality parallel corpora, automatic sentence alignment for parallel corpora becomes more important. Nowadays, aligning parallel corpora at different levels such as in documents, paragraphs, sentences, phrases and words has attracted increasing attention from researchers. Modern aligned parallel corpora have improved in both aspects of theory and technology and have been more widely used in areas of language analysis, language teaching, lexicography and machine translation.

Parallel corpora are the essential resource for statistical machine translation. The alignment quality of a parallel corpus directly influences the performance of statistical machine translation. Therefore, improved methods for automatically creating a higher quality parallel corpus and evaluating the quality of a parallel corpus are both becoming more important, which is the motivation of this study.

1.2 Aim & Objectives

The primary aim of this study is to investigate the effectiveness of a novel method using Prediction by Partial Matching (PPM) compression for alignment in order to create and evaluate the quality of Chinese-English parallel corpora. Therefore, the objectives of this study are as follows:

- Compare and contrast whether PPM performs better than other common compression methods for compressing Chinese and English text (see section 3.4).
- Determine how well the PPM-based evaluation method works for aligning Chinese-English parallel corpora at the sentence level (see chapter 3).
- Examine whether PPM code length-based metrics perform better than traditional sentence length-based metrics (see chapters 3, 4 and 5).
- Evaluate the quality of Chinese-English parallel corpora by using the novel PPM compression code length metric (see chapter 4).
- Evaluate whether PPM-based compression method works well for automatic creating Chinese-English parallel corpora from the Internet (see chapter 5).
- Investigate the PPM-based evaluation method as a way for measuring and comparing common translation systems and determine whether PPM-based evaluation method works better than BLEU evaluation measurements (see chapter 6).

1.3 Contributions

The main contribution of this thesis is to propose effective novel methods for sentence alignment of Chinese-English parallel corpora and evaluation of sentence alignment. The methods are based on compression code length ratios and compression code length differences, which are calculated using the PPM compression algorithm directly on Chinese and English parallel

sentences. The specific contributions for this study and future work can be listed as follows:

- A new effective parallel corpus alignment method has been developed and evaluated.
- A novel PPM-based parallel corpus evaluation criteria has been proposed.
- The feasibility of automatic creation of Chinese-English parallel corpora has been investigated.
- An effective novel method for the evaluation of translations and translation systems has been developed.

1.4 Publications

Three conference papers based on this study have already been published and another two journal papers are being submitted for publication. Table 1.1 shows specific papers which relate to this study.

The first conference publication, entitled "Adaptive Compression-based Models of Chinese Text", describes adaptive models of Chinese text based on the PPM text compression scheme that learns the language as the text is processed sequentially. The paper describes several character-based, word-based and part-of-speech based variants of PPM that achieve significant improvements in compression rate over existing models. Results for Chinese text contrast that achieved for English text, with character-based models outperforming the word and PoS based models rather than the other way round. The paper also explores how well these models perform at the task of Chinese

Table 1.1: Publications that relate to this study.

1	**Title**	Adaptive Compression-based Models of Chinese Text
	Authors	William J. Teahan, Peiliang Wu and Wei Liu
	In	Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP)
	Year	2014
	Status	Published
2	**Title**	Experiments with a PPM Compression-based Method for English-Chinese Bilingual Sentence Alignment
	Authors	Wei Liu, Zhipeng Chang and William J. Teahan
	In	Proceedings of Lecture Notes in Artificial Intelligence (LNAI)
	Year	2014
	Status	Published
3	**Title**	A New Hybrid Metric for Verifying Parallel Corpora of Arabic/English
	Authors	Saad Alkahtani, Wei Liu, and William J. Teahan
	In	Proceedings of Computer Science & Information Technology (CS & IT)
	Year	2015
	Status	Published
4	**Title**	Experiments with a PPM Compression-based Method for English-Chinese Bilingual Sentence Alignment
	Authors	Wei Liu, William J. Teahan and Zhipeng Chang
	In	Journal of Computer Speech and Language (CSL)
	Year	2016
	Status	Pending
5	**Title**	A PPM-based Method for Evaluation and Generation of Bilingual Parallel Corpora
	Authors	William J. Teahan, Zhipeng Chang and Wei Liu
	In	—
	Year	2016
	Status	Pending

word segmentation. The paper was presented at the 16th International Conference on Audio, Language and Image Processing (ICALIP) 2014, held in Shanghai, China. The insight gained from this paper has been an important foundation for this thesis as cited in Chapter 2.

The second conference publication, entitled "Experiments with a PPM Compression-based Method for English-Chinese Bilingual Sentence Alignment", which is based upon Chapter 3, investigates compression-based methods for aligning sentences in an English-Chinese parallel corpus. Four metrics for matching sentences required for measuring the alignment at the sentence level are compared: the standard sentence length ratio, and three new metrics, absolute sentence length difference, compression code length ratio, and absolute compression code length difference. Initial experiments with code length ratio show that using the PPM compression scheme, a method that also performs well at many language modeling tasks, significantly outperforms the other standard compression algorithms Gzip and Bzip2. The paper then shows that for sentence alignment of a parallel corpus with ground truth judgments, the compression code length ratio using PPM always performs better than sentence length ratio and the difference measurements also work better than the ratio measurements. The paper was presented at the Second International Conference on Statistical Language and Speech Processing (SLSP) 2014, held in Grenoble, France and finally published in the Lecture Notes in Artificial Intelligence (LNAI) proceedings.

The third conference publication, entitled "Aligning a New Parallel Corpus of Arabic-English", which Chapter 5 is partially based upon, discusses a new metric that has been applied to evaluate the quality in translation between sentence pairs in parallel corpora of Arabic-English. This metric combines two techniques, one based on sentence length and the other based

on compression code length. Experiments on sample test parallel Arabic-English corpora indicate the combination of these two techniques improves accuracy of the identification of satisfactory and unsatisfactory sentence pairs compared to sentence length and compression code length alone. The unsatisfactory sentence alignments such as misalignments, mistranslations etc. were randomly sampled from automatically created corpora. The new method proposed in this research is effective at filtering noise and reducing mistranslations resulting in greatly improved quality. The paper was submitted to the Fifth International Conference on Computer Science, Engineering and Applications (ICCSEA) 2015, held in Dubai, United Arab Emirates. The main contribution to this paper was the implementation, and the experiments and data analysis.

One journal paper has been accepted for future corrections, which is an extension of the second conference paper. The extension involves Chinese-English parallel corpus evaluation as discussed in Chapter 5. The paper describes further evaluation experiments with the UN parallel corpus of Chinese-English text using the code length method which indicates that there are a significant number of erroneous and poor translations in the corpus and also that the method was an effective method for identifying these unsatisfactory translations. The paper has been accepted by the Journal of Computer Speech and Language (CSL) subject to further correction.

The second journal paper—"A PPM-based Method for Evaluation and Generation of Bilingual Parallel Corpora" is discussed partially in Chapter 5 and in detail in Chapter 6. The paper discusses that automatic evaluation is important for machine translation systems to achieve accurate translation results. Most evaluation methods developed recently are based on metrics of word orders and/or structural information of sentences. This paper describes

an alternative method based on the PPM compression scheme to evaluate results of machine translation. The method calculates the ratio of estimated cross entropies (compression code lengths) between the original sentences and translated sentences. Because the ratio reflects the comparison of cross entropies that the two sentences carry, a threshold can be set to eliminate misalignment in parallel corpora. Moreover, the method can be used to evaluate machine translation results. Using the compression code length ratio of original and back translated sentences, inaccurate translations can be eliminated. This method is not language dependent and can be integrated into most machine translation packages. Our evaluation experiments on machine translation between English and Chinese indicate that the PPM-based method can effectively exclude many inaccurate translations. The method can also be used to automatically generate parallel corpora. This paper will be submitted to a journal in the very near future.

1.5 Organisation of this Dissertation

This thesis is organised into seven chapters. Chapter one presents the background information of this research along with motivation and highlights the aim and objectives of this study. This chapter also indicates the research's contributions and published papers. The second chapter provides a literature review of relevant research. Chapter three compares and contrasts PPM with Gzip and Bzip2 compression schemes for sentence alignment and uses four metrics based on sentence length and code length measurements to align a test corpus. The results show that the performance of the code length measurements are competitive. The fourth chapter proposes a novel parallel corpus evaluation method based on code length measurement. Five exist-

ing parallel corpora are used for corpus evaluation. Chapter five describes a PPM-based method used to classify a partial UN Corpus into a high-quality corpus with translations judged to be satisfactory and a low-quality corpus with erroneous and poor translations. Then the chapter shows how a PPM-based method can be used to automatically create a new parallel corpus and discusses how the method performs. Chapter six uses the code length based method combined with back translations to evaluate existing Chinese-English translation systems. BLEU scores and code length ratios are the two methods that this chapter employs for translation system evaluation. The last chapter provides further discussion and the conclusion of this research, analyses limitations and discusses future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter introduces background technologies that have been investigated by this study and explains the origins and theoretical basis for Statistical Machine Translation (SMT), corpus linguistics, parallel corpus evaluation and alignment, Prediction by Partial Matching (PPM) compression scheme as well as text encodings. PPM is the compression-based method that the study uses and is also surveyed. This chapter will firstly talk about SMT and Corpus Linguistics and introduce the PPM methodology and its variants. Then the chapter will discuss parallel corpus alignment and evaluation. Thirdly, back translation and translation quality evaluation will also be presented. Finally, this chapter will discuss the different English and Chinese encodings.

2.2 Statistical Machine Translation

Statistical Machine Translation (SMT), which is a form of machine translation, is a data-driven nonrestrictive method (Koehn, 2010). The basic idea of SMT is to statistically process a large number of parallel corpora, to build statistical translation model, and then use this model to translate text from source language to target language.

2.2.1 Overview

SMT can be divided into bilingual and multilingual systems. A system that can translate original language to two or more languages is called multilingual SMT, which is usually a combination of bilingual SMTs (Hutchins and Somers, 1992). Currently, Google Translate is using SMT approaches for most languages (Google Translate, 2014). In recent years, Google has maintained an important position in the machine translation field and leads in the machine translation evaluation which was held by the US National Institute of Standards and Technology (NIST, 2012).

The primary task of SMT is to construct a reasonable statistical model which requires the design of methods for estimating various relevant parameters (Koehn, 2010). Early word-based SMT employed the Noisy Channel Model and used maximum-likelihood criterion for unsupervised training (Chiang, 2005). In recent years, phrase-based SMT has become more common and employs various methods such as the discriminative training method and supervised training (Koehn, 2004).

2.2.2 History

The history of machine translation is almost as long as the modern computer. As early as 1949, Warren Weaver proposed the basic idea of SMT based on Shannon's information theory (Brown et al., 1990). The first to propose workable SMT models was IBM Institute researchers (Koehn, 2010). Brown et al. (1993) proposed five word to word statistical models from simple to complex, which are named "IBM Model 1" to "IBM Model 5". These five models were all noisy channel models and the algorithms of parameter estimation were all based on maximum-likelihood estimation. However, due to the limitation of computing resources that were available at the time and the lack of parallel corpora, they were unable to achieve large-scale data machine translation. Followed by Brown et al. (1993), Vogel et al. (1996) presented an effective statistical model based on the hidden Markov model, which was considered a good substitution for IBM Model 2.

A summer seminar on MT in 1999 was organised at Johns Hopkins University. This resulted in the GIZA package, which implements IBM Model 1 to IBM Model 5 (Gao et al., 2015). Subsequently, Franz-Joseph Och optimised GIZA to accelerate the training speed, especially for Training of IBM Model of 3 to 5. Meanwhile, he also proposed a more complex Model 6 and the package he published was named GIZA++ (Koehn et al., 2007). Until now, GIZA++ is still the cornerstone of most machine translation systems. Currently, for training large-scale corpora, there have been a number of existing parallel GIZA++ versions (Och and Ney, 2003).

Although word-based approaches opened up the road for SMT, the performance is restricted due to that the modelling unit is too small and that the generative model has a poor adaptability. Therefore, researchers turned to phrase-based translation methods. Och and Ney (2002) proposed the phrase

translation method which was based on maximum entropy model and resulted in significantly improved SMT performance. The method performed far better than other methods in the next few years. Furthermore, Och (2003) also proposed modifying the optimisation criteria of maximum entropy method which directly optimised for objective evaluation criteria and resulted in the method of Minimum Error Rate Training widely in use today.

Another important invention which has promoted further development of SMT is the emergence of automatic translation evaluation methods. These methods have delivered objective translation evaluation criteria for translation results and thereby avoided manual evaluation which is tedious and expensive (Koehn, 2010). One of the important methods is BLEU evaluation although some researchers have noted that BLEU performed far worse than manual evaluation and is too sensitive for some small mistakes (Papineni et al., 2002). However, BLEU is still being used as the primary criterion (if not the only) for translation evaluation due to it being fast and easy to run (Lavie and Denkowski, 2009).

Moses is currently a well maintained open-source machine translation software, which is developed by researchers from the University of Edinburgh (Koehn et al., 2007). The release of Moses simplified the cumbersome and complex process of SMT.

2.2.3 Models

Noisy Channel Model

The Noisy Channel Model assumes that the source language sentence is obtained after a noisy channel coding from the target language sentence. If we know the nature of the source and channel, we can have the probability

that destination produces source, which indicates looking for the best translation (Brown et al., 1993). The probabilities of observable signal by a given source and source occurrence are respectively named Translation Model and Language Model (Koehn, 2010). In other words, the Translation Model is correspondence between source language and target language, whereas the Language Model reflects the nature of a language itself. The Translation Model is to ensure the meaning of translation and delivers accurate and intuitive translation and the Language Model ensures the fluency of translation and presents the literariness of translation.

In the Translation Models which were proposed by IBM, the translation probability is calculated by word alignment. So-called word alignment is to know which word or words from a target language sentence corresponds to the word(s) from the source language sentence.

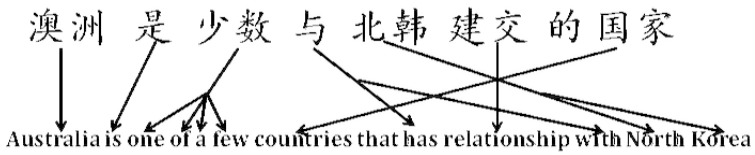


Figure 2.1: Word alignment between Chinese and English (Chiang, 2005).

Figure 2.1 shows that one word can be translated to one word, or two or more words or even nothing. Thus, the problem of obtaining translation probability is transformed to the problem of word alignment. The IBM series models, HMM and Model 6 are all word aligned parametric models. Their main difference lies in the number and type of model parameters. For example, the only parameter of IBM Model 1 is the probability of word translation, which is independent of the position of the word in the sentence (Moore, 2004). The translation probability is between words themselves rather than their positions. IBM Model 2 added a parameter which presents positions

of words in sentences and HMM model uses relative position to the previous word instead of absolute position, which was an improvement in terms of performance (Vogel et al., 1996). IBM Model 3, 4, 5 and 6 bring in the "Fertility Model" for presenting all probabilities that a word from source language is translated to different words in target language (Koehn, 2010).

For parameter estimation, the maximum-likelihood criterion can be used for unsupervised learning (Hofmann, 2001). Since there is no direct optimal solution, in practice, the expectation-maximisation (EM) algorithm is employed. According to the existing models, for each pair of sentences, the SMT model estimates probabilities for all (or the most likely) possible word alignments, counts weighted frequencies for all parameter values and finally normalises the results. For IBM Model 1 and 2, there is no Fertility Model employed, it is possible to obtain statistics for all possible word alignments by using simplified formulas, whereas for other models, it is difficult to traverse all word alignments (Koehn, 2010). One method is to use the Viterbi alignment algorithm to find the word alignment that has the highest probability. After the Viterbi alignment is obtained, the SMT model can either directly record relevant statistical results or record the statistical results after doing some further modifications (i.e. looking for neighbouring word alignments). IBM Model 3, 4, 5 and 6 are all using this method (Koehn, 2010).

It is rare that complete machine translation systems directly use the Noisy Channel Model method, but its by-product—word alignment—has become the cornerstone of various SMT systems (Chiang, 2005). Today, most systems still use GIZA++ to align words for parallel corpora (Koehn et al., 2007). Since an increasing size of parallel text resource has become available, speed of alignment has become an issue and parallel implementations such as MGIZA++ and PGIZA++ have been designed (Gao and Vogel, 2008).

Research is still ongoing with the Noisy Channel Model and word alignment. Although the alignment error rate of GIZA++ has been very low for all Indo-European languages, aligning between Indo-European languages and Arabic or Chinese are still not satisfactory due to high alignment error rates (Rama and Borin, 2011; Ravi and Knight, 2010; Riley and Gildea, 2010). Especially for Chinese, alignment error rates are usually over 30% (Riley and Gildea, 2010). Therefore, a lack of precise word alignment methods is one reason that Chinese machine translation is far behind other languages. Unsupervised alignment is still an important part although there have been a number of discriminative word alignment approaches (Dyer et al., 2011).

Optimisation Criteria

Finding the right optimisation criterion to best estimate model parameters from given training data is crucial (Koehn, 2010). In general, the training model parameters require a series of translated text and each source language sentence has one or more reference translations (Papineni et al., 2002).

In the early stages of research, discriminative training was based on maximum entropy criterion, which is simple and fast. However, a big problem was that there was no connection between sentence entropy and translation quality, so that optimising entropy in order to obtain a better translation result did not make much sense (Och and Ney, 2002). The Minimum Error Rate Training algorithm has been proposed due to that improving translation performance can be realised by optimising objective evaluation criterion such as BLEU, which is precision-based to evaluate how different the translation is compared to the original text (Callison-burch and Osborne, 2006).

Sequencing Model

Word orders among languages are very different, which is a difficulty overcome by above alignment methods and therefore a good reordering model is also necessary for discriminative training. The reordering model can be position-based, which describes the order probabilities of phrases from every pair of parallel sentences (Xiong et al., 2006). The reordering model can also be phrase-based such as Moses' reordering model, which describes whether every neighbour phrases' positions have been swapped by a given pair of phrases. Since the reality of reordering is far more complex than simply swapping positions and involves syntactic knowledge, the performance of reordering is still not satisfactory. Therefore, phrase reordering at present is still the key problem to be solved.

Decoding

Regardless of which model is used, during the actual translation process, decoding is always necessary (Koehn, 2010). The so-called decoding refers to the process of finding the translation result with the highest probability (or minimum cost) for the given model parameters for candicate sentences in source language. Similar to many sequence labelling problems such as Chinese word segmentation, decoding search can be applied by Branch and Bound algorithm, Heuristic Depth-first Search (A*) or the Viterbi algorithm (Narendra and Fukunaga, 1977; Crego and Mariño, 2006). In general, the search algorithm first constructs a search network which is a weighted finite state transducer to generate all possible translations, then searches for the optimal path on the search network (Koehn, 2010).

2.2.4 Basic Process of SMT

Similar to most machine learning methods, SMT has two stages—training and decoding, where the objective of training is to obtain model parameters and the decoding stage finds the optimal translation by using estimated parameters and given objective (Koehn, 2010). For phrase-based SMT, the definition of the training stage is not very clear. Strictly speaking, only the Minimum Error Rate Training step can be called training. However, in general, the steps of word alignment and phrase extraction have also been classified as the training phase.

Corpus Acquisition and Preprocessing

A parallel corpus is a large set of sentence pairs and each pair of sentences are translations to each other. A parallel corpus is an essential resource during preprocessing stage (Koehn, 2010). There are abundant parallel corpora which are downloadable from the Internet. An important method to improve the SMT system performance in a specific area is to find relevant target area corpora for training (i.e. law, finance, etc.).

Text preprocessing for is a necessary step for a parallel corpus after data has been obtained in order to standardise it for training (Koehn, 2010). For English, the preprocessing usually includes morpheme segmentation (i.e. let *'s* to be a word) and isolating words and symbols if they are connected (Creutz and Lagus, 2007). The main task for Chinese parallel corpora is word segmentation (Koehn, 2010). In addition, the preprocessing also involves to removal of some sentence pairs if they have siginificantly different sentence lengths or other abnormal symbols or characters (Xu et al., 2005).

After preprocessing, the main processing for constructing a parallel corpus is divided into three parts. The first part is word alignment and phrase

extraction, the second part is MERT and the third part is system evaluation (Brunning, 2010). For the data in the second and third parts, having multiple reference translations for source language sentence will be more beneficial.

Word Alignment

When using GIZA++ to align a parallel corpus, since GIZA++ is "one-way" word alignment, the alignment should be performed twice—from source language to target language and from target language to source language (Koehn et al., 2007). In general, GIZA++ requires alignments by HMM and IBM Models and depending on different sizes and iteration times for aligning parallel corpora, the training time can be very long (Och and Ney, 2003).

Phrase Extraction

The basic criteria for phrase extraction is that there must be at least one pair of words connected in each pair of parallel phrases and no word connected to another word which is outside the phrase (Koehn, 2010). The Moses package contains a phrase extraction program and the extraction result will occupy a lot of disk space (Koehn et al., 2007). The feature extraction is carried out after phrase extraction is completed, which is to calculate probabilities of phrase translations and word translations (Lewis, 1992).

Minimum Error Rate Training

Minimum Error Rate Training (MERT) optimises the training feature weight based on tuning set which is the a part of prepared data and generates the best optimisation criterion (Och, 2003). The common optimisation criteria include entropy, BLEU, Translation Edit Rate (TER), etc. (Snover et al.,

2006). The MERT stage requires a decoder that does multiple decoding for optimisation set, generates top N highest score results and adjusts feature weights (Galley and Quirk, 2011). When the feature weights are modified, the sort of N results is accordingly changed. Then the result with the highest score is the final decoding result, which will be used to calculate the BLEU score or TER. When a new set of weights are obtained, the scoring of optimisation set will be improved and in the next round re-decoding will be applied. This process will be continued until no more improvement can be observed. The N value selected affects the size of optimisation set, the model size, decoding speed and training runtime which can be from hours to days (Galley and Quirk, 2011).

Decoding and System Evaluation

Decoding can be applied based on the weights that are obtained by MERT (Koehn, 2010). System performance can usually be objectively evaluated based on the test set or even subjectively evaluated based on objective evaluation (Galley and Quirk, 2011).

2.2.5 Difficulties and Research Directions

The difficulty of SMT lies in that the information the model contains is usually low in representing the syntax and semantics of natural language, so more problems will be experienced when processing due to greater syntactic difference between languages like Chinese-English (Wu and Fung, 2009). Therefore, the lack of readability of translation results is still a problem although all words have been "correctly" translated. We can say that there is still big performance potential for SMT. Meanwhile, good performance of SMT acquiring on a huge corpus. With increasing corpus resources and in-

creasingly complex algorithms, a more powerful computer is required. For a long time, Google's leading position in the field of machine translation has benefited from their powerful distributed computation ability (Uszkoreit and Brants, 2008). With the popularity of distributed computing, the parallelism technology of SMT is another research hotspot (Koehn, 2010).

The performance of machine translation also depends on translation evaluation criteria, which is ultimately linked to subjective judgements (Koehn, 2010). In general, the improvement of translation evaluation criteria is positively effecting machine translation performance. The evaluation of translations is a difficult area and not easier than machine translation.

Machine translation eliminates the barriers between different languages and is benefiting mankind in communications via modern technologies. However, the quality of machine translation has been a problem for a long time and the goal is still far from ideal (Church and Hovy, 1993). However, Kay (1997) has pointed out that translation is a fine and exacting art, the productivity of translation would be improved and become more rewarding and more exciting with the development of computer technology.

2.3 Corpus Linguistics

Corpus Linguistics is a language research based on instances of language use and analyses grammar and syntax of natural languages (McEnery and Wilson, 2004). Corpus Linguistics also studies relationships among different languages and an increasing mount of work have been carried out on the building of the corpora which contain texts in two or more languages. Many corpora have been created by humans, but now they are mostly automatically generated by computers.

Corpus linguists believe that reliable language analysis has to be established based on natural language context and minimal interference (McEnery and Wilson, 2004). In Corpus Linguistics, an issue has been the problem of corpus tokenisation. Sinclair (2004) advocated a minimum annotation so that texts can "speak for themselves", whereas McEnery and Wilson (2004) encouraged more tokens in corpora and believed that corpora become to "repositories of linguistic information" after annotated.

The British National Corpus (BNC) is a major milestone in corpus linguistics (The British National Corpus, 2007). The BNC is a collection of written and spoken English with 100 million words from a wide range of sources and free to download. The corpus represents British English in a wide cross-section of the later part of the 20th century. The latest XML edition was released in 2007, which contains 4049 texts and occupies about 5.2Gb.

Another milestone of modern Corpus Linguistics is the book "Computational Analysis of Present-Day American English", which was published by Francis and Kučera (1967) and analysed the Brown corpus. The Brown corpus is a well-compiled American English corpus and contains over one million tokens (Francis, 1965). Another key publication was "Towards a Description of English Usage", which was published by Quirk (1960) and introduced the Survey of English Usage project. Shortly thereafter, the Boston publisher Houghton Mifflin invited Kučera to do lexicography for "American Heritage English Dictionary" with one million words and three-line citations (Mifflin, 2000). "America Heritage English Dictionary" innovatively combined prescriptive elements (how to use the language) and descriptive elements (how has the language been used).

Other publications followed suit. The COBUILD monolingual learner's

dictionary of British publisher Collins is designed for non-native English speakers learning English and uses the Bank of English corpus (Sinclair, 1987). The corpus of English Usage Survey is used in "A comprehensive Grammar of the English language" which was compiled by Quirk et al. (1985).

The Brown corpus has spawned a number of similar corpora, such as Lancaster-Oslo-Bergen Corpus of British English (Johansson et al., 1978), Kolhapur Corpus of Indian English (Shastri, 1988), Wellington Corpus of New Zealand English (Bauer, 1993), Australian Corpus of English (Collins and Peters, 1988), Frown Corpus of American English (Hundt et al., 1999) and FLOB Corpus of British English (Hundt et al., 1998). Other corpora such as International Corpus of English (ICE) and British National Corpus (BNC) which were created by publishers, where BNC contains a collection of 100 million words (Greenbaum, 1991; Leech, 1992). For contemporary American English, there have been American National Corpus and the Corpus of Contemporary American English (Macleod et al., 2002; Davies, 2010). The Corpus of Contemporary American English was created in 1990, contains over 400 million tokens and is accessible from the Internet (Davies, 2010).

The Lancaster Corpus of Mandarin Chinese (LCMC) is a well-complied Simplified Chinese corpus, which contains one million words and was designed as a Chinese match of the Freiburg-LOB Corpus of British English (F-LOB) (McEnery and Xiao, 2004). The creation of the LCMC corpus was for research on contrasting tense and aspects between Chinese and English. The LCMC corpus is a valuable resource for research into Chinese and a reliable basis for contrastive study of English and Chinese.

The first machine transcription spoken language corpus was built by Montreal French Project in 1971 and contains 100 million tokens (Poplack, 1989).

The Montreal French Project also inspired Shana Poplack to establish a larger corpus of spoken French in the Ottawa-Hull area.

In addition to the collection of modern languages, there are several corpora which collect ancient languages. For example, Andersen-Forbes database of the Hebrew Bible was created in the 1970s and used up to seven tier syntactic structures for all clauses' grammatical analyses (Miller, 2010). The Quranic Arabic Corpus is the annotated corpus of classical Arabic Koran (Dukes and Habash, 2010). The corpus contains multiple tagging tiers including morphological segmentation, part-of-speech tagging and syntactic analyses based on dependency grammar.

2.3.1 Types of Corpora

A corpus can be divided into four different types (McEnery and Wilson, 2004):

- Heterogeneous Corpora: This type of corpus has no specific collection principle and extensively collects from various sources.
- Homogeneous Corpora: Only the same category of language text is collected for this type of corpus.
- Systematic Corpora: Language text is collected according to given principles and composition ratios to make corpora balanced and systematic to represent a range of language features.
- Specialised Corpora: Only language text which is used for a particular purpose is collected for this type of corpus.

Corpora can also be divided into monolingual corpora, bilingual corpora and multilingual corpora. A parallel corpus is a general designation for bilin-

gual corpora and multilingual corpora which consist of a large number of corresponding text alignments in different languages. Modern parallel corpora with different alignment levels (i.e. document, paragraph, sentence, phrase and word) are used in the fields of Natural Language Processing (NLP), Machines Translation (MT), Machine Learning (ML) and Artificial Intelligence (AI).

Chinese is the world's most populous language whilst English is the world's most widespread language in use. There have been a number of parallel corpora in Chinese and English produced in recent decades. Some of these are as follows:

- The Corpus of United Nations: Six-language parallel corpus collected from documents of the United Nations.
- The Babel English-Chinese Parallel Corpus: Created for the research project *Contrasting English and Chinese* at Lancaster University
- Hong Kong Yearbook: The Hong Kong government has been annually publishing its Hong Kong Yearbook since 1997, which has parallel text in Simplified Chinese, Traditional Chinese and English.
- DC Corpus: Resources have been collected from free online dictionaries (Yahoo, Google, Microsoft, Dict.CN and Baidu).

Table 2.1 gives more details such as size in words and alignment level about the common Chinese-English parallel corpora. The UN Corpus is much larger than the others and therefore it is usually cut into partitions for different uses such as full-text search, SMT training, etc. To measure the size of a corpus, word count is usually used for English. However, as Chinese is not naturally segmented, so word counting for unsegmented Chinese is

Table 2.1: Details of common Chinese-English parallel corpora.

Corpus Name	**Alignment Level**	**English Words**	**Chinese Words**
Corpus of United Nations	Word	364.5M	329.3M
Hong Kong Yearbook Corpus (2007–2014)	Document	1.4M	1.8M
Babel English-Chinese Parallel Corpus	Sentence	254K	287K
DC Corpus	Sentence	725K	858K

more difficult and not as accurate as other naturally segmented language, like English. It is common to see that Chinese text is counted by characters rather than words. Table 2.1 uses an approximate Chinese word count calculation in order to compare an estimate of the Chinese word count with the English word count for different corpora in the table.

Corpus creation has been ongoing since the first corpus was created. Nowadays, an increasing number of Chinese-English parallel corpora are being created and many of them are free to use. OPUS, a project of Uppsala University, is a growing collection of translated text from the web (Tiedemann and Nygaard, 2004; Tiedemann, 2009). There are more parallel corpora available with different alignment levels and qualities. Table 2.2 shows a number of corpora which were automatically collected from the Internet by OPUS without manual adjustment.

Parallel corpus creation is usually more complicated than creating a monolingual corpus. There are usually three ways for parallel corpus creation which are manually creation, automatic creation and hybrid creation. The advantages of manually creating a parallel corpus are obviously in higher quality and greater accuracy. All tokens have been manually checked for correct alignment and correspond between source and target languages. How-

Table 2.2: Open source Chinese-English parallel corpora from OPUS

Corpus	**Documents**	**Sentences**	**Tokens**
MultiUN	67167	9.6M	259.7M
Open Subtitles 2015	9483	8.4M	74.3M
Open Subtitles 2013	1518	1.4M	12.2M
Open Subtitles 2012	845	0.7M	6.4M
Tonsil	30	0.2M	4.9M
Open Subtitles 2011	714	0.6M	4.7M
UN	1	74.1K	3.1M
TED 2013	1	0.2M	3.0M
News-Commentary	1	50.7K	2.4M
KDE4	1437	0.1M	1.1M
PHP	3274	41.7K	0.6M
SPC	1	2.2K	130.8K
Ubuntu	445	6.9K	38.1K

ever, manually created parallel corpora are expensive in terms of time and effort, which depends very much on the availability of linguists. In comparison with manually created parallel corpora, automatically created parallel corpora cost less and are created more quickly. A program can automatically obtain language resources from the web or database. The disadvantages of automatically created parallel corpora, however, are in lower quality overall and also result in many other problems during use. Therefore, a hybrid way of parallel corpus creation counteracts the weakness of automatically created parallel corpora. The hybrid parallel corpus creation usually creates a parallel corpus automatically, and which is then manually checked and corrected in order to generate a higher quality parallel corpus at lower costs.

In terms of academic value, a parallel corpus with phrase and word alignment is usually more valuable than sentence aligned corpora. In fact, most existing parallel corpora are automatically created and manually corrected

and are usually manually aligned at the sentence level. There is a large part of the Corpus of United Nations (the UN Corpus), which has been manually aligned at word level that includes over 600 million Chinese and English words as has been shown in Table 2.1.

Modern parallel corpora are widely used in natural language processing and are crucial resources for statistical machine translation. However, there are still two difficulties, which are still open to research—how to create a high quality Chinese-English parallel corpus in terms of sentence alignment and how to evaluate the sentence alignment quality of a Chinese-English parallel corpus.

2.4 Parallel Corpus Creation & Evaluation

As parallel corpus creation has been more widely used, evaluating the quality became essential for automatically created parallel corpora. The quality of a parallel corpus is usually determined by the correctness of translations (Kaalep and Veskis, 2007). There is still no way to automatically create a parallel corpus with the correctness of 100% although there are an increasing number of automatic parallel corpus creation methods have been proposed. Therefore, evaluating corpora is an important way to determine their quality.

There have been a number of methods which were designed for measuring translation quality in a parallel corpus. An important issue for the improvement of a parallel corpus is to remove duplicated text. Bilingual Evaluation Understudy (BLEU) is a method for duplicate detection for corpora (Papineni et al., 2002). The basic idea of the BLEU method is to compare how different the pair of sentences are with the scale from 0 to 1, where 1 means

the exactly same sentences and 0 indicates that the two sentences are totally different.

In addition, Alkahtani et al. (2014) have introduced a new hybrid metric for verifying Arabic-English parallel corpora, which combines two techniques—one based on sentence length and the other based on compression code length, where the compression code length calculation was using the Prediction by Partial Matching (PPM) which will be introduced in detail in Section 2.6. The hybrid metric used a distance metric based on both techniques in order to check the quality of the sentence pairs. The experiments on sample test parallel Arabic-English corpora indicated that the hybrid metric improved accuracy of the identification of satisfactory and unsatisfactory sentence pairs compared to sentence length and compression code length alone and a threshold mechanism was involved to filter out unsatisfactory translations when either the sentence length ratio or compression code length ratio values have been exceeded. 100% accuracy for the test corpus with 12,000 Arabic-English translations was achieved using threshold values 2.50 and higher for sentence length ratio combined with 2.25 and higher for compression code length ratio. The hybrid metric has been proven to be effective at filtering noise and reducing mis-translations resulting in greatly improved parallel corpus quality (Alkahtani et al., 2014). The insight from this research has helped guide the research described in this dissertation. Although the research here has focused on sentence alignment, further work will be able to focus on evaluating the quality of the alignment in a pipeline—from the document, paragraph and sentence levels, down to a finer granularity such as phrase and word.

2.5 Parallel Corpus Alignment

Parallel corpus is important resource for natural language processing and Statistical Machine Translation and the main objects that this study uses (Tian et al., 2014). A parallel corpus can be aligned at the document levels, paragraph levels, sentence levels, phrase levels, word levels, etc. Alignment at the document and paragraph levels and finding document and paragraph boundaries are still difficult tasks when the input parallel text is noisy (Church, 1993).

2.5.1 Sentence Alignment

Aligning sentences is essential for building large bilingual parallel corpora. Sentence alignment for parallel corpora has been a problem and received a lot of attention since the 1990s (Lamraoui and Langlais, 2013). Sentence alignment is an important step for Statistical Machine Translation (Sennrich and Volk, 2010). There have been three main sentence alignment approaches, which are sentence length-based, dictionary- or translation-based and partial similarity-based (Varga et al., 2005). Ma (2006) introduced Champollion, which is a lexicon-based sentence aligner. By assigning greater weights to less frequent translated words, Champollion can increase the robustness of sentence alignments (Ma, 2006). Brown et al. (1991) have been using statistical technique for aligning sentences for parallel corpora with no use of lexical details of the sentences in the 1990s. Bannard and Callison-Burch (2005) proposed using paraphrases as a pivot in another language. They defined the probabilities of paraphrases and ranked them so that they can show how to take contextual information into account (Bannard and Callison-Burch, 2005). Véronis and Langlais (2000) described the ARCADE project, which

concerns with sentence and word alignment evaluation of French-English bilingual texts. The project evaluated twelve sentence alignment systems and five word alignment systems, and revealed that sentence alignment accuracy for "normal" texts was satisfactory (over 98.5%) and degraded sharply for "imperfect" texts (Véronis and Langlais, 2000).

There are two main kinds of approaches for sentence alignment: lexical-based and statistical-based (Wu, 1994). Aligning sentences can be divided into two steps, which are sentence segmentation and matching parallel sentence pairs ("1:1", "1:N" or "N:1" types, where N≥2).

The methods of matching sentences are various. Gale and Church (1993) presented a program for aligning sentences in bilingual corpora, which was based on sentence length comparison and the distance measure from English, French and German trilingual text. They also calculated the length of sentences by calculating the number of words in each sentence. This generated similar results—96 to 97%. Wu (1994) also proposed aligning English-Chinese corpus by determining sentence length and gave a high accuracy that reached to higher than 95%. This thesis is based on previous research (Alkahtani et al., 2014) and proposes new methods that are PPM-based that give competitive results.

The main problem for sentence alignment is to recognise the type of each matching case—one to one (1:1), one to many (1:N) or many to one (N:1), where N≥2. Ordinarily, a normal sentence aligned Chinese-English parallel corpus includes 1:1, 1:N, and N:1 cases, where most of them are 1:1 and the second most cases are 1:2 and 2:1 (Braune and Fraser, 2010). N:0 and 0:N (N≥1) cases are also possible when the parallel text is not perfectly aligned (Véronis and Langlais, 2000). As one 2:0 case can be regarded as two 1:0 cases, only 1:0 and 0:1 are considered in parallel corpus sentence

alignment research.

Sentence length-based metrics are widely used for aligning sentence from bilingual parallel corpora. Gale and Church (1993) described a method based on sentence length for aligning English-French parallel corpora. In addition, Alkahtani (2015) proposed a hybrid sentence alignment method based on PPM compression code length and standard sentence length metrics and achieved an accuracy of over 96%.

Kay and Röscheisen (1993) used the dice co-efficient to calculate the probabilities of words in one language being aligned with words in the other language. Simard et al. (1992) pursued a cognate based approach to sentence alignment after analysing the errors produced in length-based alignment (ibid., page 70). While they found that cognates alone cannot produce better alignments than length differences, a two-pass program, whereby strong alignments based on sentence length are made in the first pass, and cognates are used to align the more difficult sentences in the second pass, did produce better results than the simple length-based alignment. Haruno and Yamazaki (1996) use both probabilistic and a bilingual dictionary to find word cognates to help align sentences. Like Kay and Röscheisen (1993), this is a combined sentence and word alignment program. Haruno and Yamazaki (1996) do not make use of length-based techniques because they state that these methods do not work for such structurally different languages as English and Japanese.

Papageorgiou et al. (1994) have devised a sentence alignment scheme that matches sentences on the basis of the highest matching part of speech tags, the matches restricted to content words—nouns, adjectives and verbs. With 99% accuracy, they obtained the best results of all for sentence alignment algorithms. Melamed (2000) (ibid., page 5) however points out that "*It is*

difficult to compare this algorithm's performance to that of other algorithms in the literature, because results were only reported for a relatively easy bitext."

Ma (2006) proposed a lexicon-based sentence aligner—Champollion, which achieves high precision and recall rates on noisy data. Braune and Fraser (2010) addressed a novel sentence alignment approach for both asymmetrical and symmetrical parallel corpora. YASA as a well-performed corpus alignment system uses two-step processing parallel data—cognates for delimiting search space and aligning on the reduced space (Lamraoui and Langlais, 2013). Furthermore, BLEU describes similarity of two sentences, which has been used with length-based heuristics for sentence alignment (Sennrich and Volk, 2010). Véronis and Langlais (2000) have pointed out that the accuracy of sentence alignment degrades sharply for the parallel data that are not perfectly matched. (This was one motivation for us to investigate a novel idea for sentence alignment of parallel data—using the compression-based technique adopted for this study). Another novel approach using a cache needs more memory but can significantly improve computation time.

The next section will review the PPM algorithm which is the cornerstone of the approach to sentence alignment, parallel corpora evaluation and machine translation system evaluation adopted for the research in this dissertation.

2.6 Prediction by Partial Matching (PPM)

Prediction by Partial Matching (PPM) is an adaptive online compression scheme that predicts the next symbol or character based on a prior context with fixed length. PPM is a character n-gram approach that uses a back-off mechanism similar to that proposed by Katz (1987). However, in PPM

literature, backing-off is referred to as "escaping" and was invented prior to Katz's work. Cleary and Witten (1984) proposed PPM first using the variants of PPMA and PPMB. Then PPMC and PPMD were developed by Moffat in 1990 and Howard in 1993 (Wu, 2007). The main difference between PPMA, PPMB, PPMC and PPMD is the calculation of the escape probability which is needed by the smoothing mechanism used by the algorithm for backing off to lower order models. Experiments show that PPMD in most cases performs better than PPMA, PPMB and PPMC. PPM-based methods have been widely used in natural language processing, including evaluation of text collections which ensures whether the collection is valid or consistent (Khmelev and Teahan, 2003).

Formally, the probability p of the next symbol φ for PPMD is calculated using the following formula:

$$p(\varphi) = \frac{2c_d(\varphi) - 1}{2T_d}$$

where d denotes the current coding order, $c_d(\varphi)$ denotes the number of times that the symbol φ in the current context and T_d presents the total number of times that the current context has occurred. The calculation of the *escape* probability e by PPMD is as follows:

$$e = \frac{t_d}{2T_d}$$

where t_d is the total number of unique symbols that occur after the current context. When PPMD is encoding the upcoming symbol, it always starts first from the maximum order model. A maximum order of 5 is usually used in most of the experiments (Teahan et al., 2000) and order 5 has also been found effective for Chinese text (Wu, 2007). If the model contains the prediction for the upcoming symbol, it will be transmitted according to the order 5 distribution. If the model does not contain the prediction, the

encoder will escape down to order 4. The escape process will repeat until a model is found that is able to predict the upcoming symbol, backing off if needed to a default order -1 model where symbols are equiprobable (Teahan et al., 2000).

PPM code length is the size (in bytes) of the PPM-compressed output file. When using PPM as a natural language processing tool to compress text, the code length can be used to estimate the cross-entropy of the text. The cross-entropy can be calculated by the following formula:

$$H(S) = -\frac{1}{n}\log_2 p(S) = -\frac{1}{n}\sum_{i=1}^{n} -\log_2 p(x_i|x_1 \ldots x_{n-1})$$

where $H(S)$ is the average number of bits to encode the text. PPMD with order 5 is used for English and order 6 for Chinese in subsequent chapters for compressing the byte sequence of the text. For English, a single ASCII byte represents a single English character, whereas for GB-encoded Chinese text, a Chinese character is denoted by two bytes (and therefore 5 bytes will span 2.5 characters). Text compression experiments with Chinese text (Wu, 2007) show that compressing the byte or character sequence is noticeably better than when using the word sequence, which may reflect that characters in Chinese have greater meaning associated with them. We also wish to avoid the problem of word segmentation for Chinese text, hence the reason for using bytes for the experiments described in subsequent chapters.

PPM technique is a finite-context statistical and predicts the next symbol based on a finite number of preceding symbols. It blends the probability estimates for contexts by the escape mechanism (say back-off technique). The PPM scheme can be used to provide better compression results than many other methods such as Ziv-Lempel (LZ) methods which are dictionary based (Teahan, 1995). However, the advantages of PPM algorithm can be

applied to many tasks in Natural Language Processing.

The lengths of prior characters that PPM models are based upon are various. Mostly, the max length is fixed for the whole process. Teahan (1995) has given an example for a possible particular context that was "thei". All the letters that are following "thei" are counted, so that the counts can be used to estimate the probability for the upcoming character when "thei" occurs. PPM combines all context models for the varying lengths to estimate an overall probability distribution for prediction. Finally, arithmetic coding optimally encodes the character, which occurs with respect to this distribution (Teahan, 1995).

Compression using PPM models provides excellent compression rates, but is worse in terms of execution speed (Wu, 2007). There have been a series of variants, such as PPMA, PPMB, PPMC, PPMD, PPM*, PPMII, PPM-ch, PPMO, PPMT and PPMZ (Teahan and Harper, 2001). The performance of an algorithm of PPM is dependent on the estimated escape probability (Chang, 2008). However, it is difficult to make an optimal choice for estimating a probability (Cleary and Witten, 1984). The main difference among those PPM variants is the escape methods they employ (Wu, 2007).

In summary, PPM has potential to be applied to areas of NLP such as statistical machine translation, because the algorithm achieves accurate prediction, which is fundamental to statistical-based NLP and machine translation.

2.6.1 Variants of PPM

The variants of PPM that were introduced in the last section are distinguished by the calculations of symbol and escape probabilities for the context model (Wu, 2007). Formally, we make some definitions as follows (Wu,

2007):

- A: the size of the discrete alphabet consisting of symbols ($|A| > 2$);
- D: the maximum order of the model;
- d: the current coding order of a model ($d \leq D$);
- φ: an upcoming symbol ($\varphi = x_{n+1} \in A$);
- s_d: the current context $s_d = x_n, \ldots, x_{n-d+1}$;
- $c_d(\varphi)$: the number of times that the symbol φ in the context s_d;
- t_d: the total number of unique symbols that occur after the context s_d;
- T_d: the total number of times that the context s_d has been $T_d = \sum c_d(\varphi)$.

According to the above definitions, we can list different calculations of escape and symbol probabilities for the major PPM variants as shown in Table 2.3:

Table 2.3: Comparison of different escape calculations and symbol probabilities among PPMA, PPMB, PPMC and PPMD (Wu, 2007).

PPM Variants	**Escape Probability**	**Symbol Probability**
PPMA	$e = \frac{1}{T_d+1}$	$p(\varphi) = \frac{c(\varphi)}{T_d+1}$
PPMB	$e = \frac{t_d}{T_d}$	$p(\varphi) = \frac{c(\varphi)-1}{T_d}$
PPMC	$e = \frac{t_d}{T_d+t_d}$	$p(\varphi) = \frac{c(\varphi)}{T_d+t_d}$
PPMD	$e = \frac{t_d}{2T_d}$	$p(\varphi) = \frac{2c(\varphi)-1}{2T_d}$

where PPMA and PPMB were proposed by Cleary and Witten in 1984 (Cleary and Witten, 1984), PPMC was developed by Moffat in 1990 and PPMD was introduced by Howard in 1993 (Wu, 2007).

For example, a context might have occurred 10 times with symbol a following it 4 times, symbol b following it 3 times, symbol c following it twice and symbol d following it once. After calculation, the e (escape probability) for this context for major PPM variants are $\frac{1}{11}$, $\frac{4}{10}$, $\frac{4}{14}$ and $\frac{4}{20}$. Experiments show that PPMC is normally better than PPMA and PPMB and that PPMD performs a little bit better than PPMC (Wu, 2007). Therefore, PPMC and PPMD are two well performed variants.

Table 2.4 shows an example of how PPM technique works. The example uses PPMD calculation method to process the string—*tobeornottobe* with the maximum model order of 2. The symbol c followed "Prediction" is count, p indicates probability and $|A|$ means the alphabet size (Teahan et al., 2000). Suppose following the string, the upcoming character is o, which has been seen in order 2 model ($be \rightarrow o$) with count 1 and probability 1/2. The symbol o will be encoded in 1 bit. However, if the upcoming symbol were t instead of o, which would not be seen in order 2 context. The escape ($be \rightarrow esc$) event would be coded with probability of 1/2 and the context would be truncated to the order 1 context. Similarly, there is no t follows e in order 1 context. Consequently another escape would be coded with probability of 1/2. Then the context would be truncated again to order 0 (null context). Finally, t would be encoded in order 0 because it is seen in this context with probability of 5/26. The three probabilities 1/2, 1/2 and 5/26 would be amounted to over 5 bits in this case. Furthermore, if the upcoming character is a novel one, say x, the escape event will be coded three times from order 2 to order 0 with probabilities 1/2, 1/2, 3/13, then x is encoded at order -1. Assuming that the alphabet size is 256 for pure English, the probability is 1/256 and the total encoded size is just over 10 bits (Teahan et al., 2000).

There are further PPM variants, such as PPM*, which was introduced

Table 2.4: PPM model after processing the string *tobeornottobe*; c = count, p = prediction probability (Teahan et al., 2000).

Order 2					**Order 1**					**Order 0**			
Prediction			c	p	Prediction			c	p	Prediction		c	p
be	→	*o*	1	1/2	*b*	→	*e*	2	3/4	→	*b*	2	3/26
	→	*esc*	1	1/2		→	*esc*	1	1/4	→	*e*	2	3/26
eo	→	*r*	1	1/2	*e*	→	*o*	1	1/2	→	*n*	1	1/26
	→	*esc*	1	1/2		→	*esc*	1	1/2	→	*o*	4	7/26
no	→	*t*	1	1/2	*n*	→	*o*	1	1/2	→	*r*	1	1/26
	→	*esc*	1	1/2		→	*esc*	1	1/2	→	*t*	3	5/26
ob	→	*e*	2	3/4	*o*	→	*b*	2	3/8	→	*esc*	6	3/13
	→	*esc*	1	1/4		→	*r*	1	1/8	**Order -1**			
or	→	*n*	1	1/2		→	*t*	1	1/8	Prediction		c	p
	→	*esc*	1	1/2		→	*esc*	3	3/8	→	*A*	1	$1/\|A\|$
ot	→	*t*	1	1/2	*r*	→	*n*	1	1/2				
	→	*esc*	1	1/2		→	*esc*	1	1/2				
rn	→	*o*	1	1/2	*t*	→	*o*	2	1/2				
	→	*esc*	1	1/2		→	*t*	1	1/6				
to	→	*b*	2	3/4		→	*esc*	2	1/3				
	→	*esc*	1	1/4									
tt	→	*o*	1	1/2									
	→	*esc*	1	1/2									

by Cleary & Teahan in 1997 (Wu, 2007). One of the special features is that PPM* uses unbounded length contexts. The PPM* algorithm does not select substrings in order models from higher to lower to predict, but uses all substrings from the input string to generate the prediction (Wu, 2007). Obviously, PPM* needs more resources including a larger memory and higher execution speed. PPM with Information Inheritance (PPMII) was proposed by Shkarin (2002), who pointed out that the main difficulty in estimating probabilities is that the statistics of the higher order contexts is not sufficient. PPMII uses the concept of parent and child contexts and initialises values for child contexts according to relative information gathered in the parent contexts, PPMII performs well at both compression rate and speed (Wu and Teahan, 2005). PPMT is a combination of multiple PPMD models using a Viterbi inspired algorithm (Teahan and Harper, 2001). However, there are

some restrictions for PPMT in text mining applications, such as that PPMT cannot extract hierarchical structure (Teahan and Harper, 2001). PPMZ uses better blending algorithms and its escape estimation was finely tuned and achieved the best compression as a mix with LZ77 (Teahan and Harper, 2001). In addition, as a more sophisticated model, PPMZ has the potential to perform better and can work in tandem with PPMT (Teahan and Harper, 2001).

Compared to English and other Western languages, Chinese has many specific features, such as 2 bytes or 3 bytes per character for most encodings and it is not naturally segmented. Segmenting Chinese (as well as Japanese and Korean) is an important prerequisite (Teahan et al., 2000). Therefore, another two variants—PPM-ch and PPMO have been proposed by Wu and Teahan (2005, 2008). The coding process of PPMO can be divided into two steps—*orders stream* and *symbols stream* (Wu and Teahan, 2005). PPM-ch was especially made for Chinese language process. Because every Chinese character employs two bytes (or three bytes in UTF-8 encoding), the main feature of PPM-ch is to use 16 bit symbol schemes rather than byte-based or bit-based (Wu and Teahan, 2008). Therefore, PPM-ch should also be competitive with other variants in other large alphabet size language, such as Japanese, Korean, etc. (Wu and Teahan, 2005). Experience shows that PPM-ch is competitive with byte-based approaches (i.e. PPMD etc.) and PPM-ch still has potential to be improved (Wu and Teahan, 2005). Another reason that PPM-ch achieves excellent compression results at Chinese text processing is that it imports frequency sorting techniques (Wu and Teahan, 2008). The proposal of PPM-ch opened a door for creating new PPM variants for different languages. In further research, there should be further PPM variants for processing other languages such as Japanese (PPM-jp), Korean

(PPM-kr), etc.

In summary, PPM variants have different and significant features, advantages and disadvantages in various environments. Therefore, it is difficult to say which is better than others in all areas and the best way to compare them is to simply perform compression experiments. Standard PPM works almost as well compared to the improved other PPM variants. Therefore, a decision was made to simply use standard PPM for this dissertation as these other variants require more resources.

2.7 Text Encodings

Text encoding is a certain encoding system that can present a repertoire of characters in a language (Unicode Staff CORPORATE, 1991). In order to complete this literature review, we also need to consider the encoding schemes used to encode each language. There are a number of encodings have been designed and used for different languages. Table 2.5 shows some common encodings for different languages.

Table 2.5 listed some different encodings for same languages, such as that there are two encodings—Big5 and HKSCS for Traditional Chinese. Big5 is widely used in Taiwan whereas HKSCS is the main encoding for Hong Kong. There are three common encodings for Japanese in Table 2.5, which are Shift JIS, EUC-JP and ISO-2022-JP where Shift JIS and EUC-JP are respectively used for Windows and Unix systems. The middle column of Table 2.5 indicates how many bytes each character occupies. Characters for most western language encodings are encoded using one byte, whereas oriental languages like Japanese and Chinese encode characters using at least two bytes due to larger alphabet sizes. Particularly, English and Chinese

Table 2.5: Common character encodings for different languages (Benoit, 2013).

Character Encoding	**Byte(s) per Character**	**Language(s)**
ASCII	1	English
ISO-8859-1	1	Western Europe
ISO-8859-2	1	Western and Central Europe
ISO-8859-5	1	Cyrillic alphabet
ISO-8859-6	1	Arabic
ISO-8859-7	1	Greek
ISO-8859-8	1	Hebrew
ISO-8859-11	1	Thai
ISO-8859-13	1	Baltic languages plus Polish
Shift JIS	2	Japanese
EUC-JP	up to 3	Japanese
ISO-2022-JP	2	Japanese
GB2312	2	Simplified Chinese
Big5	2	Traditional Chinese
HKSCS	2	Traditional Chinese
GBK	2	Simplified and Traditional Chinese
GB18030	up to 4	Simplified and Traditional Chinese
UTF-8	up to 6	Multilingual

encodings are introduced in the following sections.

2.7.1 English Encodings

The earliest text encoding of English is ASCII, which is the first encoding designed for information interchange and includes all English letters and basic symbols (Benoit, 2013). ASCII originally was encoded using 128 characters including English letters, common symbol and punctuations and 32 non-printing characters. Table 2.6 shows the main part of basic ASCII character encoding, where "Hex" means the hexadecimal value of each character.

As one byte has eight bits, which means each byte has 256 characters, the extended ASCII encoding uses another 128 spaces to encode more characters which are widely used in Europe countries and is called ISO-8859-1. Table 2.5

Table 2.6: The basic ASCII Table without 32 non-printing characters (Coded Character Set, 1986).

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
20	Space	34	4	48	H	5C	\	70	p
21	!	35	5	49	I	5D	]	71	q
22	”	36	6	4A	J	5E	^	72	r
23	#	37	7	4B	K	5F	_	73	s
24	$	38	8	4C	L	60	‘	74	t
25	%	39	9	4D	M	61	a	75	u
26	&	3A	:	4E	N	62	b	76	v
27	’	3B	;	4F	O	63	c	77	w
28	(	3C	¡	50	P	64	d	78	x
29	)	3D	=	51	Q	65	e	79	y
2A	*	3E	¿	52	R	66	f	7A	z
2B	+	3F	?	53	S	67	g	7B	{
2C	,	40	@	54	T	68	h	7C	—
2D	-	41	A	55	U	69	i	7D	}
2E	.	42	B	56	V	6A	j	7E	~
2F	/	43	C	57	W	6B	k	7F	DEL
30	0	44	D	58	X	6C	l		
31	1	45	E	59	Y	6D	m		
32	2	46	F	5A	Z	6E	n		
33	3	47	G	5B	[	6F	o		

shows more encodings based one ISO-8859 for different languages. The main difference among them is that the extension parts encode different language characters. English as a universal alphabet is included in most character encodings, which encode ASCII as Table 2.6 showed as a part of the whole encodings, such as ISO-8859-1, ISO-8859-2, ISO-8859-5, etc. shown from Table 2.5 (Benoit, 2013).

2.7.2 Chinese Encodings

Since Chinese government issued simplified Chinese characters in 1964, Chinese has been divided into Simplified Chinese and Traditional Chinese, which are currently based in different areas. Simplified Chinese is used in Chinese mainland and some Southeast Asian countries (e.g. Malaysia, Singapore),

whilst Traditional Chinese is the official language in Taiwan, Hong Kong and Macau. There are also many overseas Chinese people still using Traditional Chinese.

GB2312 is the first Simplified Chinese character encoding that was issued in 1980. GB2312 includes 6,763 Chinese characters, which covers 99.75% of Chinese character input and compatible with ASCII. GB2312 encoding occupies 2 bytes for each Chinese character, where the high byte is ranged from 0xB0 to 0xF7 (72 values) in hexadecimal and the low byte is from 0xA1 to 0xFE (94 values). Therefore, there are 72×94=6768 places for Chinese characters in total, where where are 5 empty spaces. In Chinese mainland, GB2312 is a widely used encoding, which supports Simplified Chinese but does not include any Traditional Chinese character.

Big5 was issued in 1983 and is one of the most common encodings for Traditional Chinese. Big5 includes 13,060 Traditional Chinese characters and is also using double-byte encoding. The high byte is ranged from 0x81 to 0xFE and the low byte is using both 0x40–0x7E and 0xA1–0xFE.

HKSCS is another Traditional Chinese character encoding, which is based on Big 5 encoding and extended 4,500 more Chinese characters that Big5 does not include but Hong Kong is using. HKSCS encoding was especially designed for Hong Kong and Macau and also occupies two bytes per character.

However, as Traditional Chinese characters cannot be ignored although Simplified Chinese is the main language in Chinese mainland, GBK is an extension of GB2312 and includes both Simplified Chinese and Traditional Chinese. GBK encoding is the first extension of GB2312 and also uses double-byte encoding. The high byte is from 0x81 to 0xFE and the low byte is between 0x40 and 0xFE.

GB18030 is the latest version of Chinese government standard encoding and encoded by 1 byte, 2 bytes and 4 bytes. The single-byte part matches ASCII and the double-byte part includes all Simplified and Traditional Chinese characters and symbols. The qual-byte part in an extension of CJK Unified Ideographs including Japanese characters, Korean characters, Vietnam characters, symbols, etc. Table 2.7 shows the description of different ranges. GB18030 encoding has been diffusely used for Chinese text resources.

Table 2.7: Table and description of GB18030 encoding (Lunde, 2009).

Type	**Range**	**Total**	**Occupied**	**Details**
Single-byte	0x00–0x7F	256	128	ASCII
Double-byte	High 0xB0–0xF7 Low 0xA1–0xFE	6768	6763	Chinese chars
	High 0x81–0xA0 Low 0x40–0xFE	6768	6080	Chinese chars
	High 0xAA–0xFE Low 0x40–0xA0	8160	8160	Chinese chars
Quad-byte	High 0x81–0x82 Low 0x30–0x39 High 0x81–0xFE Low 0x30–0x39	25200	6530	CJK
	High 0x95–0x98 Low 0x30–0x39 High 0x81–0xFE Low 0x30–0x39	8160	8160	Chinese chars

2.7.3 UTF-8 Encoding

UTF-8 encoding combines all natural languages is capable of all possible characters (Benoit, 2013). UTF-8 is an implementation of Unicode, so that ASCII is completely encoded in UTF-8 and each English character occupies one byte. However, although a Chinese character uses two bytes in Unicode, UTF-8 encodes both Simplified and Traditional Chinese into three or

four bytes per character. Common Chinese characters in UTF-8 encoding occupies three bytes, which approximately enclosed complete GBK encoding (over 21,000 characters). Four bytes encoding includes the rarely used Chinese characters and other languages' characters. Table 2.8 shows Chinese range in UTF-8 encoding. The "x" in Table 2.8 is used to be replaced by Unicode. One of the advantages of UTF-8 encoding is that it is possible to save texts in two or more languages to one file (e.g. parallel corpora). Therefore, UTF-8 is a popular encoding for corpus linguistics.

Table 2.8: English and Chinese in UTF-8 encoding ranges (Unicode Staff CORPORATE, 1991).

Unicode	**UTF-8**	**Byte(s)**	**Details**
0000—007F	0xxx xxxx	1	ASCII
0080—07FF	110x xxxx 10xx xxxx	2	Non-Chinese
0800—FFFF	1110 xxxx 10xx xxxx 10xx xxxx	3	Common 91,000 Chinese characters
0001 0000—001F FFFF	1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx	4	Rest rarely used Chinese characters
0020 0000—03FF FFFF	1111 10xx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx	5	Not available
0400 0000—7FFF FFFF	1111 110x 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx	6	

2.8 Summary & Conclusion

This chapter has firstly reviewed SMT literature including the history and important models of SMT. Then we have reviewed the basic process of SMT, how it works and what it needs, as well as the difficulties of SMT and research directions.

Corpus-based Linguistics is the basic approach of this study which has been discussed in this chapter. We have reviewed a number of important corpora that were created in recent decades as well as the types of corpora. Some useful parallel corpus evaluation methods have also been subsequently discussed, which are essential for evaluating the quality of translations and parallel corpora. Parallel corpus alignment is an important approach that this research uses. At different levels, alignment brings about various potential problems. Parallel corpora with sentences accurately aligned are important for statistical machine translation. We have reviewed the literature about parallel corpus alignment and analysed the methods especially for aligning sentences for parallel corpora. This chapter has also reviewed various PPM models, compared major PPM variants (PPMA, PPMB, PPMC and PPMD) and discussed later variants of PPM. Particularly for Chinese language, PPMO and PPM-ch have been introduced, which are both based on 16 bits symbol scheme. Experience showed that PPMO and PPM-ch are competitive with other variants of PPM especially for Chinese text. However, byte-based PPM (standard PPM) is also a competitive method for compressing natural language text and achieves excellent compression rates (Wu, 2007). A method for sentence alignment using PPM codelengths has also been discussed. Finally, this chapter introduced text encodings which are designed for different natural languages and especially for English and Chinese encodings. UTF-8 is an universal encoding and supports all natural

languages, which is a popular choice for corpus creation.

In the next chapter, this thesis will start to adopt PPM-based methods for compression and propose some PPM-based methods for sentence alignment for parallel corpora.

Chapter 3

Aligning Chinese-English Parallel Corpora using PPM

3.1 Introduction

The aim of this chapter is to justify whether PPM-based compression method outperforms than other common compression methods (Gzip and Bzip2) and to determine how well PPM-based compression method performs for Chinese-English parallel sentence alignment. This chapter is based on the conference paper that has been presented at SLSP 2014 and published in the LNAI proceedings (see Table 1.1, page 5).

This chapter firstly reports comparison results for aligning Chinese-English parallel corpora using PPM and other methods, then finds out the compression scheme that is the most effective at aligning Chinese-English parallel corpora. The scheme is used for aligning sentence alignment and also for comparing between sentence length and compression code length based metrics.

This chapter is organised as follows. The next few sections motivate the

use of compression-based methods for alignment, and describes four distance metrics for matching sentences, two based on sentence length and two based on calculating the compression code length of the sentences. The sections also describe several compression algorithms used in the experiment—PPM, Gzip and Bzip2, and then describes how the compression code lengths can be calculated using a relative entropy approach and "off-the-shelf" compression software. The alignment algorithm we have used is then described next. Two experiments are then described—the first to find out which compression algorithm works best for the code length ratio metric; and the second to compare which of the four metrics perform best at aligning a corpus which was constructed with ground truth judgments concerning the alignment. Conclusions are provided in the final section.

3.2 Background & Motivation

Accurate alignment of textual elements (e.g. paragraphs, sentences, phrase) in a parallel bilingual corpus is an important part of natural language processing and a crucial step for statistical machine translation. A number of different approaches have been developed over the years for aligning sentences between comparable text in a bilingual parallel corpusâfor example, those based on using: sentence length; word co-occurrence; cognates; dictionaries; and parts of speech.

The assumption behind length-based approaches to sentence alignment is that short sentences in the source language will be translated into short sentences in the target language, and the same for longer sentences, and that there is enough variation in sentence length between adjacent sentences to correct mis-alignments when they occur. Gale and Church (1993) aligned

sentences in English-French and English-German corpora by calculating the character length of all sentences, producing a Cartesian product of all possible alignments, then aligning the most plausible alignments iteratively until all sentences are accounted for. Their overall accuracy rate for both corpora was 96% (97% for English-German and 94% for English- French). The best results were for 1:1 alignments, where one sentence in one language corresponds to one sentence in the other language. For 1:1 alignments, the error rate was only 2%. However, there was a 10% error rate for 2:1 alignments and 33% error rate for 2:2 alignments. In comparison for English-Chinese corpora, Wu (1994) also proposed aligning English-Chinese corpora by determining sentence length (in bytes) and also produced a high accuracy of over 95% (Wu, 1994). Length-based measurement has also had satisfactory results for evaluating the corpus extracted from China National Knowledge Infrastructure (CNKI) (Ding et al., 2011).

In recent years, there have been relatively few new proposals for parallel corpora sentence alignment (Yu et al., 2012). Existing sentence alignment algorithms are not able to link one-to-many or many-to-one mutual translations (Kutuzov, 2013). This chapter focuses on adopting a novel compression-based approach as the distance measure to determine whether two sentences are aligned.

3.3 Methodology

3.3.1 Compression-based Alignment

Our idea of using compression-based measures for alignment hinges on the premise that the compression of co-translated text (i.e. documents, paragraphs, sentences, clauses, phrases) should have similar compression code

lengths (Behr et al., 2003). This is based on the notion that the information contained in the co-translations will be similar. Since compression can be used to measure the information content, we can simply look at the ratio of the compression code lengths of the co-translated text pair to determine whether the text is aligned. That is, if you have a text string (i.e. paragraph, sentence, or phrase) in one language, and its translation in another language, then the ratio of the compression code lengths of the text string pair should be close to 1.0 since the information in the texts should be similar (in comparison, this may not be the case when comparing sentence lengths, especially for non-related languages). Alternatively, we can use a relative entropy related measure, and use an absolute code length difference measure—in this case, a value close to 0 indicates that the text string pair are closely aligned.

Formally, given a text string of length n symbols $S^L = x_1 x_2 \dots x_n$ in language L and a model p_L for that language, then the cross-entropy is calculated as follows (Chang, 2008):

$$H(S^L) = -\frac{1}{n} \log_2 p_L(S^L);$$

i.e. the average number of bits to encode the text string using the model $p_L(S^L)$.

$$p_L(S^L) = \sum_{i=0}^{n} p_L(c_i | c_1, ..., c_{i-1})$$

where c_i means the ith character of S^L.

3.3.2 Distance Measures

Four metrics for matching sentences required for measuring the alignment at the sentence level are compared: the standard sentence length ratio (SLR), and three further metrics, absolute sentence length difference (SLD), com-

pression code length ratio (CR), and absolute compression code length difference (CD):

$$SLR = \max\left\{\frac{L(S^E)}{L(S^C)}, \frac{L(S^C)}{L(S^E)}\right\} \tag{3.1}$$

$$SLD = \left|L(S^E) - L(S^C)\right| \tag{3.2}$$

$$CR = \max\left\{\frac{H(S^E)}{H(S^C)}, \frac{H(S^C)}{H(S^E)}\right\} \tag{3.3}$$

$$CD = \left|H(S^E) - H(S^C)\right| \tag{3.4}$$

where L represents sentence length and H means code length. S^E and S^C denote English and Chinese sentences.

SLR has been more widely used in recent years than others. Mújdricza-Maydt et al. (2013) have been using SLR and achieved good performance. CR has already been defined by Alkahtani et al. (2014) for Arabic-English parallel sentence alignment but has not yet been applied to Chinese-English sentence alignment. SLD and CD are new metrics that were devised during this research project. The remainder of this section will describe compression schemes that we have used in the code length calculations for the experiments described below.

3.4 Experiment 1: Comparing Different Compression Algorithms for Alignment Purposes

The purpose of this first experiment was to determine the best compression algorithm for calculating the code length based metrics defined in the

previous section in order to perform the sentence alignment. For the experiment, a test corpus was needed to provide the ground truth data in order to investigate the effectiveness of the different compression algorithms. For the training corpus, we manually selected 1,000 Chinese-English parallel sentences. For the test corpus, we chose 1,000 matching Chinese-English parallel sentences from the DC parallel corpus (Chang, 2008) at random. Table 3.1 shows sample calculations for the first three sentence pairs in the corpus and Figure 3.1 graphs the CR values for all the sentences, for the three compression schemes PPMD, Gzip and Bzip2. In order to compute the code length values (as shown in bytes in Table 3.1), the concatenation of all of the sentences in the corpus was used as the training text to prime the compression models, and the values were calculated by the formula for h_t (see Section 3.4.3). These values were then inserted into the formula (with $H(S) = h_t$) listed in Section 3.3.2 to calculate CR.

There are a large number of compression methods that have been released for compressing data, documents and texts. For this study, we chose two common compression methods, which are Gzip and Bzip2; along with the PPM method to evaluate that how well each performs.

3.4.1 Gzip & Bzip2

Gzip (also called GNU zip) was created by the GNU project and written by Jean-Loup Gailly and Mark Adler (Gzip, 2012). It uses a dictionary-based Lempel-Ziv based method as opposed to the statistical context-based approach of PPM. Gzip is now a popular lossless compression utility on the Internet and Unix operating system.

Bzip2 is another lossless compression algorithm that was developed by Julian Seward (Bzip2, 2012). It uses a block sorting compression algorithm

that makes use of the Burrows Wheeler method to transform the text. Bzip2 performs better than Gzip but the speed is slower.

The reason for choosing PPM, Gzip and Bzip2 in the experiments reported below is that the three schemes represent very different compression methods—statistical (context) based, dictionary and block sorting. A primary motivation for this experiment was to determine which scheme was most effective when applied to the problem of sentence alignment for parallel corpora.

3.4.2 PPMD

The PPM compression scheme and its variants have been introduced in Section 2.6. PPMD usually produces the best compression, therefore this experiment employs PPMD as the chosen variant of PPM.

As we have already discussed the optimal maximum orders of PPM for Chinese and English in section 2.6, this chapter consequently uses order 5 for English and order 6 for Chinese. This is due to the Chinese text using UTF-8 encoding which occupies 3 bytes for each Chinese character and as a result the optimal maximum order for Chinese is two Chinese characters.

The LCMC and Brown corpora are selected as the training data for Chinese and English compression respectively because they are both well-complied, large enough and are high quality.

The PPMD compression program has been coded in the C language in the Text Mining Toolkit (Teahan et al., 2006) and invoked from a Python function via Unix Terminal.

3.4.3 Calculating Code Lengths of Gzip, Bzip2 and PPMD

We will use a relative entropy method to calculate the compression code lengths for PPM, Gzip and Bzip2. This allows us to use "off-the-shelf" software without having to re-implement the compression schemes. Since the size of the text being compressed in each sentence is relatively small, these compression schemes will not have had sufficient data to compress the text effectively since their models are uninitialised and therefore not well tuned for the languages being compressed (English and Chinese). To overcome this problem, a simple expedient is to prime the models using a large representative training sample for each language. The relative entropy technique allows us to do this in order to calculate the code length using the formula $h_t = h_{T+t} - h_T$ where h is the size of a file after it has been compressed, T represents the large training text and t is the testing text (i.e. the sentence being compressed) for which the compression code length calculation is being computed. The method simply calculates the difference in size between the compressed training text with testing text added and the compressed training text by itself.

3.4.4 Comparison among PPMD, Gzip and Bzip2

The compression code length values (in bytes) and the corresponding code length ratios are shown in Table 3.1 for three compression algorithms—Gzip, Bzip2 and PPMD. The codelength ratio is calculated using equation 3.3 that we have discussed in Section 3.3.2.

Note that for the twelfth sentence pair (Id 0012), the h_t value for Bzip2 was 0 (the compression size in bytes of the training and testing text was

Table 3.1: Compressing code length ratios and code length values (in bytes) using Gzip, Bzip2 and PPMD for the first ten sentences in the test corpus.

Sent. ID	**Language**	**Gzip (bytes)**	**Gzip CR**	**Bzip2 (bytes)**	**Bzip2 CR**	**PPMD (bytes)**	**PPMD CR**
0001	Chinese	71	1.614	49	2.020	43	1.265
	English	44		99		34	
0002	Chinese	40	3.077	26	2.385	23	1.438
	English	13		62		16	
0003	Chinese	66	1.886	57	2.140	39	1.444
	English	35		122		27	
0004	Chinese	22	1.692	11	7.091	14	1.273
	English	13		78		11	
0005	Chinese	30	1.579	20	3.900	20	1.333
	English	19		78		15	
0006	Chinese	39	1.444	20	3.350	21	1.000
	English	27		67		21	
0007	Chinese	25	1.667	13	4.846	16	1.333
	English	15		63		12	
0008	Chinese	44	1.630	36	1.565	20	1.250
	English	27		23		16	
0009	Chinese	18	1.200	3	23.333	15	1.071
	English	15		70		14	
0010	Chinese	35	1.167	36	2.250	21	1.048
	English	30		81		22	
0011	Chinese	27	1.929	3	19.667	19	1.357
	English	14		59		14	
0012	Chinese	70	2.059	42	42.000	44	1.760
	English	34		0		25	
0013	Chinese	85	1.349	60	1.767	47	1.043
	English	63		106		49	
0014	Chinese	38	3.167	20	2.600	23	1.643
	English	12		52		14	
0015	Chinese	41	2.278	40	1.650	22	1.294
	English	18		66		17	
0016	Chinese	63	2.250	46	2.130	26	1.368
	English	28		98		19	
...	...	...	...	...	...	...	...

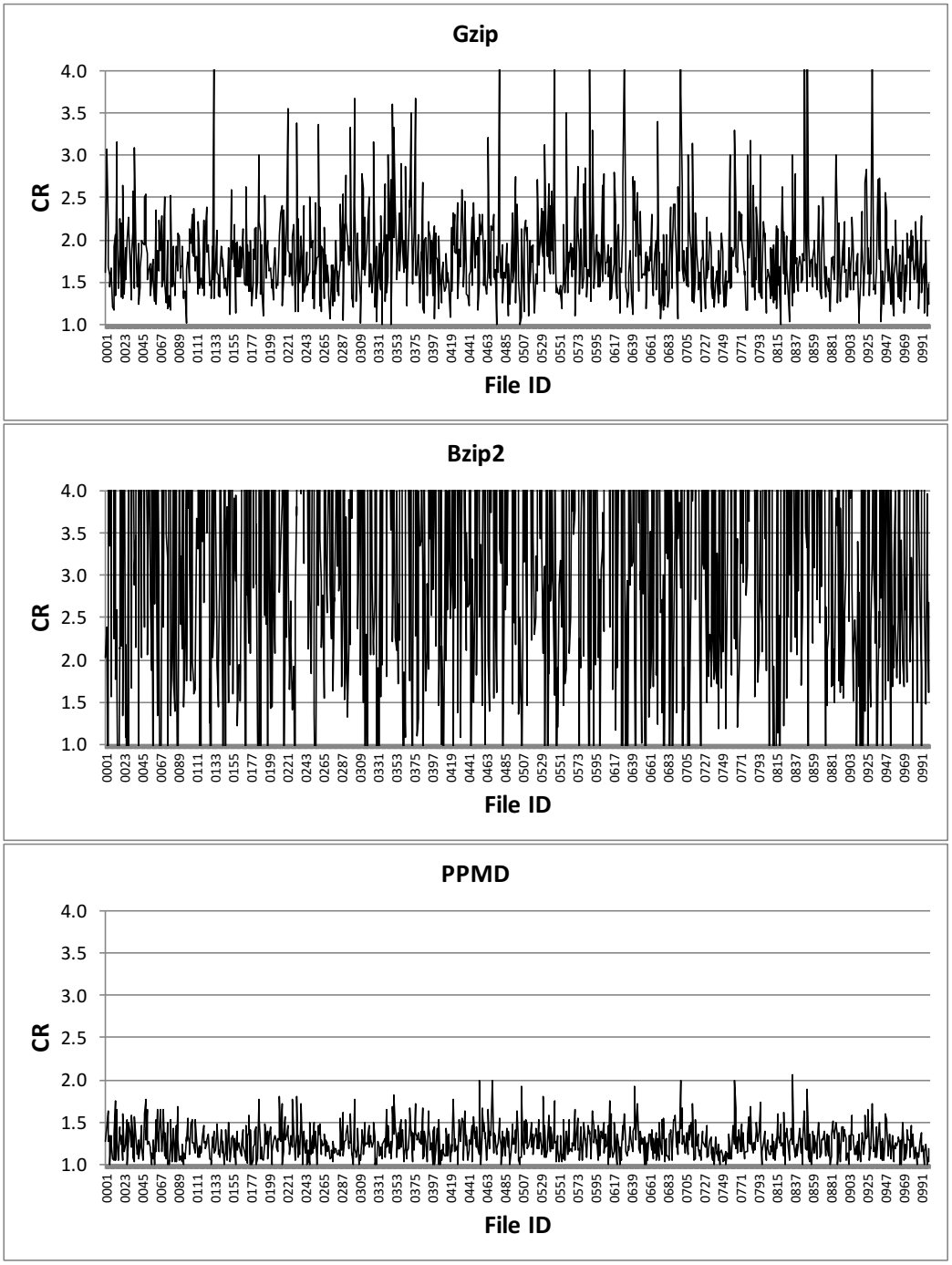


Figure 3.1: Adjusted codelength ratios of the 1000 training models for the three compression algorithms.

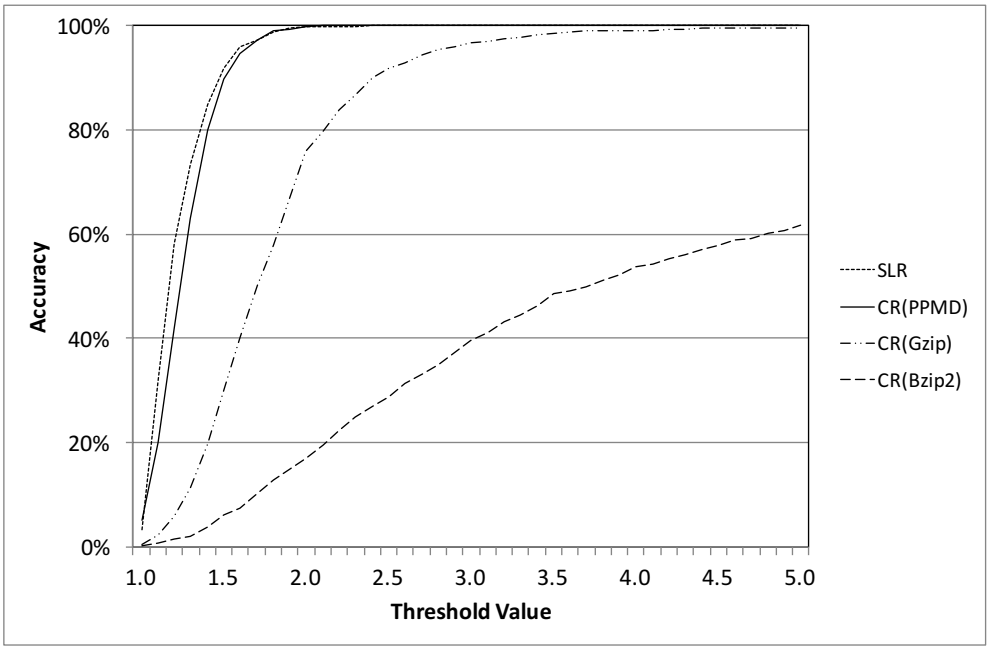


Figure 3.2: Percentage of sentence pairs in text corpus below different SLR and CR values.

exactly the same as the training text by itself). To avoid cases when h_t was 0, a value of 1 was added to h_t in order to avoid infinite values resulting for the CR ratio calculation. There were also 70 negative code length values (7%), which was probably because of the block sorting algorithm—a larger block may cause a more favourable compression with more text and this only happened for Bzip2 compression scheme. These values were set to 1 for the experiment. We can see from the graphs that the code length ratios of PPMD are the most stable with only a few values above 2.0. In comparison, Gzip has greater variation, with many instances when the CR value exceeds 4.0 despite the sentence pairs chosen for the corpus being accurate translations of each other. (The graphs were truncated to a maximum CR value of 4.0 in order that the three graphs could be directly compared). The widest variation clearly belongs to Bzip2, where most values are higher than 10.

Figure 3.2 graphs the percentages of how many sentence pairs are below a certain SLR or CR value (for PPMD, Gzip and Bzip2). From the figure, we can see that the CR values for PPMD performs better than Gzip and Bzip2 at identifying matching sentences for the lower threshold values with similar values to the SLR metric. However, the behaviour for CR values calculated using Gzip and Bzip2 are noticeably different. For example, if we focus on the range between 1.0 and 1.5, there are 930 sentences out of 1,000 in this range for PPMD, but for Gzip and Bzip2, the amounts are much lower (633 and 129). Due to the testing corpus being manually selected from the DC corpus, which is very well aligned at sentence level and the quality of translations are very high, the SLR metric in this case showed a better performance than CR. However, in the next section, we will compare how the sentence length- and compression code length-based metrics perform in detail.

It is not clear why PPM performs significantly better at alignment than the other two compression schemes, since Gzip and Bzip2 are known to also provide good estimates of the cross-entropy, although Gzip frequently flushes its dictionary, whereas Bzip2 uses a non-streaming approach unlike the other two algorithms and this may affect the relative entropy calculations. Further investigation is required to determine the reasons for the difference and also to check whether this result occurs for all language pairs and for other alignment tasks. In fact, preliminary experiments with other languages such as Arabic and Welsh indicate that PPM performs better as well (Alkahtani, 2015; Humphreys, 2008).

3.5 Experiment 2: Chinese-English Parallel Sentence Alignment

The purpose of the second experiment was to determine how well the four distance measures (SLR, SLD, CR and CD) perform for Chinese-English parallel corpus sentence alignment. The calculations of the four distance measures have been introduced in Section 3.3.2.

3.5.1 Preparation of Corpus for Experiment

A corpus was prepared for this experiment in order to test sentence alignment algorithms. The test corpus should be well-compiled and correctly aligned at the sentence level.

We manually selected satisfactory Chinese-English translations from public bilingual text from the Internet as the test corpus, which includes 1,000 1:1 parallel sentences, fifty 1:2 and 1:3 sentences and fifty 2:1 and 3:1 sentences placed throughout the corpus in an ad hoc manner. All the sentences were bilingual news or parallel articles downloaded from the Internet on various topics. The English part of the corpus includes 15932 words and 92508 characters, and the Chinese part has 29046 Chinese characters. The 1,100 sentence pairs were then randomly shuffled to ensure variation in the experiment.

A sample of the corpus is shown below. The corpus file uses XML markup conventions with tags <Ex> identifying the English sentences and tags <Cx> identifying the Chinese sentences, where "x" indicating the index number of the sentence which is from 1 to 1,100. The following sample shows the first three sentences of the test corpus.

English text:

<E1>The rear cover of the iPhone tells you it is designed in California and assembled in China.</E1>
<E2>The phone sells, in the absence of carrier subsidy, for about $700.</E2>
<E3>It matters little where we pass the remnant of our days.</E3>
...

Chinese text:

<C1>iPhone后盖上写道，它在加州设计，在中国组装，在没有运营商补贴的情况下，这部手机的售价为700美元左右。</C1>
<C2>我们在什么地方度过我们的余年已经无关紧要。</C2>
<C3>卵和胚发育能力的不同是否被代谢分布所平衡？</C3>
...

Theoretically, when we examine these sentences for alignment purposes, the following are five possible solutions:

1. *E1 matches C1* *(Wrong)*
2. *E1 matches C1 and C2* *(Wrong)*
3. *E1 matches C1, C2 and C3* *(Wrong)*
4. *E1 and E2 match C1* *(Correct)*
5. *E1, E2 and E3 match C1* *(Wrong)*

The correct answer is the fourth one—*E1* and *E2* match *C1*, and then *E3* matches *C2*. The purpose of the alignment algorithm described in the next section is to automatically determine this correct alignment.

3.5.2 Alignment Algorithm

This section describes the algorithm that was used to align sentences. Although optimal alignment can be computed in polynomial time using Dynamic Programming techniques (Bertsekas, 2011), for our experiment, we have chosen to investigate an alternative complete depth-limited search based method described as follows.

Alignment of sentences may be one to one (1:1), one to many (e.g. 1:2, 1:3), many to one (e.g. 2:1 and 3:1) and many to many (e.g. 2:2). For the work described here, one to zero (1:0) and zero to one (0:1) were not considered. For efficiency reasons for our alignment algorithm, we do not consider the many to many case or the 1:n and n:1 cases where n>3. In contrast, Moore (2002) also proposed $n\leq 2$ because the situation of n>3 is extremely rare. However, we have found that n=3 did happen in Chinese-English parallel corpora, so therefore for our experiments, we use $n\in[2, 3]$. Therefore, for our setting, the search for the best alignment can be considered to be a 5-tree with five branches at each node as shown in Figure 3.3. If the 1:1 and 0:1 cases were to be included, this would require a 7-tree to be used, but as stated, this has not been investigated and is left for future work.

The search begins at the node labelled "Start" at depth d=0 in the tree where the algorithm is positioned at the beginning of each of the two list of sentences being aligned. In this example, the lists of sentences have been denoted as [*ABCDEF...*] and [*abcdef...*]. From the Start node there are five possible alignments to examine at depth d=1—a 1:1 mapping where sentence A is aligned with sentence a, a 1:2 mapping, where sentence A is instead aligned with the first two sentences in the second list, denoted by ab, a 1:3 mapping for sentence pairs A and abc, the 2:1 mapping for the pair AB with a and the 3:1 mapping for the pair ABC and a.

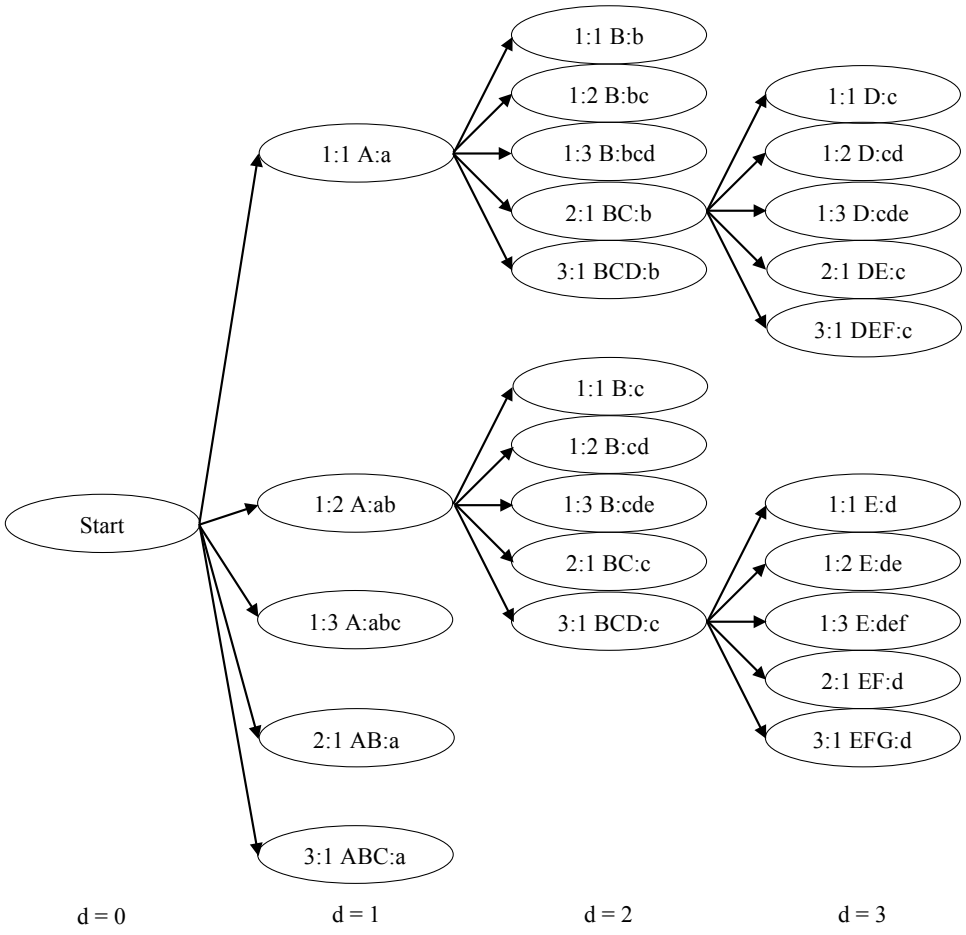


Figure 3.3: 5-tree for aligning sentences.

For each node at depth d=1, there are five child nodes at d=2 that then have to be searched in turn. Note that only a subset of the set of nodes in the 5-tree are shown in Figure 3.3 as it is not possible to display the full 5-tree in the diagram within the space available. The figure shows the expansion of the first two nodes at depth d=1, and two selected nodes at depth d=2 for illustration purposes. For example, the top node at depth d=3 represents the alignment where sentence A has been aligned with sentence a, then sentences BC have been aligned with b, then sentence D has been aligned with c.

The path cost from a node to one of its child nodes is defined as a calculation result by a given distance metric that measures the quality of the specific alignment, such as ones based on sentence length (SLR and SLD) and ones based on code length (CR and CD). The aim of the search is to find a path with the minimum sum of path costs through the tree to a leaf node (which is determined by the maximum depth of the tree).

The complexity of the search for the 5-tree is 5^d. Therefore, when d=9, searching the best path in the 5-tree with the minimum path cost will need to compare 5^9=1,953,125 paths i.e. find the minimum sum of cost paths from nearly two million numbers. In our experiments described below, we have explored the case when $d \in [1, 9]$. Experiments with the four distance metrics show that in most cases, the deeper the search, the better the overall alignment quality, but this is at the cost of significantly longer time spent on the search.

In order to align the full list of sentences, a sliding window method was adopted. An alignment at a particular position is chosen using the 5-tree search which then determines the width of the window according to the alignment. The algorithm then advances to the next position after the window and so on until the entire text has been aligned.

3.5.3 Experimental Results

The purpose of this experiment was to compare the four different metrics defined in Section 3.3.2. We used the test corpus described in Section 3.5.1 in order to evaluate the effectiveness of the different metrics.

As previously stated, there are five possible options—"1:1", "1:2", "1:3", "2:1" and "3:1", and each alignment requires a minimum number of sentences for each language and the number depends on the depth value d. The

minimum required amount of sentences can be calculated as $3 \times d$. The maximum d that this experiment uses is 9, therefore, a minimum number of 27 sentences are provided to guarantee that there are enough nodes (sentence combinations) to meet all possible solution requirements. The preliminary results after running programs for the four different path cost calculation methods are shown as Table 3.2.

The accuracies in the last rows of the four tables were calculated as follows:

$$Accuracy = \frac{Amount_{correct}}{Amount_{total}} \times 100\%$$

where $Amount_{total}$ was always 1,100 in this evaluation and $Amount_{correct}$ was the number of sentence pairs that were correctly aligned out of the total number of sentences. The alignment algorithm described in Section 3.5.2 was applied to the problem of aligning the test corpus. Table 3.3 compares at various search tree depths the sentence alignment accuracies that resulted using the four different metrics. From the table, we can see that difference based metrics (SLD and CD) always performed better than their corresponding ratio based metrics (SLR and CR) and that code length metrics (CR and CD) performed better than sentence length metrics (SLR and SLD). Overall, the code length difference metric (CD) is the best performed metric in this comparison.

Figure 3.4 shows the performance tendencies of the four metrics where we can see significant improvements with growing depths for SLD, CR and CD. These improvements may be because Chinese and English are unrelated languages and therefore the SLR and SLD metrics are not as effective due to differences between sentence lengths. However, SLR did not show a growth trend. It is reasonable to believe that there will be more competitive results if the depth of the 5-tree is greater than 9 although this would be at a

Table 3.2: Alignment accuracy produced on the test corpus using the alignment algorithm and the sentence length ratio metric.

Sentence Length Ratio Metric									
Depth=	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total	1100	1100	1100	1100	1100	1100	1100	1100	1100
Wrong	137	128	129	138	146	136	125	129	127
Correct	963	972	971	962	954	964	975	971	973
Accuracy	87.5%	88.4%	88.3%	87.5%	86.7%	87.6%	88.6%	88.3%	88.5%

Sentence Length Difference Metric									
Depth=	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total	1100	1100	1100	1100	1100	1100	1100	1100	1100
Wrong	107	70	75	63	68	65	62	58	53
Correct	993	1030	1025	1037	1032	1035	1038	1042	1047
Accuracy	90.3%	93.6%	93.2%	94.3%	93.8%	94.1%	94.4%	94.7%	95.2%

Code Length Ratio Metric									
Depth=	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total	1100	1100	1100	1100	1100	1100	1100	1100	1100
Wrong	128	109	94	85	79	72	76	74	71
Correct	972	991	1006	1015	1021	1028	1024	1026	1029
Accuracy	88.4%	90.1%	91.5%	92.3%	92.8%	93.5%	93.1%	93.3%	93.5%

Code Length Difference Metric									
Depth=	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Total	1100	1100	1100	1100	1100	1100	1100	1100	1100
Wrong	92	70	64	61	53	49	45	49	43
Correct	1008	1030	1036	1039	1047	1051	1055	1051	1057
Accuracy	91.6%	93.6%	94.2%	94.5%	95.2%	95.5%	95.9%	95.5%	96.1%

Table 3.3: Comparison at various search tree depths of sentence length alignment accuracies for different metrics: SLR, SLD, CR and CD.

Depth	SLR	SLD	CR	CD
1	87.5%	90.3%	88.4%	**91.6%**
2	88.4%	**93.6%**	90.1%	**93.6%**
3	88.3%	93.2%	91.5%	**94.2%**
4	87.5%	94.3%	92.3%	**94.5%**
5	86.7%	93.8%	92.8%	**95.2%**
6	87.6%	94.1%	93.5%	**95.5%**
7	88.6%	94.4%	93.1%	**95.9%**
8	88.3%	94.7%	93.3%	**95.5%**
9	88.5%	95.2%	93.5%	**96.1%**

significant cost in search time.

Although not optimised, the speed of code length calculation is slower than sentence length calculation especially when depth $d \geq 6$. It takes about 4.1 seconds on a Macbook Pro laptop (Processor: Intel Core i5 2.4GHz) per 5-tree search at $d=6$ and 66.1 seconds at $d=9$. Note that there are some dips in Figure 3.4, especially for depth $d=5$ for SLR. One of the possible reasons is that sentences of the test corpus were not in a natural sequential order, and therefore the results may be affected by this.

In addition, more recent sentence alignment methods make use of multiple cues. Haruno and Yamazaki (1996) have proposed combining both statistical and dictionary information for aligning Japanese-English bilingual text and precision and recall rates outperformed methods using just statistics and dictionary information alone. Alkahtani et al. (2014) employed a combined PPM compression-based code length ratio and sentence length ratio method for Arabic-English translation quality evaluation to improve the quality. A primary purpose of this chapter has been to establish a baseline performance

for a method based purely on the code length, and investigation of hybrid methods such as that devised by Alkahtani et al. (2014) (i.e. combing SL and CL metrics) has been left as future work.

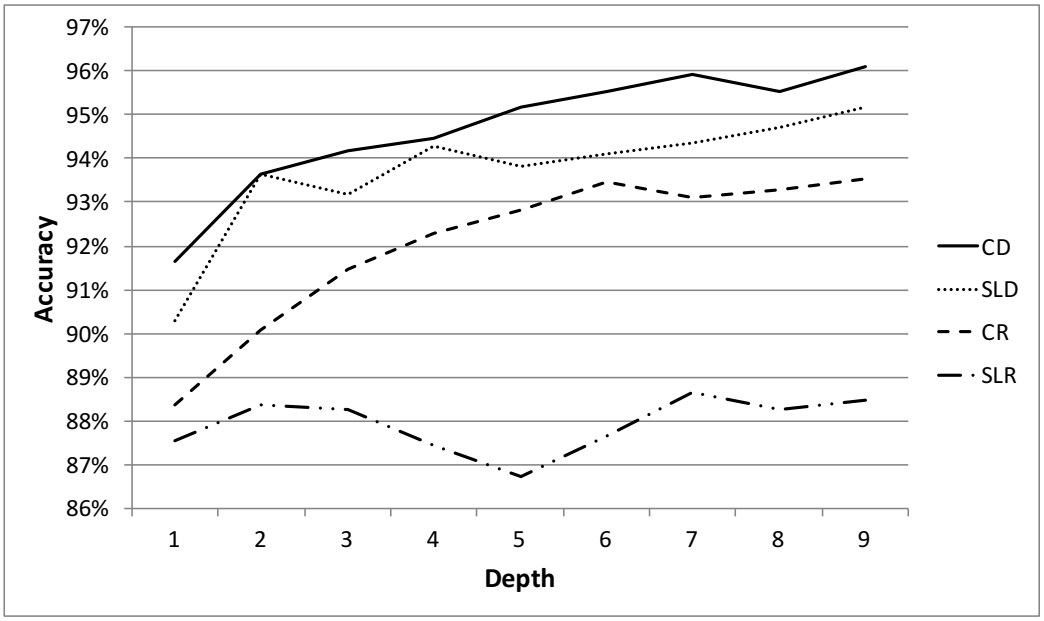


Figure 3.4: Comparison at various search tree depths of sentence length alignment accuracies for the different metrics: SLR, SLD, CR and CD.

Clearly, the calculation of the compression code lengths for the compression based alignment method incurs greater computational overheads than the sentence length based method (although the extra overhead is not as great if the (arithmetic) coding part of the PPM compression calculation is removed and only the modelling part is performed). For example, calculation of compression code lengths minus the arithmetic coding for the entire testing corpus after loading the English and Chinese training models requires on average 0.440 seconds on a MacBook Pro. Length-based alignment techniques can also be used as a quick way to filter the search space (as done by Moore (2002) and Varga et al. (2005)) prior to the applications of more

Table 3.4: PPMD compression speed performance for the testing corpus.

Language	One Sentence		Two Sentences		Three Sentences	
	Files	Average Second(s)	Files	Average Second(s)	Files	Average Second(s)
English	1152	0.418	1151	0.415	1150	0.415
Chinese	1151	1.182	1150	1.181	1149	1.188

complex models including those using compression code length calculations.

Table 3.4 shows the average speed performance for PPMD compression using the same testing corpus for this experiment. We tested the English and Chinese corpora split as a single sentence, then as two sentences and three sentences and calculated the average speed required for the compression codelength calculations separately. Interestingly, we can see that compressing a Chinese sentence takes significantly longer than compressing an English sentence.

Furthermore, we also tested the speed for each sentence alignment of a 5-tree depth-limited search in depth from 1 to 9 as Table 3.5 shows, where we can see that the speed became significantly longer when the depth is greater than 7. Table 3.5 also shows that the elapsed time of the depth-limited search is increasing exponentially.

3.6 Conclusion

Two new distance metrics have been introduced for matching sentences for alignment of parallel corpora. Two of the metrics are based on computing the compression code length of the sentences as this is an accurate measure of the information contained in the text. The idea is that if the sentences are aligned, then the information contained in sentences that are co-translations

Table 3.5: PPMD compression speed performance for the 5-tree depth-limited search at different depths.

Depth	**Average Seconds**
1	3.601
2	3.615
3	3.586
4	3.600
5	3.722
6	4.085
7	6.108
8	16.178
9	66.065

of each other should match. Experimental results show that the compression-based measures will give a more accurate metric well founded in information theory than alternative metrics based on sentence length which are essentially cruder estimates of the information. Overall, the best metric for determining sentence alignment was based on absolute compression code length difference between sentence pairs. Absolute difference based metrics (including when using sentence length) were also more effective than using ratio based metrics.

The experimental results also show that the PPM compression scheme is the most effective for calculating the compression code lengths for alignment purposes compared to Gzip and Bzip2. PPM provides better entropy estimates than Gzip of Bzip2, and this is reflected in the alignment results. In addition, Gzip frequently flushes its model, whereas Bzip2 uses a non-streaming approach, and this may contribute to these algorithms being less effective for alignment purposes. We are confident that the PPM alignment method will also be effective for alignment down to phrase and even word levels. In terms of extendibility, the PPM compression code length metrics

can also be combined with dictionary-based alignments, integrate cognate feature, etc. for further experiments.

Chapter 4

Evaluating the Quality of Chinese-English Parallel Corpora

4.1 Introduction

Normally, parallel corpora are large in size and therefore it is difficult to evaluate their quality. This chapter will focus on the sentence alignment quality of parallel corpus evaluation, which is based on sentence length and PPM-based code length measurements. More experimental results for Chinese-English parallel corpora are presented for the proposed evaluation approach. The new approach for parallel corpus evaluation makes it easier compare different corpora and therefore make a decision more quickly which corpus is more suitable for a specific purpose.

This chapter also employs and compares sentence length metrics and code length for the Chinese-English parallel corpus evaluation and does a further evaluation to compare sentence length and code length metrics on

more Chinese-English parallel corpora. The hypothesis being tested in this chapter is that a code length ratio of a sentence pair from parallel corpora reflects the translation quality better than using the sentence length ratio.

4.2 Selection of Parallel Corpus

This section discusses the corpora used for those experiments. To accomplish the study, three Chinese-English parallel corpora which were collected from totally different resources have been used as testing corpora for the initial experiments—the DC Corpus, the HK Corpus and the UN Corpus. The DC Corpus was built using resources from free online Chinese-English dictionaries of Yahoo, Google, Microsoft, Dict.CN and Baidu (Chang, 2008). Therefore, the DC Corpus has various categories for each sentence and all sentences are not naturally sequenced. The HK Corpus for the first experiment was selected in Simplified Chinese and English and manually built from the Hong Kong Yearbook 2006. The corpus was also imperfectly manually aligned at the sentence level and a number of translations are mistranslated. The domain of the HK Corpus is government reports. The original United Nations (UN) Corpus is a six-language parallel corpus (Arabic, English, French, Simplified Chinese, Spanish and Russian). The corpus that this chapter uses is a selected part of the original UN Corpus which includes general assembly reports of the years between 2001 and 2007 and has been manually sentence-aligned, which includes normal documentation released by the UN in Simplified Chinese and English. All of the three testing corpora are encoded by UTF-8 encoding and sentence-segmented by line breaks.

The training corpora this chapter used are the Brown Corpus (Francis, 1965) for English and the LCMC Corpus (McEnery and Xiao, 2004) for

Table 4.1: Details of DC Corpus, HK Corpus and UN Corpus.

Corpus	Size	English Words	Chinese Characters	English Sentences	Chinese Sentences
DC	7.67MB	725,382	1,286,996	81,455	81,455
HK	2.01MB	172,717	313,798	6,432	6,432
UN	20.44MB	1,795,364	3,160,485	55,748	55,748

Simplified Chinese. The Brown University Standard Corpus of Present-Day American English (Brown Corpus) was built in the 1960s and the second edition was issued in the 1970s. The Brown Corpus includes a wide range of styles and varieties of prose by native speakers of American English. The Lancaster Corpus of Mandarin Chinese (LCMC Corpus) was built using 15 categories including reportage, reviews, essays, prose, fiction, humour, etc. This experiment used the fiction part of LCMC as the training Chinese corpus. Both the Brown and LCMC Corpora are encoded by UTF-8 encoding and sentence-segmented by line breaks.

Table 4.1 shows the details of the three corpora that the first experiment used, whereas Table 4.2 lists the two training corpora in detail. It can clearly be seen that the UN Corpus is much larger than the other two testing corpora. Table 4.1 also shows that the UN Corpus has more English words and Chinese characters than the DC Corpus but has fewer sentences, which means that the average sentence length of UN Corpus is significantly longer. The Brown and LCMC Corpora are not parallel so that "—" in Table 4.2 indicates fields that are not applicable.

Table 4.2: Details of the training corpora.

Corpus	Size	English Words	Chinese Characters	Sentences
Brown	5.97MB	1,024,884	—	51,126
LCMC	4.55MB	—	1,547,546	37,932

4.3 Methodology

The PPM compression-based method for aligning parallel sentences has already been introduced in Section 2.6 (page 33) and Section 3.3.1 (page 51). The method that this chapter uses is mainly for evaluating parallel corpora by compressing the whole corpora and the individual sentences. Sentence length and code length metrics are used to evaluate the alignment quality for the sentences in the three testing corpora.

PPMD performed very well and has achieved satisfactory results for aligning parallel sentences (see Chapter 3). Therefore, we continue using PPMD as the chosen PPM variant for the following experiments. A training corpus is not necessary for PPM to compress large corpora, so in the first step there was no training corpus used for compressing the three whole testing corpora. Chang (2008) has concluded that the best maximum order for compressing English text is 5 and 6 and for Chinese is 6 (2 Chinese characters in UTF-8 encoding due to each Chinese character occupying up to 3 bytes). Therefore, the maximum orders used in this chapter for English and Chinese are respectively 5 and 6. After compression, compressed sizes (code length values), compression speed, sentence length ratios (SLR) and code length ratios (CRs) were obtained, where compression speed is averaged by compressing each corpus ten times. The sentence length value for the whole corpus is actually its file size because the three parallel corpora are saved as raw text.

The experiment compared sentence length ratio (SLR) and code length ratio (CR) for the whole parallel corpora in the first part of experiment 1 in order to determine the quality of corpora.

In the second part of experiment 1, the Brown and LCMC corpora were used as training corpora for PPMD compression method to accomplish compressions for all sentences from the three testing corpora. Then a comparison of how many sentences' sentence length and code length values are greater than their corresponding translations in the other language was recorded. The experiment also determined average sentence length and code length values for all sentences and discussed the overall qualities of the three testing corpora based on the average SLRs and CRs in order to see whether the average values represent well the overall sentence alignment quality of parallel corpora. Scatter plots are a good method to visually compare the distribution of sentence length and code length for each sentence pair. In addition, SLRs and CRs of sentence pairs indicate the corpus quality of specific sentence pairs in detail. A parallel corpus with good quality should have most CR values which are closer to 1.0. However, according to the hypothesis proposed for this chapter, SLR values are not so sensible for evaluating the quality of a parallel corpus as CR values. This will also be verified in the following experiment.

4.4 Experiment 1: DC, HK and UN Corpora Evaluation

After separately compressing English and Chinese text files for the three testing corpora, Table 4.3 shows the results, where the UN Corpus has the best compression ratio—the lowest bpc (bytes per character) value. A lower

Table 4.3: Comparing sentence lengths, code lengths and speed for the DC, HK and UN corpora.

Corpus		DC	HK	UN
English	**SL (bytes)**	4,004,829	1,112,245	11,546,440
	CL (bytes)	992,840	245,514	2,088,429
	bpc	1.983	1.766	1.447
	Speed	7.3s	1.6s	19.7s
Chinese	**SL (bytes)**	3,674,786	898,412	8,893,849
	CL (bytes)	1,049,874	238,121	1,908,172
	bpc	2.286	2.120	1.716
	Speed	8.7s	1.5s	17.8s
SLR		1.090	1.238	1.298
CR		1.057	1.031	1.094

value of bpc indicates a better compression and is a better estimate of the cross-entropy (Teahan, 2010).

Calculating the sentence length for a corpus is easy and fast, whereas calculating the code length is slower due to the process of compressing. Table 4.3 presents English and Chinese text's sentence length, codelength and their ratios. Compared with the row of SLR, the CR values are closer to 1.0, which indicates that English and Chinese text for each corpus carries similar amounts of information. However, SLR does not make much sense for measuring the information carried.

After all the English and Chinese sentences of the three testing corpora were compressed, Table 4.4 describes how many English sentences' sentence length and code length values are greater than, equal to and smaller than Chinese. Clearly, most English sentences are longer than Chinese sentence in the three testing corpora. Especially for the UN Corpus, over 94% of English sentences are longer. However, after using compression, for comparison, the

Table 4.4: Comparison of sentence length and code length greater than each other among DC, HK and UN Corpus.

Corpus	Sentence Length			Code Length		
	English Greater	Equal	Chinese Greater	English Greater	Equal	Chinese Greater
DC	64.25%	4.04%	31.71%	28.97%	13.77%	57.26%
HK	73.07%	0.75%	26.18%	36.82%	2.41%	60.77%
UN	94.85%	0.41%	4.74%	31.46%	6.85%	61.69%

situation is clearly different. More sentence pairs have equal code length values and most Chinese code length values became greater. The reason of this phenomenon is that PPM-based compression for Chinese results in a lower bpc value than for English as shown in Table 4.3.

Figure 4.1a, 4.1c and 4.1e are scatter plots for the three testing corpora. Each point indicates a sentence pair and is located by its English and Chinese sentence length values. The x-axis is for English sentence length values whereas the y-axis is for Chinese sentence lengths. Both of the axes for these plots 400 bytes. In contrast, points in Figure 4.1b, 4.1d and 4.1f are located by their English and Chinese code length values. The x-axis and y-axis respectively are for English and Chinese code length values. Both of these scales are restricted to a maximum of 100 bytes.

The DC Corpus in Figure 4.1a and 4.1b shows fewer noisy points, which reflects that the sentences of the corpus are probably good quality translations reflecting that they were manually collected.

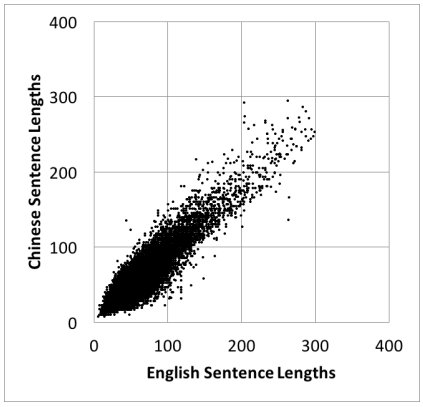
Figure 4.1c and 4.1d plot distributions for the HK Corpus. Figure 4.1c clearly shows that there are many noisy points, which indicates that there are many unsatisfactory translations or mistranslations. Most points in Figure 4.1d for $x \leq 40$ and $y \leq 40$ reflecting that most sentences in the HK

Corpus are short.

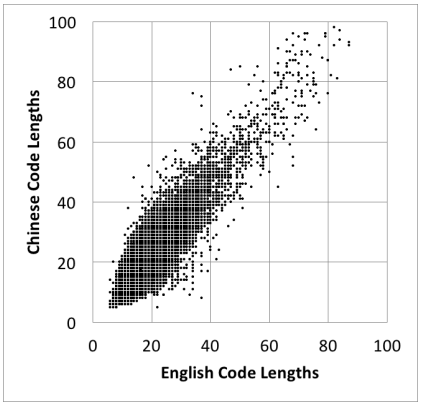
Figure 4.1e and 4.1f show the distribution of the larger UN Corpus. There are fewer noisy points than Figure 4.1c but there are more noisy points than Figure 4.1a reflecting that there are a certain number of unsatisfactory translations and mistranslations but most points are in the normal area. Figure 4.1f has almost diagonal symmetry and indicates that most CR values are closer to 1.0. However, we can see that some of the plots are not centred around the diagonal line $y = x$, which reflects that there are still a number of unsatisfactory translations in the UN corpus.

Table 4.5 presents the average values of sentence length and codelength of English and Chinese and their ratios for the three testing corpora. Higher average sentence length ratios and average sentence code length ratios indicates a higher risk that the sentence pair is an unsatisfactory translation including mistranslation, misalignment of a good translation, etc., whereas lower ratios indicate better quality translations. Sentence length ratios in Table 4.5 show that the SLR of 1.764 for the HK Corpus is likely to have more unsatisfactory translations or mistranslations than the others and the SLR of 1.220 for the DC Corpus is likely to have the fewest unsatisfactory translations or mistranslations. The CR values of the DC and UN corpora are both 1.178, which is a reasonable value for a high quality corpus. However, with the CR of 1.529 and the SLR of 1.764, the HK Corpus can be identified as a lower quality parallel corpus than the others, and it is very likely that it includes many unsatisfactory translations or mistranslations.

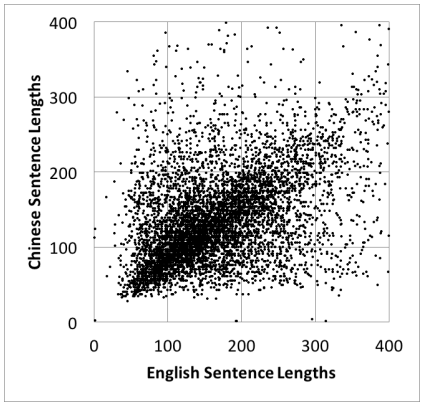
To support the initial summing-up from Table 4.5, Figure 4.2 shows percentages of how many sentence length ratios, code length ratios, sentence length differences and code length differences are greater than the values on the x-axis for the DC, HK and UN Corpora. Theoretically, for a high quality



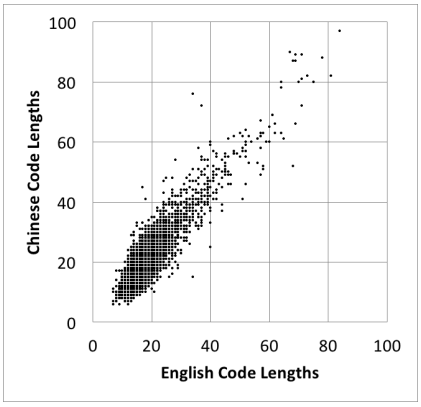
(a) SL values for the DC Corpus.



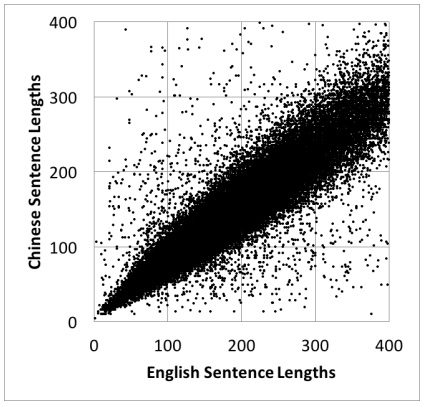
(b) CL values for the DC Corpus.



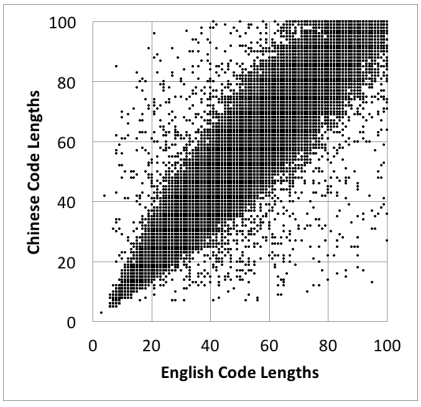
(c) SL values for the HK Corpus.



(d) CL values for the HK Corpus



(e) SL values for the UN Corpus



(f) CL values for the UN Corpus

Figure 4.1: Scatters for sentence lengths and code length of DC Corpus, HK Corpus and UN Corpus.

Table 4.5: Comparing average sentence lengths, code lengths, sentence length ratios and code length ratios for the DC Corpus, the HK Corpus and the UN Corpus.

Corpus	**English**		**Chinese**		**SLR**	**CR**
	SL	**CL**	**SL**	**CL**		
DC	49.166	16.275	45.144	17.664	1.220	1.178
HK	172.924	49.117	139.678	53.411	1.764	1.529
UN	207.118	52.801	159.537	56.939	1.357	1.178

parallel corpus, most CR values should close to 1.0. From the CR curve of Figure 4.2a, less than 15% of CR values are higher than 1.4 and all CR values are less than 1.9. The SLR curve does not show a large difference with CR, and reflects the higher quality of the corpus since almost all sentence pairs of the DC corpus were manually collected. This is also shown in Figure 4.2b, where 100% of CD values and 16.5% of SLD values are below 15.

For Figure 4.2c, the CR curve clearly shows that there are likely to be many unsatisfactory translations or mistranslations in the corpus because there are still over 15% with CR values greater than 2.0. The CR curve should be very sheer in the range between 1.0 and 1.2 for a high quality parallel corpus. However, the CR curve in Figure 4.2c is not sheer and there are over 37% of CR values greater than 1.4, which indicate a lower quality parallel corpus. Compared to CR, the SLR curve in Figure 4.2c is similar in appearance with a narrow gap. In addition, the SLD and CD curves in Figure 4.2d also highlight the lower quality of the HK corpus compared to the DC corpus. There are over 40% of CD values and 80% of SLD values higher than 15.

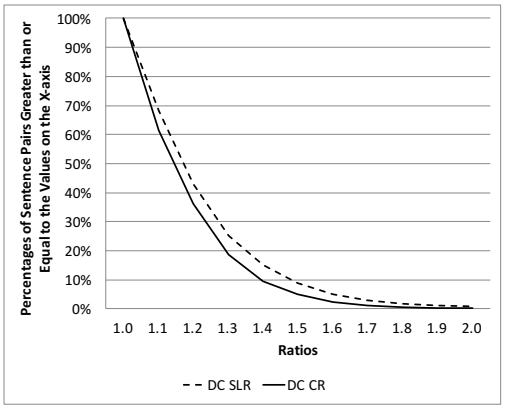
The CR curve in Figure 4.2e shows that the quality of the UN corpus is better than the HK corpus. There are less than 7% of CR values greater

than 1.4 and over 1% are greater than 2.0. A greater difference between the SLR and CR curves would indicate that there are fewer sentences with considerable length differences (which usually indicates a problem in alignment or translation). Compared to the DC and HK corpora, Figure 4.2f shows that there are less than 15% of CD values and 85% of SLD values higher than 15, which indicates a significantly better quality than the HK corpus and closer to the quality of the DC corpus. Additionally, the wider gaps between sentence length and code length curves in Figures 4.2e and 4.2f than Figures 4.2a and 4.2b is another indication that sentences in the UN corpus have less length differences than the DC corpus.

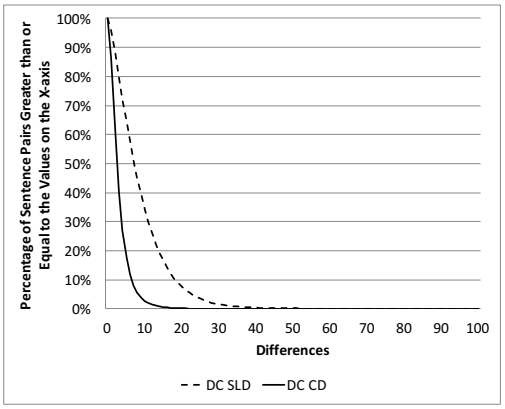
Figures 4.3, 4.4 and 4.5 have shown the values for the DC, HK and UN corpora about the four metrics (SLR, CR, SLD and CD). The x-axis shows the sentence numbers for each sentence. The three figures for the three corpora show the distributions of the four metrics in detail, which is an extension of Figure 4.2 for comparing the three corpora. These figures provide a quick way of visually confirming where possible misalignments have occurred (since these are represented by the frequency of spikes in the graphs and their amplitudes).

4.5 Experiment 2: KDE4 and GNOME Corpora Evaluation

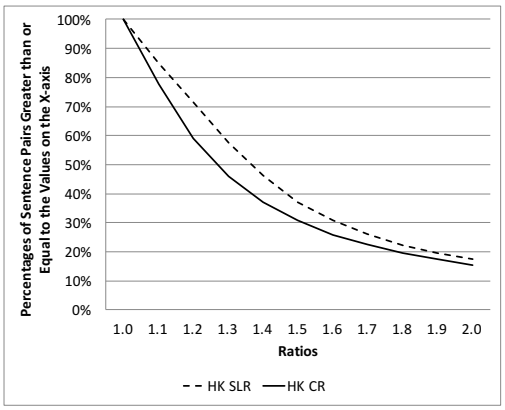
In order to explore further how effective the code length approach is at evaluating the quality of Chinese-English parallel corpora, another experiment was conducted with more corpora—the KDE4 corpus and the GNOME corpus. The two corpora were both automatically generated and downloaded from OPUS (OPUS, 2015a,b) and consequently have lower sentence align-



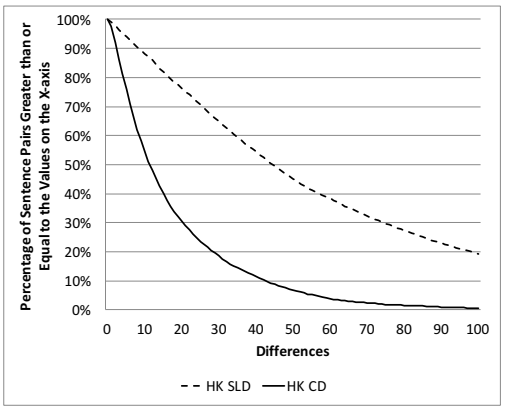
(a) SLRs and CRs of the DC corpus.



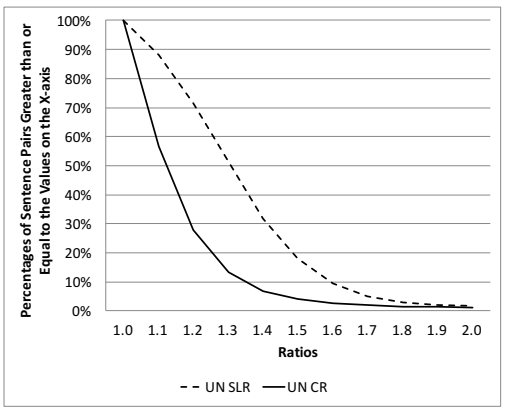
(b) SLDs and CDs of the DC corpus.



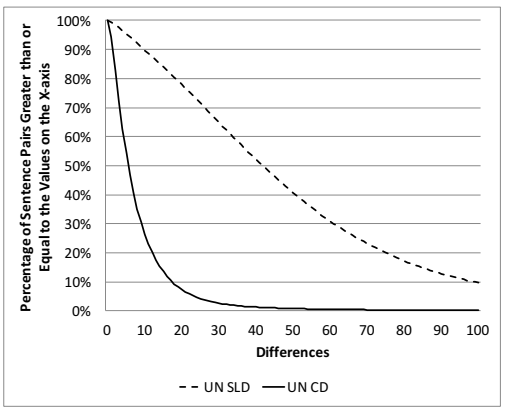
(c) SLRs and CRs of the HK corpus.



(d) SLDs and CDs of the HK corpus.



(e) SLRs and CRs of the UN corpus.



(f) SLDs and CDs of the UN corpus.

Figure 4.2: Percentages of SLR, CR, SLD and CD values greater than given threshold values for the DC, HK and UN Corpora.

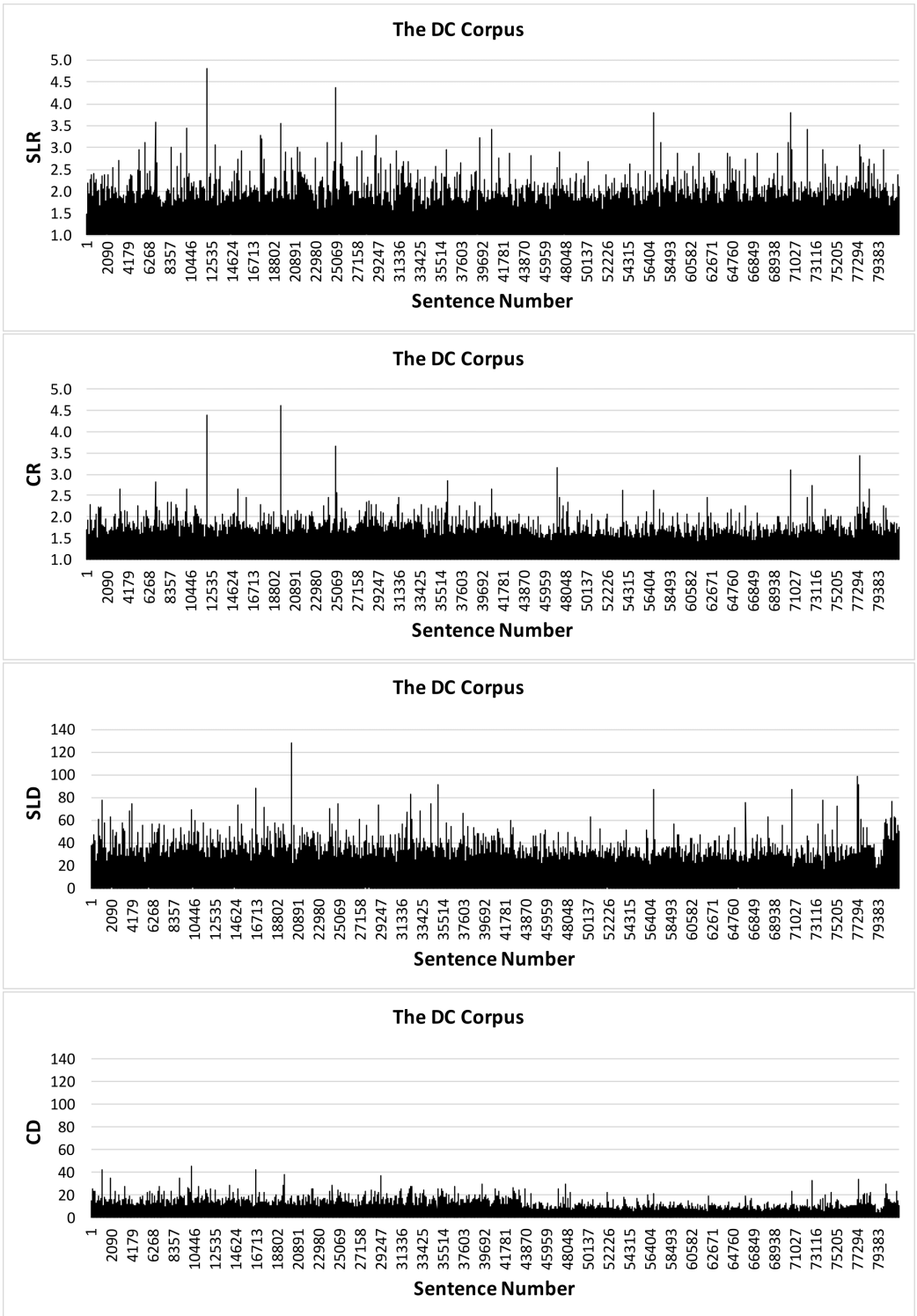


Figure 4.3: SLR, CR, SLD and CD values for the DC corpus.

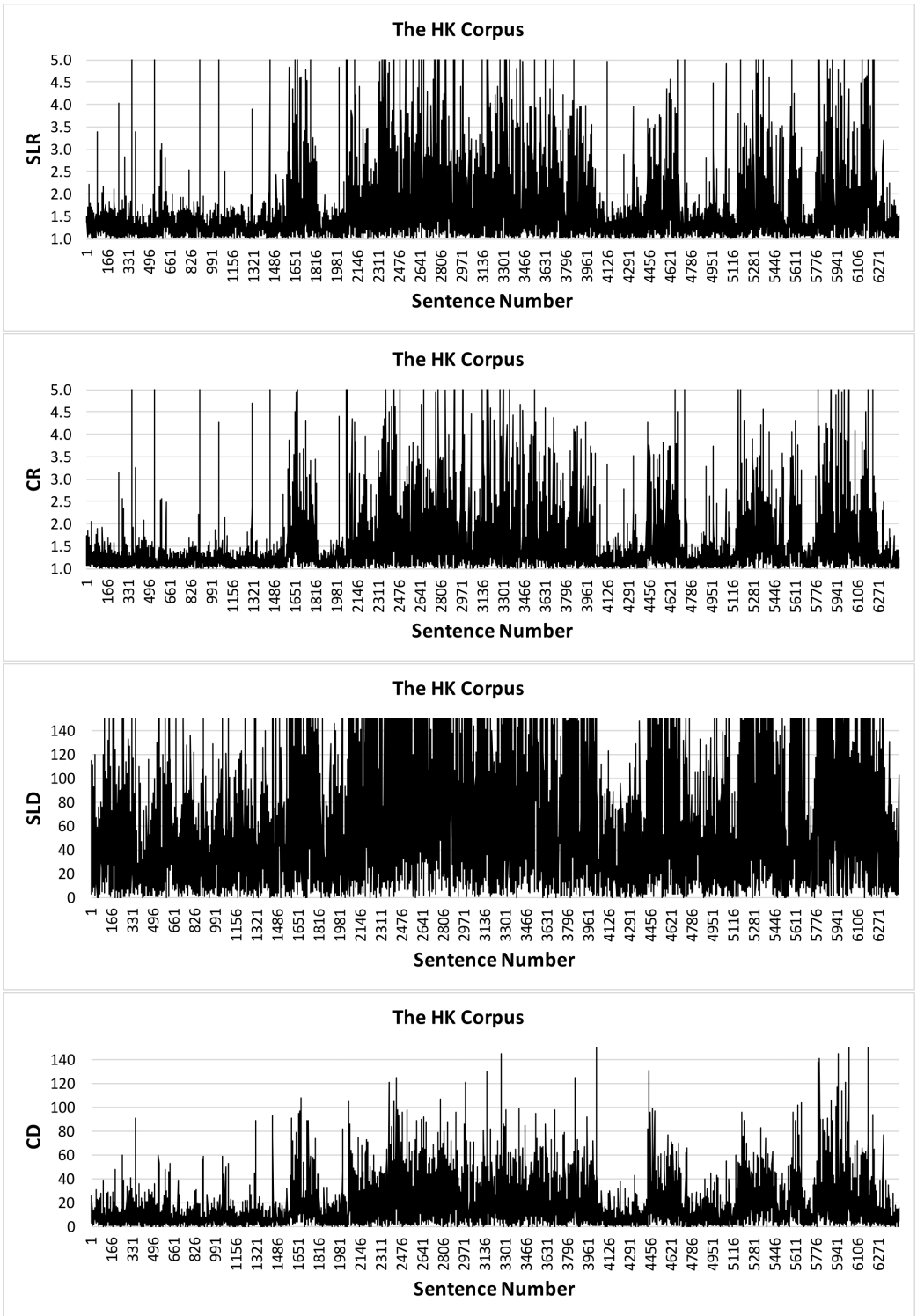


Figure 4.4: SLR, CR, SLD and CD values for the HK corpus.

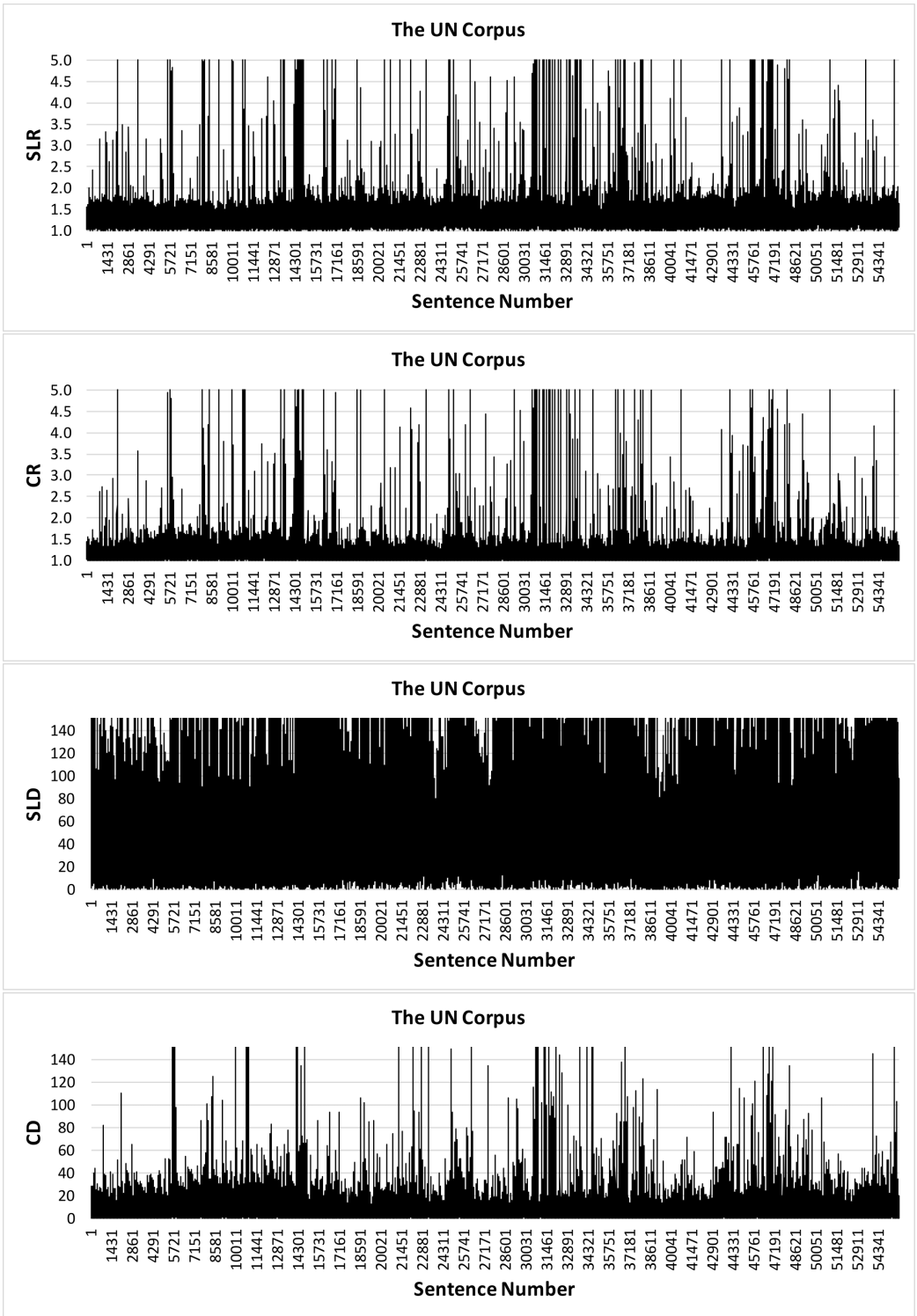


Figure 4.5: SLR, CR, SLD and CD values for the UN corpus.

ment quality than the DC, HK and UN corpora. A considerable amount of translations from the two corpora will not be acceptable by human evaluators. Due to the large amount of English characters that were found in the Chinese part of the KDE4 corpus, we ran a simple script to remove all English characters from the Chinese part of the KDE4 corpus and then the cleaned corpus is marked as "KDE4_C".

The KDE community is a collaborative team for developing free open source softwares and resources on different platforms. There have been a number of applications developed by KDE in vary fields such as education, communication, entertainment, etc (KDE, 2015). The KDE4 Corpus is a collection of localisation data for KDE version 4 in ninety two languages and contains over 75K files, 60M tokens and 8.8M sentence fragments (OPUS, 2015b). Because the KDE4 Corpus is not entirely parallel (Tiedemann, 2009), this experiment uses the Chinese-English part of the raw data and trimmed the part to make sure that they are entirely parallel.

The data of the GNOME Corpus was collected from museum labels, pharmaceutical leaflets and tutorial dialogues in over a hundred and eighty languages for studying aspects of discourse (OPUS, 2015a; Poesio, 2004). The GNOME Corpus contains over 113K files, 267M tokens and 58M sentence fragments (Poesio, 2015).

Table 4.6 shows the details of the Chinese parts of the KDE4 and GNOME Corpora. Both corpora for this experiment are encoded in UTF-8 encoding.

Table 4.6: Details of KDE4 and GNOME Corpora.

Corpus	**English**			**Chinese**		
	Size	**Files**	**Words**	**Size**	**Files**	**Characters**
KDE4	4.0MB	988	642,341	4.5MB	988	2,476,882
GNOME	35.6MB	2,066	55,826,630	33.5MB	2,066	16,578,330

Table 4.7: Comparing sentence lengths, code lengths and speed for uncleaned and cleaned KDE4 and GNOME corpora.

Corpus		KDE4	KDE4_C	GNOME
English	SL (bytes)	3,988,582	3,988,582	55,867,431
	CL (bytes)	998,956	998,956	8,679,855
	bpc	2.004	2.004	1.243
	Speed	5.6s	5.6s	70.3s
Chinese	SL (bytes)	4,495,894	3,109,123	46,863,083
	CL (bytes)	1,007,927	733,425	7,042,702
	bpc	1.794	1.887	1.202
	Speed	6.4s	3.7s	49.7s
SLR		1.127	1.283	1.192
CR		1.009	1.362	1.232

After separately compressing English and Chinese text files for the three corpora (KDE4, KDE4_C and GNOME), Table 4.7 shows the details of comparison, where the GNOME Corpus had a lower bpc value than others.

From Table 4.7 we can clearly see that the bpc values of the KDE4 Corpus for English and Chinese are significantly worse than the GNOME Corpus, meaning that the text in this corpus is much more compressible and that due to the high bpc values, the CL values have a worse estimation of cross-entropy. After manually checking, a number of unsatisfactory translations and mistranslations have been found from the Chinese part of the raw KDE4 Corpus. However, Table 4.7 unfortunately does not show an obvious improvement of the KDE4_C corpus, which means that the manual improvement has not been reflected by compressing the whole corpora.

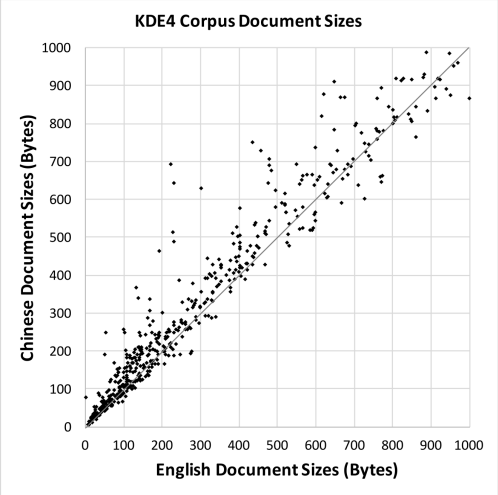
For the speed of the PPMD compression scheme, we can see that it took just over 70 seconds for compressing a large text that contains nearly 56 million sentences, which is satisfactory for this experiment.

Next, we evaluated the KDE4 and GNOME Corpora down to document level. Figure 4.6 shows four scatter plots for the two corpora document size distributions. Theoretically, the distributed plots of document code length for an ideal Chinese-English parallel corpus should be almost at the diagonal. However, Figures 4.6b and 4.6d are noisier than Figures 4.6a and 4.6c respectively, which more clearly shows that a number of unsatisfactory translations or mistranslations have been recognised by the PPM compression method. Particularly for Figure 4.6d, there are substantially more points far away from the diagonal.

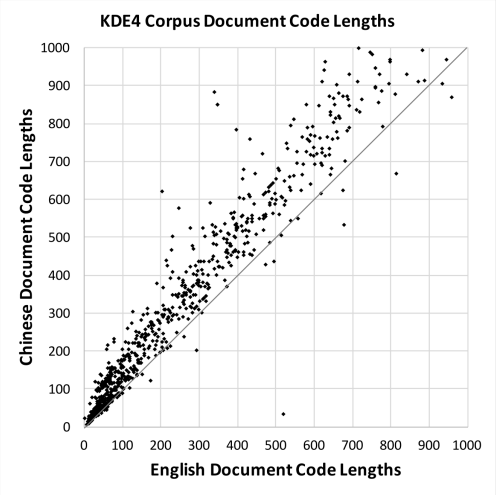
To ascertain further details of the quality of the three corpora, they have also been also compressed at the sentence level. As the KDE4 Corpus contains 138,613 sentences and GNOME has 1,336,282 sentences, however, due to the huge number of sentences, we cut down the number of sentences to the first 100,000 for each corpus. We used PPMD by training on the Brown and LCMC corpora with maximum order 5 and 6 respectively for English and Chinese text to compress all sentences.

Figure 4.7 shows the 100,000 sentences' sentence length and code length distributions for the KDE4 Corpus. From Figure 4.7a we can see that there are many noisy points which are even far away from the diagonal. For Figure 4.7b, most Chinese code length values are greater than English code length values. Moreover, there are few points at the diagonal and Figure 4.7b also shows that many translations probably are unsatisfactory, which has already been manually checked and verified.

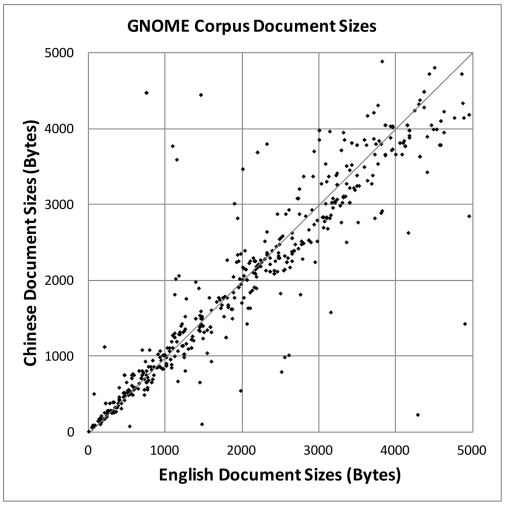
Figure 4.8 plots the distributions for the KDE4_C corpus. Compared to Figure 4.7a, Figure 4.8a shows fewer points for which the Chinese sentences and code lengths are longer than the English sentences and there are many points along the diagonal in Figure 4.8b because of the cleaning.



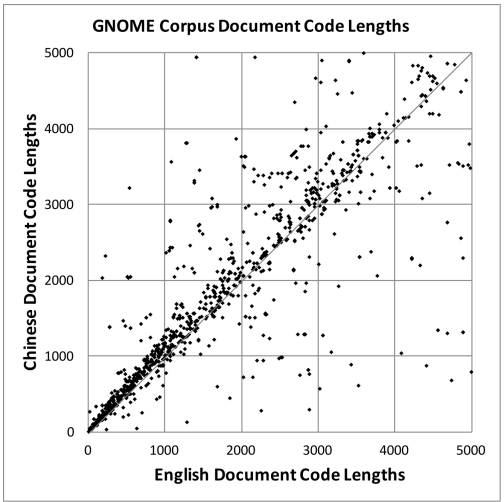
(a) KDE4 corpus document size distribution.



(b) KDE4 corpus document code length distribution.

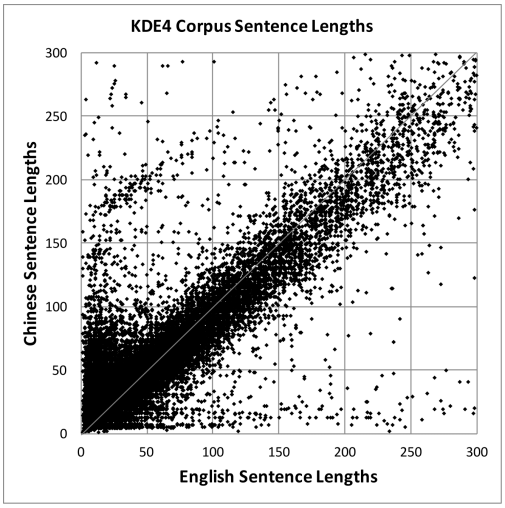


(c) GNOME corpus document size distribution.

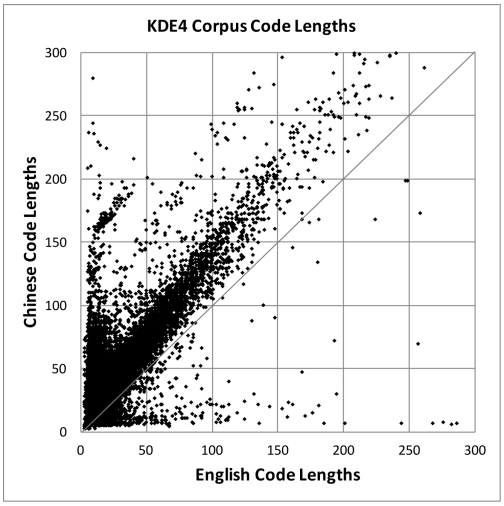


(d) GNOME corpus document code length distribution.

Figure 4.6: Scatter plots for the KDE4 and GNOME corpora comparing document sizes and PPM compression code lengths.

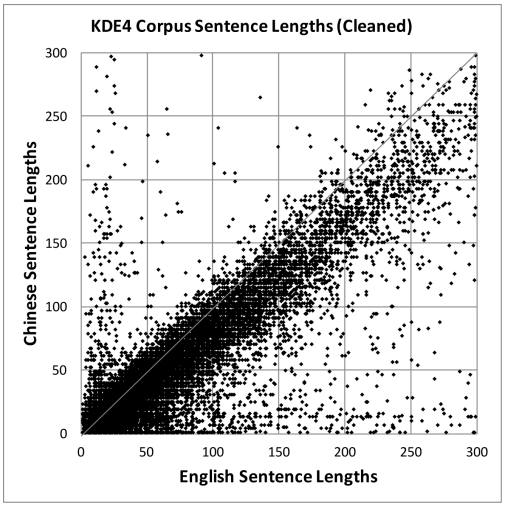


(a) KDE4 corpus sentence length distribution.

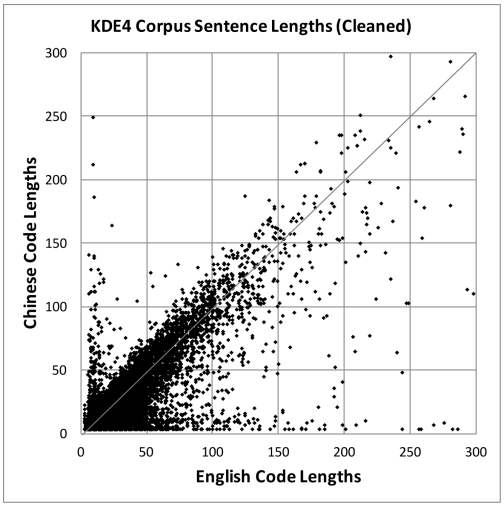


(b) KDE4 corpus sentence code length distribution.

Figure 4.7: Scatter plots for KDE4 corpus sentences comparing sentence lengths and PPM compression code lengths.

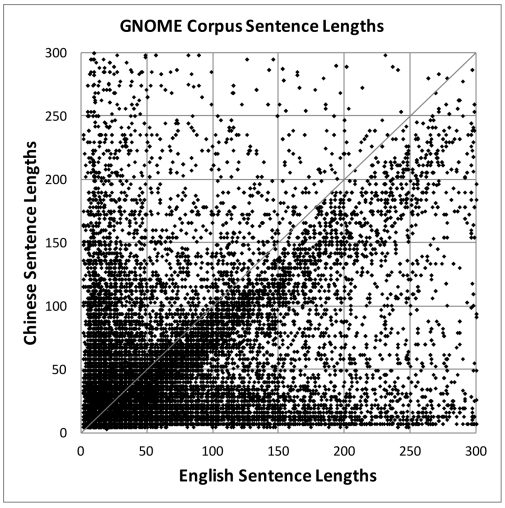


(a) Cleaned KDE4 corpus sentence length distribution.

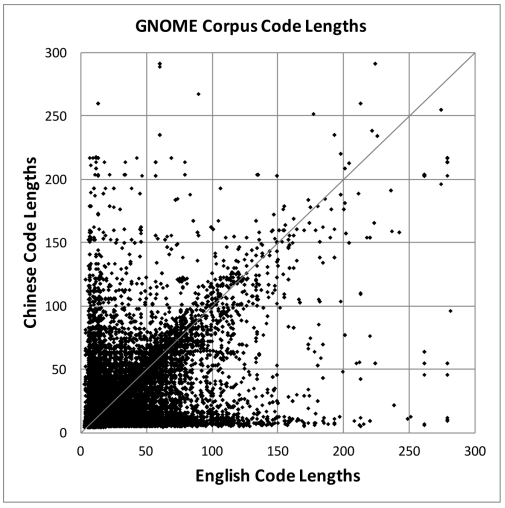


(b) Cleaned KDE4 corpus sentence code length distribution.

Figure 4.8: Scatter plots for KDE4_C corpus sentences comparing sentence lengths and PPM compression code lengths.



(a) GNOME corpus sentence length distribution.



(b) GNOME corpus sentence code length distribution.

Figure 4.9: Scatter plots for GNOME corpus sentences comparing sentence lengths and PPM compression code lengths.

The GNOME corpus is loaded by the same way and shown in Figure 4.9. Compared to both Figures 4.7a and 4.8a, Figure 4.9a shows that the GNOME Corpus is much more noisy for sentence lengths. The PPM compression code lengths from Figure 4.9b do indicate that there are probably more satisfactory translations included in the corpus but also a larger number of unsatisfactory translations or mistranslations.

Figure 4.10 shows percentages of how many sentence length ratios, code length ratios, sentence length differences and code length differences are greater than the values on the x-axis for the KDE4, KDE4_C and GNOME Corpora. From the CR curve of the KDE4 corpus in Figure 4.10a, over 52% of CR values are higher than 1.4 and 20% of CR values are higher than 2.0. Particularly, we can clearly see that the CR curve shows worse results than the SLR curve, which is an indication of that the corpus is of less quality. In

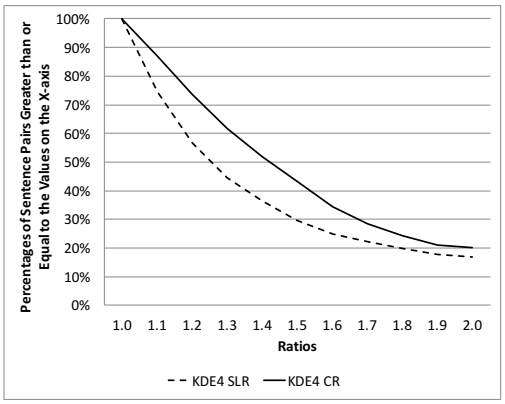
Figure 4.10b, the CD curve does not show a large difference with SLD and there are respectively 20% and 18% of CD and SLD values higher than 15.

Figures 4.10c and 4.10d highlight a significant improvement after the KDE4 corpus was cleaned. For the KDE4_C corpus, there are 30% of CR values and 41% of SLR values higher than 1.4 and there are 14% of CR values and 18% of SLR values higher than 2.0. Compared to Figure 4.10a, Figure 4.10c clearly shows that there is an improvement for the KDE4_C corpus. However, the CR curve in Figure 4.10c is still not so sheer as the DC and UN corpora in the previous experiment. There are still 3% of CD values and 14% of SLD values of the KDE4_C corpus higher than 15.

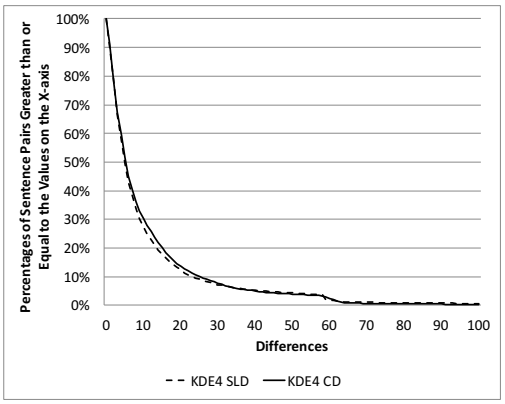
The CR curve in Figure 4.10e indicates that the quality of the GNOME corpus is probably better than the KDE4 corpus and slightly worse than the KDE4_C corpus. There are over 40% of CR values and 53% of SLR values greater than 1.4 and over 19% of CR values and 29% of SLR values greater than 2.0. Additionally, 14% of CD values and 34% of SLD values of the GNOME corpus are higher than 15 in Figure 4.10f.

Figures 4.11, 4.12 and 4.13 shows the details of the KDE4, KDE4_C and GNOME corpora using the four metrics (SLR, CR, SLD and CD) for each of the sentences in the corpora. An improvement can also be clearly seen by comparing the KDE4 corpus to the KDE4_C corpus from Figures 4.11 and 4.12, where most of the CR and CD values become lower in Figure 4.12

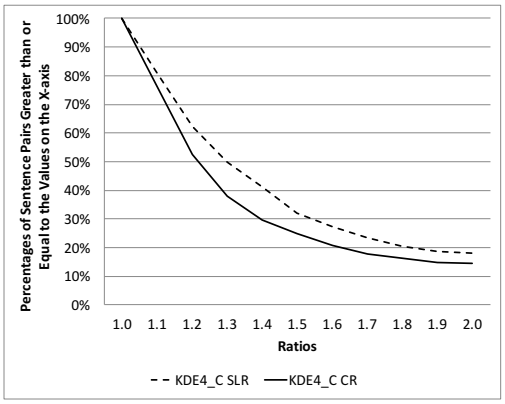
In summary, PPM-based code length measurement works better than sentence length-based measurement for evaluating the quality of original raw data of Chinese-English parallel corpora. Figures 4.7b, 4.8b and 4.9b are more effective than than Figures 4.7a, 4.8a and 4.9a for presenting the quality of the KDE4, KDE4_C and GNOME Corpora. The scatter plots in the figures are also effective at indicating the overall quality of the three corpora,



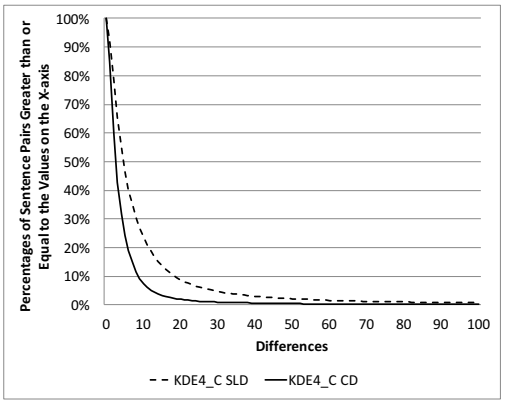
(a) SLRs and CRs of KDE4.



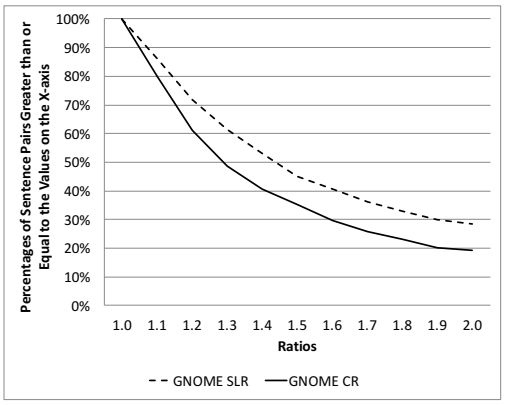
(b) SLDs and CDs of KDE4.



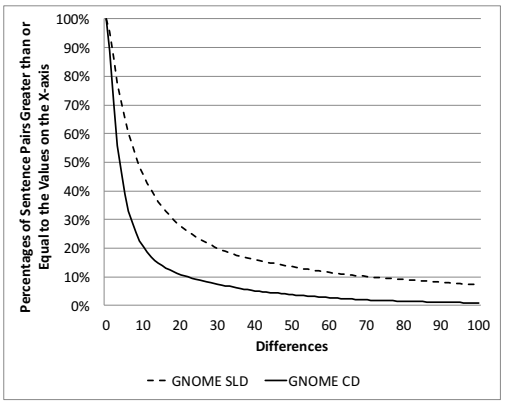
(c) SLRs and CRs of KDE4_C.



(d) SLDs and CDs of KDE4_C.



(e) SLRs and CRs of GNOME.



(f) SLDs and CDs of GNOME.

Figure 4.10: Percentages of SLR, CR, SLD and CD values greater than given threshold values for the KDE4, KDE4_C and GNOME Corpora.

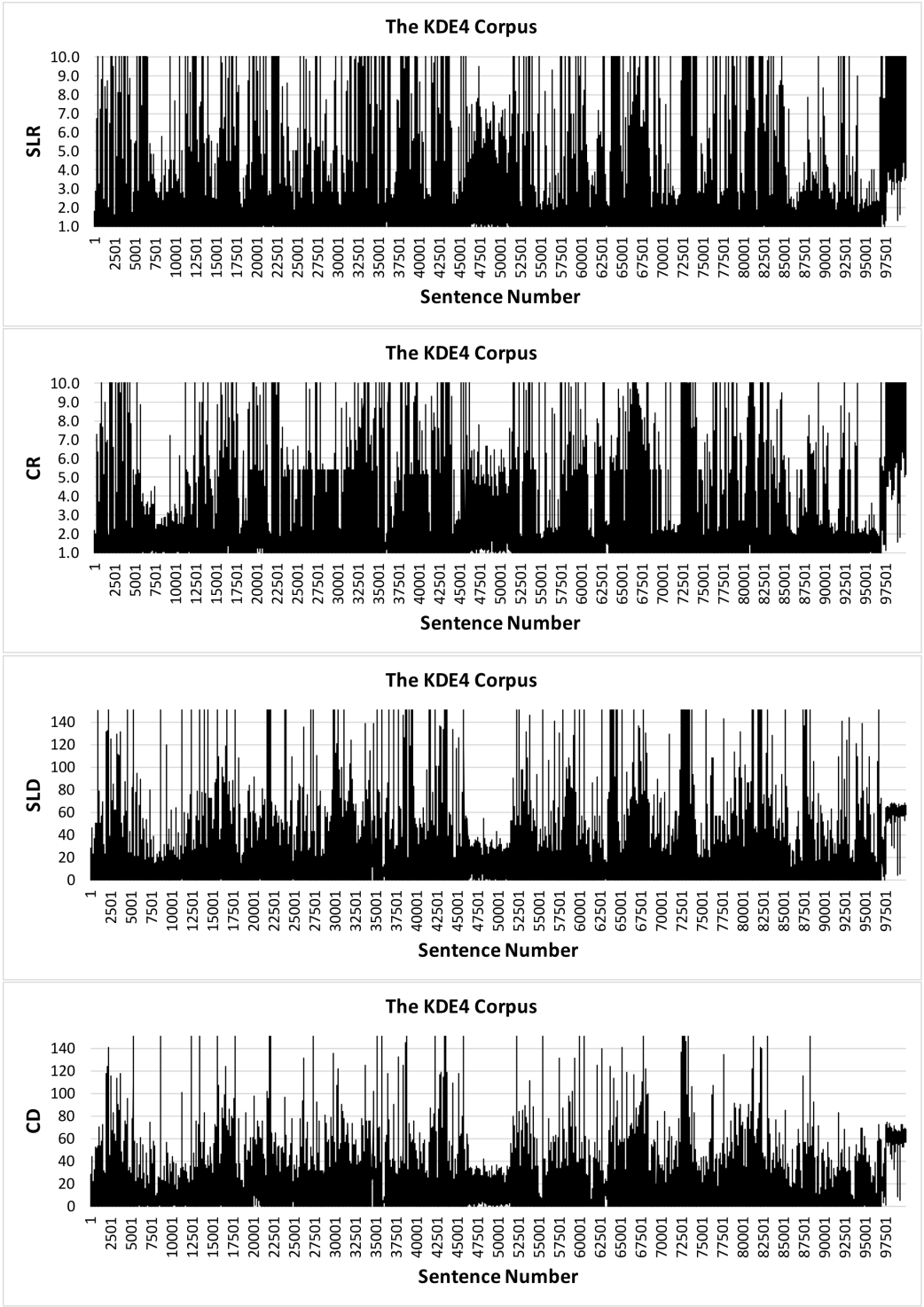


Figure 4.11: SLR, CR, SLD and CD values for the KDE4 corpus.

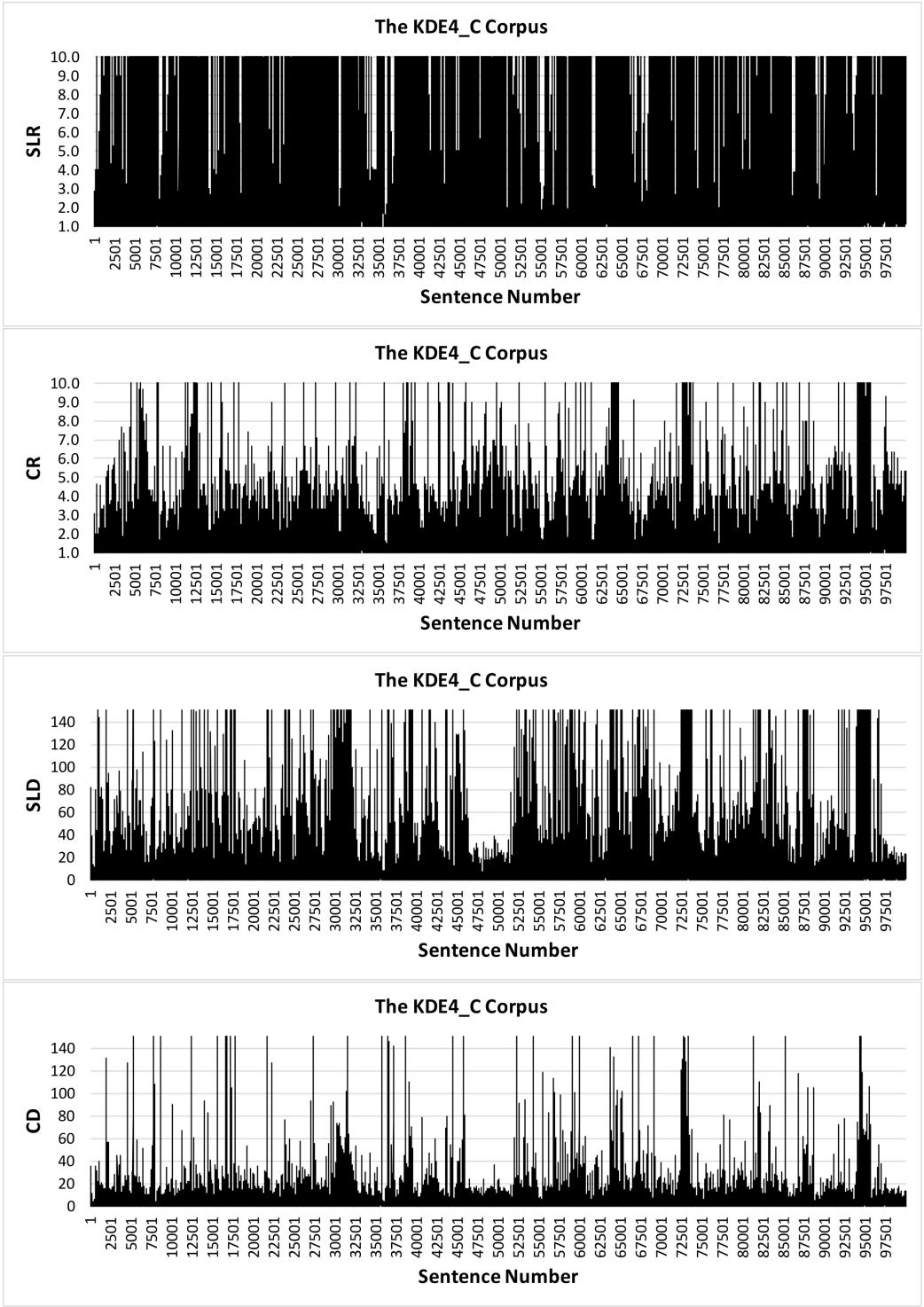


Figure 4.12: SLR, CR, SLD and CD values for the KDE4_C corpus.

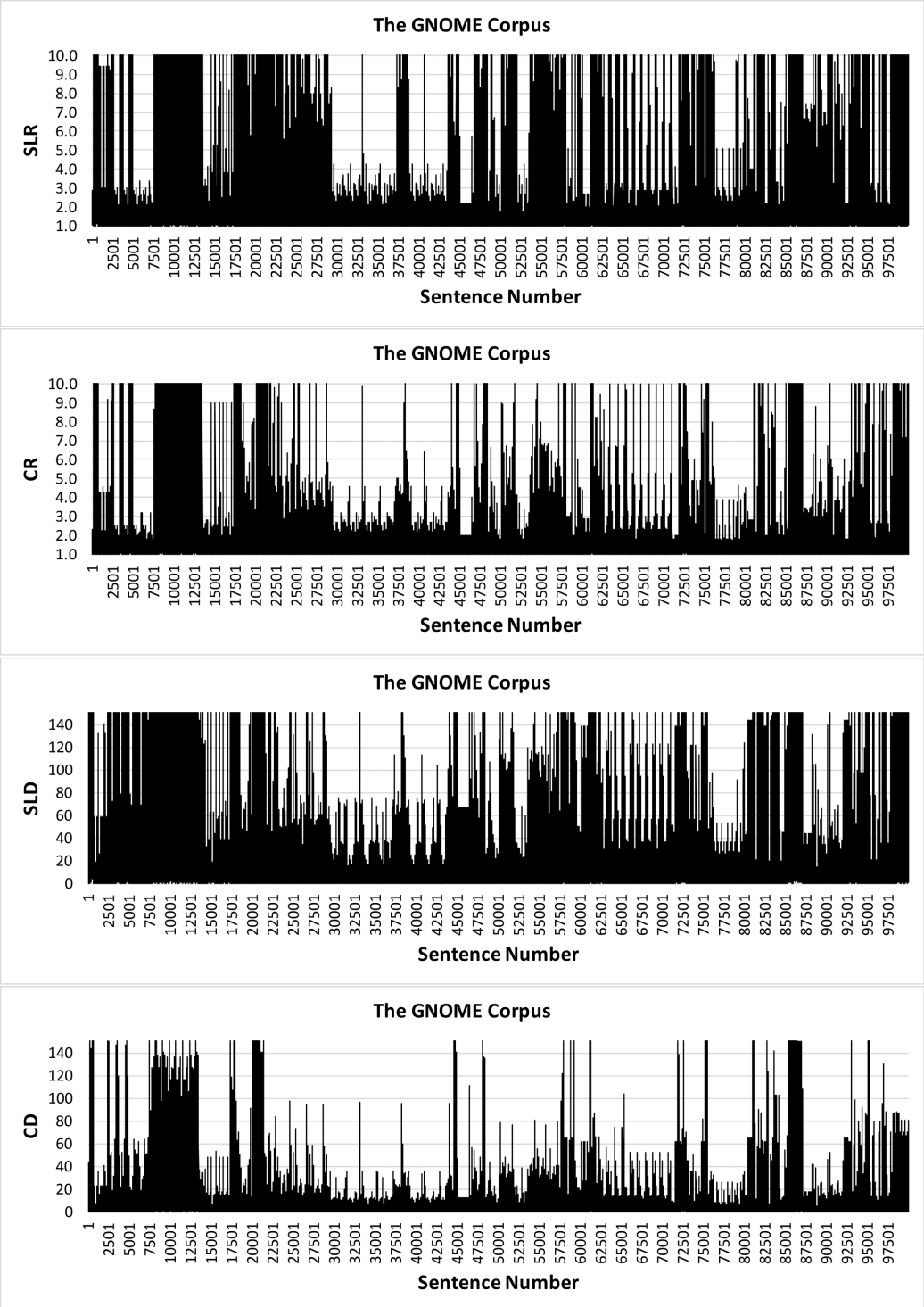


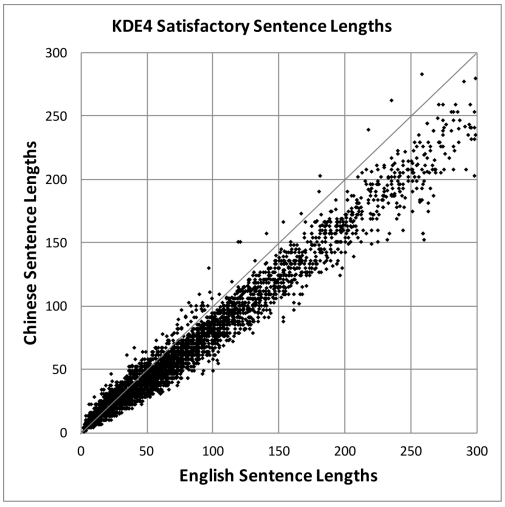
Figure 4.13: SLR, CR, SLD and CD values for the GNOME corpus.

especially for the improvement of the KDE4_C corpus.

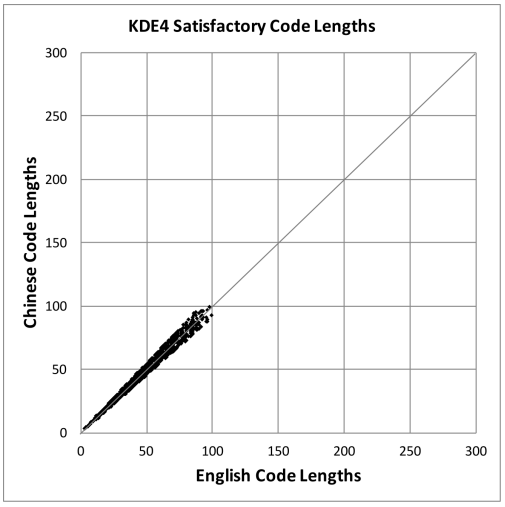
As a further method for comparing the effectiveness of the SLR and CR metrics, we have also manually selected satisfactory and unsatisfactory sentence pairs (see Appendices I, II, III and IV for examples) from the KDE4_C and GNOME corpora and analysed them in the following paragraphs. For the judgement of satisfactory sentence pairs, we selected all sentences with the CR values lower than 1.1 before manually checking to show how the SLR values distribute for those satisfactory translations. For unsatisfactory sentence pairs, we selected all sentences with the SLR values lower than 1.1 before manually checking to show how the CR values distribute for those unsatisfactory translation.

In the manually-selected sentences, there are in total 19,781 sentence pairs judged as satisfactory translations and 14,422 sentence pairs as unsatisfactory translations or mistranslations from the KDE4_C corpus and the comparison of distributions are shown as Figure 4.14. Figure 4.14a presents the satisfactory sentence pairs by sentence length distribution and Figure 4.14b shows the pairs by code length distribution, where we can see that Figure 4.14b presents more accurate information for satisfactory translations.

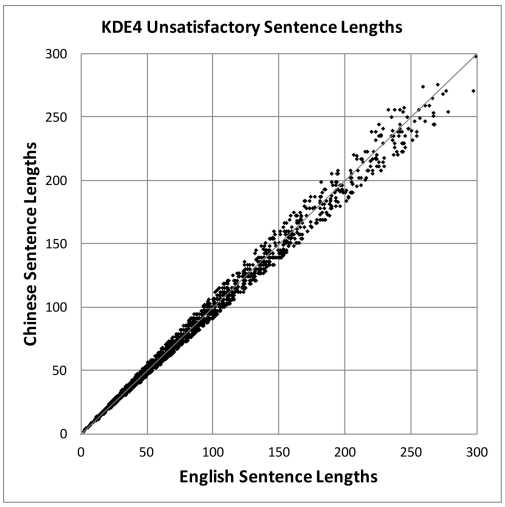
Moreover, a similar evaluation has been experienced for the GNOME corpus. There are 17,093 satisfactory Chinese-English sentence pairs and 10.240 unsatisfactory translations or mistranslations have been manually selected from the GNOME corpus. Figure 4.15 shows the comparison of distributions between sentence length and code length. From Figure 4.15a we can see that there are more noisy points than Figure 4.14a, which seems that there are more unsatisfactory translations in the satisfactory group. However, Figure 4.15b shows a very different result where there is no more noisy points for the compression code lengths. Clearly, Figure 4.15d indicates the presence



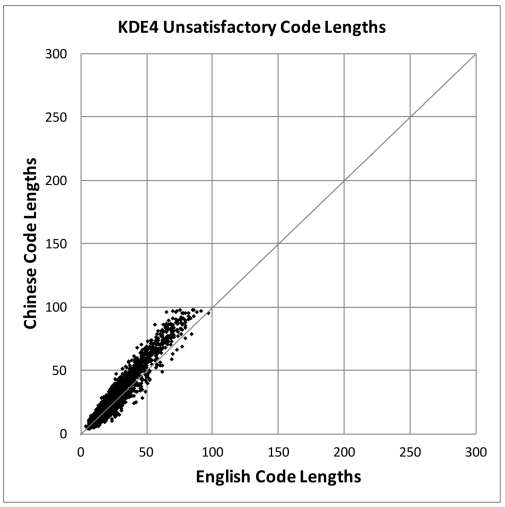
(a) SL distribution for the satisfactory part of the KDE4_C corpus.



(b) Sentence CL distribution for the satisfactory part of the KDE4_C corpus.



(c) SL distribution for the unsatisfactory part of the KDE4_C corpus.



(d) Sentence CL distribution unsatisfactory part of the KDE4_C corpus.

Figure 4.14: Scatter plots of distributions for satisfactory and unsatisfactory parts of the KDE4_C corpus for sentence-based measurements.

of unsatisfactory translations or mistranslations better than Figure 4.15c because of more noisy points.

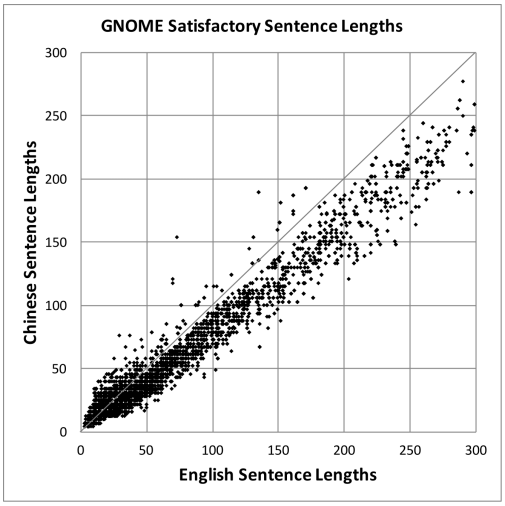
All the scatter plots of Figures 4.14 and 4.15 indicate that PPM compression code length metric (CR) performs better than sentence length metric (SLR) for evaluating the quality of translations and parallel corpora.

In summary, out experimental results show that code length metrics are better at identifying the quality of the corpora than the sentence length metrics. Choosing raw data for the KDE4 and GNOME Corpora with no preprocessing to limit the quality was also the reason to show more clearly that PPM-based code length metrics works for corpus evaluation.

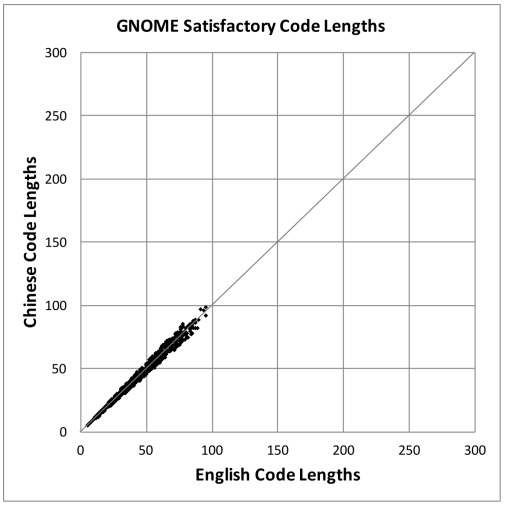
4.6 Conclusion

The first experiment for the evaluation of the three testing corpora—DC, HK and UN which were built by different ways with different sizes—show that using the PPM compression code length metric is an effective method to evaluate a parallel corpus or compare the quality of two or more parallel corpora. The experimental results were compared in different ways to describe the different features among the three testing corpora. The experiment was divided into two main steps. The first step was to compress the whole corpora and compare the overall results to obtain an initial conclusion. The second step produced further conclusions after compressing all sentences individually by using PPM code length method and comparing the three testing corpora according to sentence code length ratios.

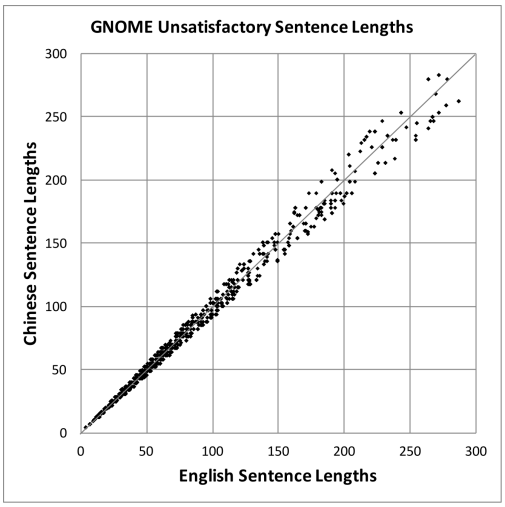
In fact, the three testing corpora have already been manually reviewed and analysed. The DC corpus is not with natural sequence but has the fewest unsatisfactory translations or mistranslations, and therefore the DC corpus



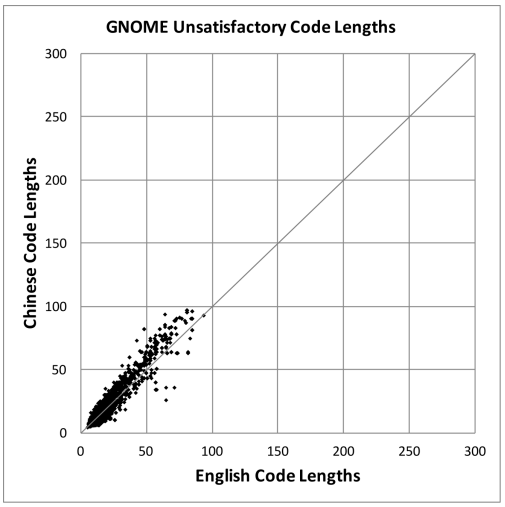
(a) SL distribution satisfactory part of the GNOME corpus.



(b) Sentence CL distribution satisfactory part of the GNOME corpus.



(c) SL distribution unsatisfactory part of the GNOME corpus.



(d) Sentence CL distribution unsatisfactory part of the GNOME corpus.

Figure 4.15: Scatter plots of distributions for satisfactory and unsatisfactory parts of the GNOME corpus for sentence-based measurements.

should be least noisy. The HK corpus is the smallest corpus with natural sequences but is most noisy because there are a high percentage of unsatisfactory translations or mistranslations. The UN corpus is the largest corpus with natural sequence. There are overall the most satisfactory translations in UN Corpus and a reasonable small percentage of noise. In terms of translation quality, the DC corpus presented a better quality than the other two corpora with the HK corpus having the lowest quality. The results concluded from the first experiment matched the manual reviews of the three testing corpora.

The KDE4 and GNOME corpora have been automatically collected from the Internet and as a result there is no quality guaranteed. That is why we processed them in the second experiment. The experiment at results clearly showed that the quality of the KDE4 and GNOME corpora is worse than the DC, HK and UN corpora. Although the GNOME corpus performed slightly better than the KDE4 corpus, they are still not satisfactory to use unless any further cleaning is done on them. From the second experiment, we have also seen that PPM-based compression method is effective for recognising unsatisfactory Chinese-English parallel corpora and even unsatisfactory part from the corpora. Therefore, we have reason to believe that PPM is also effective for automatically creating a new Chinese-English parallel corpora with high quality.

Especially for those "quick and dirty" parallel corpora automatically collected from various untrustworthy sources, an effective corpus evaluation method is more necessary for fast checking the overall quality before or during preprocessing. In the aspect of general assessment of the corpora, the code length values are more sensitive for the alignment quality especially for the case of misalignment with similar sentence lengths. Code length-based

graphs that we have already shown can also more clearly present the quality distribution of the corpora than sentence length-based and they are also valuable for filtering out noise for the quality improvement by a threshold value. Moreover, based on experiments in this chapter, it is reasonable to expect that the PPM code length method will also be effective for improving and adjusting the quality of parallel corpora and for creating parallel corpora. In the next chapter, we investigate using the PPM-based code length metrics for automatic creation of new parallel corpora.

Chapter 5

Automatic Creation of New Parallel Corpora

5.1 Introduction

Evaluation of a corpus is an important initial process that analyses the corpus prior to its application to some natural language processing task (in this case, machine translation) in order to determine that the corpus is essentially fit for purpose. Ideally, a parallel corpus should be of the highest quality, with very few of the translation pairs deemed unsatisfactory by a human translator (such as sentences in each language identified by the corpus as being co-translations of each other). Alkahtani et al. (2014) have discussed a new parallel corpus alignment strategy for Arabic-English using a new parallel corpus that contains over 58 million words. In this chapter, we will adopt a similar approach, and extend and adapt it to the task of automatic creation of Chinese-English parallel corpora.

The aim of the experiment in this chapter was to investigate the CR and SLR metrics further, specifically to see how effective they might be at

evaluating the quality of an existing parallel corpus of Chinese-English. The final objective of this chapter is to automatically download text from a web source then to evaluate its quality in order to automatically create a high quality corpus. For the corpus being evaluated in this experiment, we use the United Nations Parallel Corpus (UN Corpus) that was purchased from the Linguistic Data Consortium (LDC) (the catalog entry can be found at `https://catalog.ldc.upenn.edu/LDC2013T06`). The UN corpus consists of the United Nations parliamentary documents that were produced between the year of 1993 and 2007 in six languages (Arabic, Chinese, English, French, Russian and Spanish) (LDC, 2014). This experiment uses the English and Chinese parallel text only.

Part of this chapter is based on a co-authored paper that was published at the Proceedings of Computer Science & Information Technology (CS & IT) (Alkahtani et al., 2014). The results for that paper have been replaced with results for Chinese-English documents rather than Arabic-English documents.

5.2 Corpus Preparation

We use the way the text files in the UN corpus are named to concatenate the files together into separate partitions. Each file in the corpus with the same common prefix (such as "A_", "APLC_", "BWC_", "CAC_") was concatenated together to form a smaller set of 62 files. The purpose of this was to evaluate if parts of the corpus are of lesser quality than other parts. For the three largest partitions – for files with prefixes "A_", "E_" and "S_" – where the partition size was significantly larger than the other partitions, we created 17 further sub-partitions (resulting in 79 partitions in total) using

the first two file name prefixes instead of just the first (i.e. "A_A_", "A_B_", "A_C_", etc.). Table 5.1 lists the top 25 partitions according to size that were created from the corpus.

We also created a smaller testing corpus (which we will call UN1) in order to conduct further experiments as detailed below. This testing corpus was created by randomly selecting 10,000 translations that were manually judged satisfactory by a human translator and a further 10,000 translations that were found to be unsatisfactory. Higher quality mis-translations involving mostly poor language were deliberately chosen, rather than including grossly mismatched sentence pairs where a significant proportion of one of the sentences was missing. This was in order to make the identification task described below more difficult.

Table 5.2 shows 10 examples of translations in the UN1 testing corpus that were deemed to be unsatisfactory. These examples would easily be recognised by a Chinese-English bilingual speaker as being clearly incorrect.

5.3 Corpus Examination

We first describe some initial compression-based experiments we conducted on the UN and UN1 corpora to determine their quality. The PPM (PPMD) compression method in these experiments uses (as in previous chapters) for English text the Brown corpus as the training corpus with maximum order 5 whereas for the Chinese UTF8 encoded text, the LCMC corpus is used as the training corpus with maximum order 6.

We initially examined the first 20 English and Chinese documents for the year of 2007 that can be found in the UN corpus. These documents were published on the UN website and extracted directly from the source.

Table 5.1: The top 25 partitions that were created from the UN Corpus ordered by size.

Prefix(es)	**Sizes**	**English Characters**	**English Words**	**Chinese Characters**	**Chinese Words**
A_D	482MB	182,591,195	31,183,067	51,385,638	28,163,710
A_B	456MB	172,823,842	29,706,469	48,169,188	26,543,392
A_C	404MB	152,948,620	26,123,544	42,835,382	23,531,898
A_A	390MB	147,545,315	25,417,884	41,751,213	22,736,388
E_A	388MB	146,632,474	24,919,737	41,228,420	22,623,052
S_F	369MB	137,050,357	23,979,849	37,250,238	20,839,343
E_B	305MB	115,063,544	19,489,858	32,451,048	17,737,611
E_C	294MB	109,420,902	18,725,728	42,345,548	17,182,463
A_I	279MB	103,478,773	17,962,377	34,787,417	16,540,533
CEDAW	274MB	103,067,150	17,963,191	28,942,607	16,285,562
A_H	261MB	98,130,208	16,967,474	27,162,542	14,942,692
A_E	255MB	95,521,423	16,338,965	27,107,643	14,723,671
S_D	247MB	91,746,483	15,833,752	27,321,521	14,386,321
A_F	202MB	75,674,398	12,772,757	20,549,036	11,312,694
E_D	196MB	73,088,853	12,296,777	20,840,335	11,447,711
A_G	192MB	71,811,986	12,111,291	19,572,002	10,661,344
TD	172MB	64,234,475	10,672,470	18,396,317	10,008,012
S_C	140MB	51,391,726	8,847,194	14,369,349	7,881,988
S_B	117MB	42,278,342	7,327,762	12,077,044	6,532,908
FCCC	105MB	37,985,436	6,529,493	11,432,884	6,282,123
A_J	97MB	34,941,608	5,927,423	9,333,900	5,221,124
UNEP	93MB	33,997,730	5,717,759	10,963,708	5,708,433
S_A	87MB	30,778,915	5,353,305	8,783,850	4,730,451
DP	80MB	29,451,737	4,933,348	8,655,000	4,789,897
JOURNAL	77MB	28,662,921	5,072,303	9,503,428	4,489,532
...	...	...	...	...	...
TOTAL	6.29GB	2,405,638,636	412,335,751	698,657,261	372,559,741

Table 5.2: Sample of unsatisfactory sentence pairs that appear in the UN1 testing corpus. All examples are misaligned where the Chinese sentences present totally different meanings.

	Language	Sentence
1	English	The Conference of the Parties may wish to consider the report.
	Chinese	缔约方大会收到了秘书处分别就下列事项编制的若干说明。
2	English	Over the past two years, as described below.
	Chinese	其中主要是非政府组织和研究所。
3	English	The results of the actuarial valuation as at 31 December 1999.
	Chinese	以便保护国际组织前工作人员的权利。
4	English	Level officials from the Russian Federation.
	Chinese	法庭认为这项限制规定也牵涉类似的申请。
5	English	I thank members for their cooperation.
	Chinese	我们是否能够保护自然环境?
6	English	It must have a human face, or it will not be for us.
	Chinese	因此，本组织必须进行改革。
7	English	This morning, I call on the observer of the Holy See.
	Chinese	冰岛、爱尔兰、以色列、意大利、日本、哈萨克斯坦、肯尼亚。
8	English	Up to the general elections envisaged for 2005.
	Chinese	包括各主要利益攸关者都参加这一对话。
9	English	As well as on the request made by the Niger.
	Chinese	第五委员会内部已经形成共识。
10	English	File the application in the name of the said staff member.
	Chinese	其时限应延长至一年。
11	English	The sources of conflict and war are pervasive and deep.
	Chinese	冲突和战争的根源既普遍又深远。
12	English	The United Nations has not closed its door.
	Chinese	联合国没有关闭它的大门。
13	English	The proposal has since been implemented.
	Chinese	这个建议已经实施。

Table 5.3: Compressed Chinese and English document sizes for the first 20 document pairs in the UN corpus for the year 2007.

File Number	English File Size (Bytes)	English Code Length	Chinese File Size (Bytes)	Chinese Code Length	File Size Ratio	Code Length Ratio
1	5123	1179	3549	1195	1.444	1.014
2	4285	1006	2794	921	1.534	1.092
3	4838	1107	3378	1087	1.432	1.018
4	3201	777	2258	782	1.418	1.006
5	4734	1182	3548	1203	1.334	1.018
6	4174	921	2758	850	1.513	1.084
7	4629	1037	3234	1008	1.431	1.029
8	23870	5367	17468	5114	1.366	1.049
9	11019	2518	7902	2469	1.394	1.020
10	6882	1584	4762	1481	1.445	1.070
11	14231	3353	10227	3111	1.392	1.078
12	83536	19303	64637	18100	1.292	1.066
13	5600	1299	4287	1301	1.306	1.002
14	381	128	277	131	1.375	1.023
15	598	172	440	178	1.359	1.035
16	526	145	370	135	1.422	1.074
17	640	246	485	235	1.320	1.047
18	24512	5846	19848	5606	1.235	1.043
19	30182	6312	21499	5329	1.404	1.184
20	5001	1264	3697	1101	1.353	1.148

Therefore, they provide an excellent means for testing the issues involved in automatically extracting text from the web. Table 5.3 lists the file sizes and the code length ratio (CR) results for these documents (in the last column of the table). From the results, we can see that the code lengths match very closely with the greatest CR value being just above 1.1. In contrast, the ratio of file sizes shown in the second to last column produces a much wider range of values. This is an example of one way a quick check can be done at the document level to evaluate the quality of the documents in a corpus using the CR measure.

Table 5.4 shows the compression code length values for both English and Chinese for the text files of the first 25 largest partitions. The rightmost column denotes the CR values. The results show that the CR values range from 1.156 for the UNEP partition up to 1.365 for the JOURNAL partition. The later value is a bit high compared to the other CR values, and indicates that the quality of this partition should be manually checked further. However, overall the CR values at the document level are satisfactory and there is no strong evidence that these partitions might include too many bad translations.

Then we proceeded to examine each of the documents in the UN corpus by compressing them sentence by sentence. As a quick check, we first extracted 20 sample sentences from the corpus and compressed them by hand to see if the code length values being generated were consistent. The purpose of these preliminary experiments were to determine how effective the primed PPM compression code length method was as a sentence matching metric. A key requirement of using the CR metric is that the compression code lengths in the two different languages should be the same for sentences that are co-translations of each other. The intuition is that if the sentences are satisfactory co-translations, then they should convey exactly the same amount of information. Since compression code length is an effective method for measuring information (see Teahan (1998) for several references), then we would expect that roughly 50% of the compression code lengths of sentences in one language to be longer than compression code lengths of sentences in the other language, and vice versa.

Table 5.4: Compression results for the first 25 largest partitions of the UN Corpus.

Partition	**English Code Length**	**Chinese Code Length**	**Code Length Ratio**
A_D	32,367,236	26,182,996	1.236
A_B	30,572,355	24,849,839	1.230
A_C	26,956,346	21,870,356	1.233
A_A	26,129,025	21,205,135	1.232
E_A	26,554,056	21,873,781	1.214
S_F	22,747,892	18,502,638	1.229
E_B	20,505,794	16,887,199	1.214
E_C	19,103,395	15,976,374	1.196
A_I	18,651,238	15,524,779	1.201
CEDAW	18,926,742	15,653,662	1.209
A_H	16,894,934	13,895,469	1.216
A_E	17,092,655	13,950,896	1.225
S_D	16,747,725	13,597,582	1.232
A_F	12,555,823	10,229,730	1.227
E_D	12,774,631	10,702,209	1.194
A_G	11,830,764	9,880,600	1.197
TD	11,199,343	9,545,721	1.173
S_C	9,346,340	7,583,846	1.232
S_B	7,683,355	6,268,739	1.226
FCCC	6,147,395	5,076,222	1.211
A_J	5,855,547	4,745,225	1.234
UNEP	5,952,242	5,149,506	1.156
S_A	5,485,807	4,488,439	1.222
DP	5,081,918	4,389,021	1.158
JOURNAL	3,684,739	2,698,862	1.365

5.4 Sentence Length Analysis

Clearly, this correlation would not be expected for sentence lengths. It is quite common that English sentences are shorter than their co-translation counterparts other languages. However, this should not be the case for compression code lengths if our intuition about the correlation between information is correct. If we find that the compression code lengths do not correlate, then the reason for this is more likely to be as a result of a less effective compression algorithm being used for one language resulting in a less accurate estimate of the information contained in the sentence.

The initial results we obtained on some sample sentences randomly selected from the satisfactory translations of Year 2007 of the UN corpus are shown in Table 5.5 and are presented here in order to illustrate how the process works. The results include both the raw sentence lengths and compression code length values for character-based and byte-based Chinese sentences in the second and third columns. For example, a Chinese sentence—今天真热。—includes five Chinese characters including a full stop. The sentence length based on characters is obviously 5; whereas when using UTF-8 encoding for Chinese text, the sentence length based on bytes will be $5 \times 3 = 15$ because with UTF-8 encoding, each Chinese character in this example sentence requires 3 bytes. Usually the ratio between Chinese characters and bytes in the UTF-8 encoded text can range from 1:2 up to 1:3 depending on the number of non-Chinese characters in the text. The code length values in bytes are shown in the third and fourth columns for the English and Chinese sentences that were obtained by using PPMD with max order 5 for the English text and order 6 for the Chinese text.

The results show that the Chinese character sentence length values are always significantly lower than their English byte sentence length values in-

Table 5.5: Compression results of some sample sentences taken at random from the UN corpus.

Sentence ID	Sentence Length			Code length (bytes)	
	English	Chinese (Char-based)	Chinese (Byte-based)	English	Chinese
1	165	39	111	45	44
2	128	59	109	51	66
3	96	39	105	32	38
4	207	51	151	46	45
5	150	31	91	37	31
6	109	28	76	27	30
7	113	30	78	29	32
8	90	24	70	24	25
9	61	21	61	16	17
10	95	25	73	21	27
11	83	23	67	27	27
12	157	56	152	56	64
13	111	30	88	26	26
14	164	32	94	35	28
15	138	39	115	32	31
16	213	61	173	55	54
17	103	31	91	26	27
18	92	22	64	24	23
19	60	18	52	17	19
20	69	17	49	19	18

dicating that counting Chinese characters is an unsuitable way of measuring sentence lengths in order to calculate the SLR metric. When we measure the length of the Chinese sentences using bytes instead, we also see that there still is significant variation between the sentence length values between the two languages in many cases. In contrast, the variation in code length values is not as great.

5.5 Compression Code Length Analysis

As shown by Table 5.1, the size of the partitions in the UN corpus are quite large. The total size of the corpus is over 6 Gigabytes with over 57 million sentences. Although it was relatively easy to generate the compression results above for the entire partition documents, it quickly became apparent that generating results sentence by sentence was going to be more problematical. We estimated that generating compression code lengths for all the sentences in the UN corpus would take too long to process. Therefore we chose at most 1000 sentence pairs taken from the beginning of each partition in the corpus instead.

We determined what percentage of times the sentence length (measured both in characters and bytes) and code length values were greater for each language and a summary of the results are shown in Table 5.6 for the top 25 partitions of the corpus. From the sentence length results shown in columns two and three of the table, we can see that English sentences are always longer than character-based Chinese sentences. For some partitions such as "A_B", "CTBT-ART", "HCR", etc., 100% of English sentences are longer. For the byte-based sentence length measurement, fewer English sentences are longer than Chinese character-based, but we can still see that too many English sentences are longer than Chinese. The results show that in most cases (98% and above), if sentence length is measured by counting characters, then English sentences are longer. When sentence length is measured using bytes, there is a slightly greater percentage of Chinese sentences that are greater, but this is usually only 5 to 10% of cases. When using PPM to generate code lengths, however, the comparison between languages are more evenly spread around 50% as hoped.

This illustrates an excellent way of evaluating the quality of the different

partitions. Where the percentage of code lengths for one language is significantly greater than for another language (greater than 60%, for example), then this provides strong evidence that the sentences in this partition need to be examined further to determine their actual quality. For example, as shown in the table, over 75% of the code lengths for Chinese sentences in the UNEP partition are greater than their English counterparts. In several other smaller sized partitions not shown in the table, the disparity was even greater. In one partition (DL), 100% of the Chinese sentences were greater; in another partition (T), 0% of the Chinese sentences were greater. Two other partitions had a high percentage for Chinese sentence code lengths being greater (93.7% for BWC and 75.4% for UNEP) but generally, most partitions produced results around 50%. After investigation of these suspicious partitions, we found that they contained many unsatisfactory translations and even mistranslations. We were surprised because we did not think that there would be so many poor or erroneous translations in the corpus, but on the other hand, this did provide strong evidence to support the code length based approach to alignment that we had been adopting in this chapter.

5.6 Experiment 1: Parallel Corpus Quality Evaluation

Next, we investigated how well the sentence length ratio (SLR) and code length ratio (CR) metrics performed at identifying the satisfactory and unsatisfactory sentence pairs that we had placed into the UN1 testing corpus. Table 5.7 shows the accuracies for the identification task for different threshold values. If the sentence length ratio or code length ration exceeded the threshold value, then the sentence pair would be deemed to be unsatisfactory,

Table 5.6: Sentence length and code length comparison for the top 25 largest partitions of the UN Corpus where 1000 sentence pairs were taken from the beginning of each partition. The percentage values indicate what percentage of the sentences were longer for the English sentence rather than the Chinese sentence and vice versa.

Partitions	**Sizes**	**Character SL**		**Byte SL**		**Code length (CL)**	
		English	**Chinese**	**English**	**Chinese**	**English**	**Chinese**
A_D	358KB	99.60%	0.20%	87.60%	11.60%	49.80%	43.30%
A_B	338KB	100.00%	0.00%	79.80%	19.60%	45.30%	46.90%
A_C	352KB	99.50%	0.40%	91.70%	7.60%	42.70%	50.10%
A_A	296KB	96.70%	2.90%	88.00%	10.50%	35.90%	51.10%
E_A	340KB	100.00%	0.00%	96.70%	2.70%	52.80%	41.10%
S_F	378KB	99.80%	0.20%	99.80%	0.20%	50.00%	39.40%
E_B	319KB	100.00%	0.00%	85.10%	13.10%	50.30%	43.50%
E_C	342KB	99.60%	0.40%	96.50%	3.10%	49.60%	43.00%
A_I	357KB	99.80%	0.20%	97.30%	2.40%	22.60%	70.90%
CEDAW	354KB	99.20%	0.80%	95.30%	4.30%	47.40%	47.70%
A_H	352KB	100.00%	0.00%	98.00%	1.60%	34.50%	59.00%
A_E	366KB	99.90%	0.00%	99.40%	0.30%	45.60%	48.30%
S_D	310KB	98.20%	1.80%	93.60%	6.00%	23.60%	68.80%
A_F	312KB	99.80%	0.20%	97.20%	2.80%	34.90%	57.20%
E_D	346KB	99.80%	0.20%	78.80%	19.60%	61.70%	32.90%
A_G	387KB	99.80%	0.00%	94.20%	5.10%	46.70%	46.40%
TD	358KB	99.80%	0.20%	94.50%	4.50%	43.10%	50.70%
S_C	317KB	99.60%	0.40%	95.90%	2.50%	32.90%	60.60%
S_B	342KB	99.40%	0.40%	94.10%	5.50%	43.00%	50.90%
FCCC	294KB	99.00%	0.80%	93.70%	5.30%	47.60%	44.00%
A_J	265KB	99.70%	0.30%	93.40%	5.80%	38.90%	50.20%
UNEP	313KB	99.20%	0.80%	91.40%	8.20%	19.20%	75.40%
S_A	341KB	100.00%	0.00%	93.70%	5.50%	46.90%	44.20%
DP	319KB	99.40%	0.60%	95.00%	4.50%	34.20%	55.90%
JOURNAL	303KB	99.80%	0.20%	92.30%	6.10%	53.40%	41.60%
...	...	...	...	...	...		
Total	23.2MB	99.50%	0.44%	93.29%	5.97%	41.75%	50.82%

Table 5.7: Comparison of accuracies of the CR metric at identifying satisfactory and unsatisfactory sentence translations using different threshold values for the UN1 testing corpus.

Threshold Value	Translation Accuracy for Satisfactory Sentences		Translation Accuracy for Unsatisfactory Sentences		Overall Accuracy	
	SLR	**CR**	**SLR**	**CR**	**SLR**	**CR**
1.25	73.87%	88.03%	83.12%	100.00%	78.50%	94.02%
1.50	95.58%	99.39%	65.39%	70.30%	80.49%	84.85%
1.75	99.00%	99.97%	48.38%	46.12%	73.69%	73.05%
2.00	99.68%	100.00%	32.09%	27.28%	65.89%	63.64%
2.25	99.84%	100.00%	21.39%	16.88%	60.62%	58.44%
2.50	99.94%	100.00%	14.54%	9.49%	57.24%	54.75%
2.75	99.97%	100.00%	10.64%	5.43%	55.31%	52.72%
3.00	99.98%	100.00%	7.96%	2.90%	53.97%	51.45%
3.25	99.99%	100.00%	6.52%	1.87%	53.26%	50.94%
3.50	100.00%	100.00%	5.43%	1.17%	52.72%	50.59%
3.75	100.00%	100.00%	4.77%	0.72%	52.39%	50.36%
4.00	100.00%	100.00%	4.20%	0.42%	52.10%	50.21%

otherwise it would be deemed to be satisfactory. This was then compared with ground truth judgments that had been made by a human translator to determine the accuracy.

For the 10,000 satisfactory translations, we can see that all CRs were less than 2.00 whereas sentence length values reached up to 3.50. For the 10,000 unsatisfactory translations, most SLR and CR values should be much higher than the satisfactory translations and we can see this reflected in the table – all unsatisfactory translations' CR values are higher than 1.25. When we combine the unsatisfactory and satisfactory results to derive the overall accuracies on the testing corpus, we can see that the best threshold value is 1.50 for SLR and 1.25 for CR. The CR results reflects the relative quality of the unsatisfactory sentence pairs that were chosen for the UN1 testing corpus.

Figure 5.1 graphs the frequency of SLR and CR values obtained for both the satisfactory and unsatisfactory sentence pairs from the test corpus. The figure shows that for satisfactory sentence pairs, the frequency of low SLR and CR values is highest around the 1.0 value (over 3500 for SLR and over 5000 for CR), and this number drops off very rapidly as the threshold increases. The trend is very different for the unsatisfactory sentence pairs. The frequency of SLR values for these sentences is relatively constant at under 1000, but the picture for the CR values is very different. The frequency rapidly increases as the value rises, with the greatest frequency occurring for CR = 2.1. The steep drop off beyond this points reflects that the unsatisfactory sentence pairs that were chosen were mostly well-formed - i.e. if there was half a sentence missing in one sentence of the pair compared to the other (as a result of a 2:1 or 1:2 sentence mis-alignment, for example), this would normally result in a CR value much higher than 2. However, as discussed above, these types of unsatisfactory sentences were deliberately not chosen for the test corpus and sentences where the mistranslation was not obvious (except to a translator) were chosen instead.

Figure 5.2 graphs the accumulated frequencies of sentence pairs that have SLR and CR values greater than or equal to the threshold value shown on the x-axis. So, for example, all sentence pairs should have an SLR or CR value of 1.0 or above, so the frequency on the y axis reflects the total number of sentence pairs in the test corpus (10,000 in this case). The accumulated frequencies drop off rapidly for both SLR and CR on the satisfactory sentence pairs so that around the threshold value of 1.4 or 1.5 there are very few sentence pairs that have an SLR or CR value above these thresholds. In contrast for the unsatisfactory sentences, the accumulated frequency shows a steady (almost linear) trend downwards showing that there is more variability

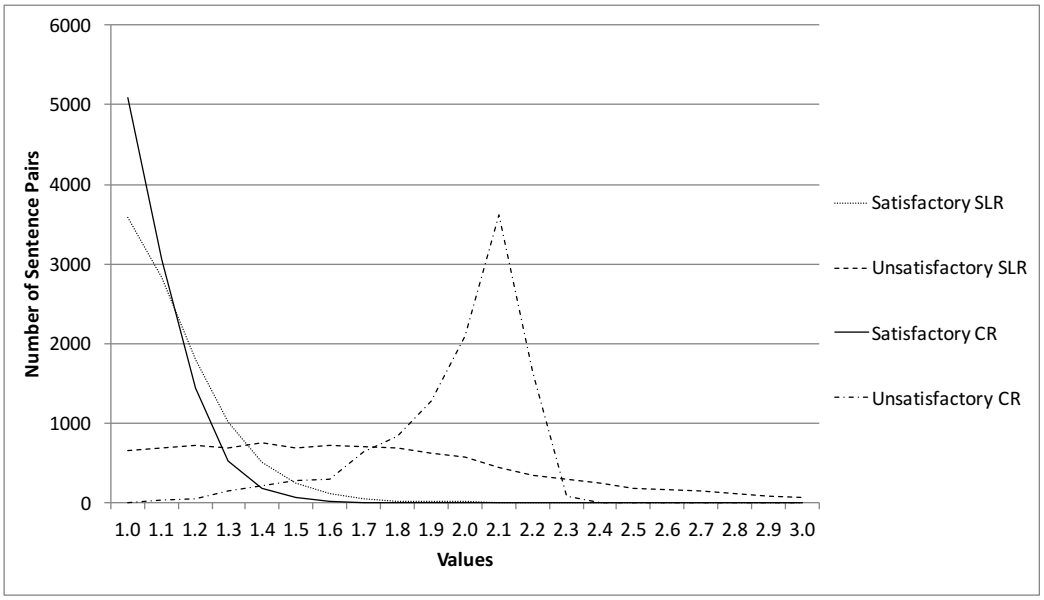


Figure 5.1: The number of satisfactory and unsatisfactory sentence pairs of the UN corpus by SLR and CR in different threshold values.

in the SLR values that are generated for these sentences. The accumulated frequency trend for the CR values, however, indicates that relatively few of the unsatisfactory sentence pairs have a CR below 2.0. This indicates that the CR metric has accurately identified most of the unsatisfactory sentence pairs.

Finally, for evaluating the quality of the output of the compression code length based alignment categorisation, a small sample of 100 pairs of sentences were randomly selected, which were deemed to be satisfactory and unsatisfactory from the UN Corpus using a compression code length ratio of 1.5. After manually checking the output, we have calculated the True Positive Rate (TPR) as 100% and the True Negative Rate (TNR) as 82%. More details are shown in Table 5.8.

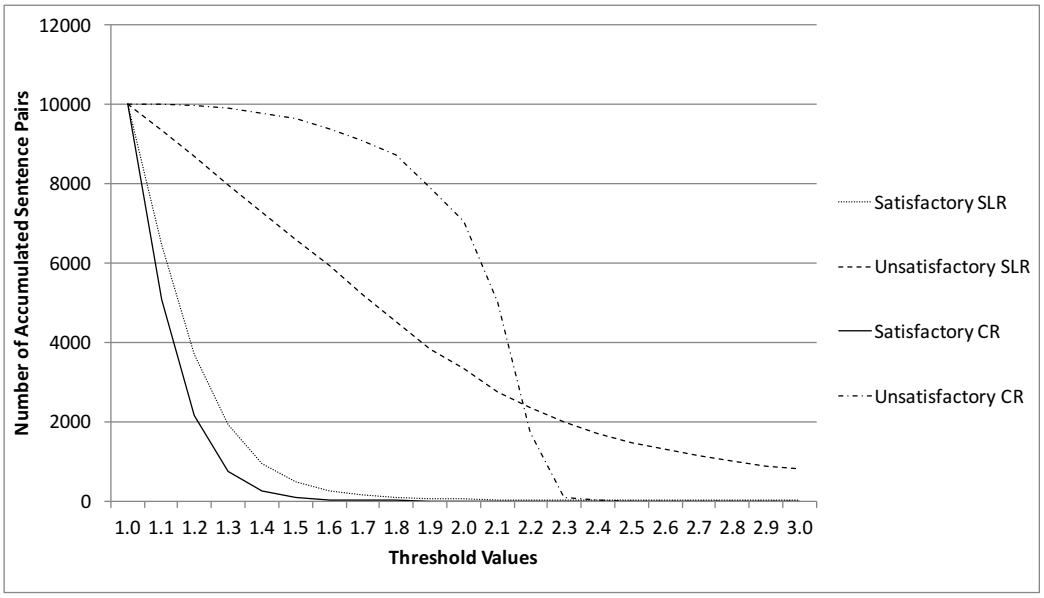


Figure 5.2: The accumulated number of satisfactory and unsatisfactory sentence pairs of the UN corpus by SLR and CR values.

Table 5.8: True Positive Rate and False Positive Rate for 100 satisfactory sentences and 100 unsatisfactory sentences.

Positive	**Negative**	**TP**	**TN**	**FP**	**FN**	**TPR**	**FPR**
59	41	50	41	9	0	100%	18%

5.7 Experiment 2: Automatic Creation of New Parallel Corpora

This experiment presents how well the sentence length ratio (SLR), sentence length difference (SLD), code length ratio (CR) and code length difference (CD) performed at aligning parallel text which is automatically obtained from the Internet. Section 3.3.2 has talked about distance measures for SLR, SLD, CR and CD, which are calculated by Equation 3.1, Equation 3.2, Equation 3.3 and Equation 3.4. Nowadays, bilingual text resource from the Inter-

net becomes more ample and substantial (Trieu et al., 2015). The training and testing corpora that this experiment used were both downloaded from the Hong Kong Yearbook website (`http://www.yearbook.gov.hk`). The website issues the Hong Kong Yearbook annually in English, Simplified Chinese and Traditional Chinese and each issue includes 21 categories as shown in Table 5.9. The training corpus was downloaded from the website manually whereas the testing corpus was automatically downloaded using a program. The testing corpus is a combination of Category 1 (Constitution and Administration), Category 6 (Employment), Category 8 (Health) and Category 15 (Public Order) from Hong Kong Yearbook Corpus 2014 and has been manually aligned for the use of accuracy calculation. The details of training and testing corpora are shown in Table 5.10.

Section 3.5.2 (page 63) has introduced the basic methodology for sentence alignment. This experiment used depth-limited algorithm for the 5-tree search for aligning the noisy testing corpus that was automatically downloaded and generated from the Internet. However, compared to the alignment results in Chapter 3, accuracies of sentence alignment for this testing corpus were expectedly lower because there is no guarantee that every translation is satisfactory or manually checked for the testing corpus. The code length values of CR and CD are calculated using PPMD with maximum order 5 for English and 6 for Chinese.

Table 5.11 presents the accuracies for the four metrics (SLR, SLD, CR and CD) at different depths. The total number of alignments for the testing corpus is 1,076, which have all been manually checked for correct alignment. Clearly, the best accuracies of CR and CD are both in depth 10. Figure 5.3 indicates the tendencies for the four measurements along with the depth values from 1 to 10. Compared with similar results in Figure 3.4 on page 69,

Table 5.9: Categories of Hong Kong Yearbook Corpus.

No.	Category
1.	Constitution and Administration
2.	The Legal System
3.	The Economy
4.	Financial and Monetary Affairs
5.	Commerce and Industry
6.	Employment
7.	Education
8.	Health
9.	Food Safety, Environmental Hygiene, Agriculture and Fisheries
10.	Social Welfare
11.	Housing
12.	Planning, Land and Infrastructure
13.	Transport
14.	The Environment
15.	Public Order
16.	The Media, Communications and Information Technology
17.	Population and Immigration
18.	Travel and Tourism
19.	Recreation, Sport, Culture and the Arts
20.	Religion and Custom
21.	History

Table 5.10: Hong Kong Yearbook Corpus for training and testing.

Use	Year (20××)	Align. Method	Size (Bytes)		Sentences	
			English	Chinese	English	Chinese
Training	07–13	Man.	7,983,865	6,327,021	34,837	35,273
Testing	14	Auto.	212,657	167,811	1,171	1,157

Table 5.11: Sentence alignment accuracies in different depths for Sentence Length Ratio (SLR), Sentence Length Difference (SLD), Code Length Ratio (CR) and Code Length Difference (CD).

Depth	**CR**	**CD**	**SLR**	**SLD**
1	81.41%	87.27%	72.68%	78.07%
2	81.41%	87.27%	72.68%	78.07%
3	82.25%	88.38%	72.12%	78.44%
4	82.81%	88.85%	72.40%	78.72%
5	82.34%	89.50%	72.49%	78.62%
6	81.88%	89.41%	73.23%	77.79%
7	81.78%	88.94%	**74.26%**	79.09%
8	82.06%	89.41%	73.88%	80.39%
9	82.62%	89.03%	73.33%	79.91%
10	**82.90%**	**89.59%**	72.96%	**80.76%**

the overall accuracies for the automatically generated testing corpus are lower than the testing corpus that was manually checked. In addition, SLD performed better than CR in Chapter 3 but for this testing corpus as Figure 5.3 shows, SLD is always below CR. In addition, Table 5.11 also indicates that there is no growth trend for SLR with increasing search depth.

In summary, as the test corpus was automatically downloaded from the Internet and there was no manual revision, more noise was expected and this is reflected in the results. However, the code length metrics (CR and CD) outperform the sentence length metrics (SLR and SLD). The accuracies of CR and CD still have room for improvement by improving the search algorithm. A hybrid approach which combines the CL and SL based metrics found effective for Arabic-English (Alkahtani et al., 2014) also should be investigated.

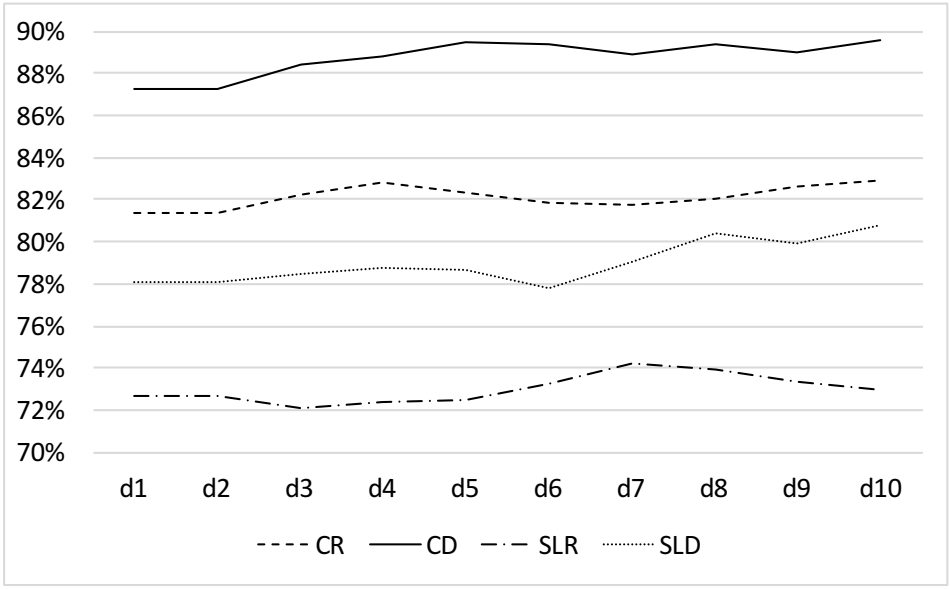


Figure 5.3: Sentence alignment accuracies in different depths for Sentence Length Ratio (SLR), Sentence Length Difference (SLD), Code Length Ratio (CR) and Code Length Difference (CD).

5.8 Conclusion

Evaluation experiments with the UN corpus which is a large parallel corpus of Chinese and English show that the PPM code length method is very effective at determining the quality of sub-sections of the corpus as the method can easily be applied on whole documents or partitions of the corpus as well as on separate sentences. The experimental results on a sample of sentences in the corpus indicate that there are a significant number of erroneous and poor translations in the corpus. These should be removed before using the corpus for training purposes for natural language processing or a statistical machine translation system. The experiments also indicate that the PPM code length metric is an effective method for filtering out unsatisfactory translations for this purpose.

From the experiments it can be seen that a new parallel corpus can be automatically created based on sentence length and PPM code length methods, where PPM code length method performs better. It is also possible to control the quality of a new parallel corpus creation by removing sentence pairs that exceed code length based thresholds.

Chapter 6

Back Translation & Translation System Evaluation

6.1 Introduction

In this chapter, we will evaluate translation systems by using back translations to compare translation results from different translation systems.

This chapter will focus on Chinese back-translation-based translation system evaluation. We will compare three translation systems—Google Translate, Baidu Translate and Youdao Translation. Google Translate is being used world-wide whereas Baidu and Youdao Translations are two popular choices in China. Secondly, we will talk about the PPM-based method that will be used for translation system evaluation. Next, an experiment to evaluate the three translation systems is conducted and the PPM-based experimental results are presented. Finally, according to the results we have obtained, we conclude which translation system is better and justify the evaluation method we have used.

6.2 Chinese-English Translation Systems

This section introduces three popular Chinese-English translation systems. The first was developed by an English-based company—Google (`http://www.google.com`) and two by Chinese-based companies—Baidu (`http://www.baidu.com`) and NetEase (`http://www.youdao.com`).

6.2.1 Google Translate

Google, Inc. is one of the biggest search engines in the world so far. Google started providing translation before before 2006 and offered a web interface in its initial stages with a few western languages (German, Spanish, French, Portuguese, Italian, *etc.*). It started with Chinese-English translations from 2006 and has been developing mobile interfaces for Android and iOS. Google announced an Application Program Interface (API) for its translation services in 2011, which is available for developers to build applications and other softwares.

Google Translate API is no longer free since the new version was released. The usage fees of Google Translate API for both translation and language detection are 20 USD (approx. 13 GBP) per one million characters (Google, Inc., 2014). In this thesis, we will be using the API to link our program and Google Translate service.

6.2.2 Baidu Translate

Baidu Translate was released in July 2011 by Baidu, Inc.—the biggest search engine in China. Baidu Translate is a collaborative project of Baidu, Inc. and Chinese Academy of Sciences (CAS) and also one of the fastest growing translation systems in China which is getting increasingly popular.

The web interface of Baidu Translate is free and supports Simplified and Traditional Chinese, Cantonese, Classical Chinese, English, Japanese, Korean, *etc.* Baidu Translate started supporting mobile interfaces for Android and iOS in February 2013. In recent years, a Baidu Translate API has become available and free to the public. However, an API key is needed, which can be issued from Baidu developer centre and is free for low-frequency-users (lower than 1000 time requests per hour and 2 million characters). For more than 2 million characters, Baidu will charge 70 CNY (approx. 7 GBP) for each million characters (Baidu, Inc., 2014).

6.2.3 Youdao Translation

Youdao Translation is another one of the most popular translation systems in China and web and mobile interfaces were published by NetEase, Inc. in 2007 and 2009. Up to April 2015, NetEase Youdao (web and mobile interfaces) has had 500 million users and 70% of the market share (NetEase, Inc., 2014).

The Youdao Translation API is also free to use for the public. An API key is needed, but is also free upon application. However, there is a limit of sending no more than 1000 requests per hour.

6.3 Back Translation

Back translation is the process of translating a language that has already been translated into a foreign language back to the original language (Paegelow, 2008). Back translation can be used by the same translation or a different one. Back translation is an important method for translation quality evaluation and applied in many technical areas (Ozolins, 2009). People have found

that the method of manual back translation is much more effective than direct translation in terms of translating precision (Weidmer, 1994). Therefore, when applied by translation systems, back translation method can be a measurement for checking the quality of translation systems.

Entropy can presents the information that a language text carries, which has been introduced in Section 3.3.1. Theoretically, a perfect translation and the source text should have the same entropy, whereas an imperfect translation has different entropy with the source text and the smaller the entropy values close the better the translation is. For a sentence in source language, different translation systems can translate the same sentence to different ones in target language. The sentences of source language and target language are likely similar and not easily to be manually evaluated that which translation performed better. However, if the translations that are translated back to the source language by respective translation systems and translated to the target language again, the difference between the source text and original translation will be magnified. If this kind of back translation is repeated two or more times, the difference between the source text and original translation will be exponentially increased. This methodology of back translation can be used for translation quality evaluation and also translation system comparison.

Table 6.1 shows an example of back translations. The original English and Chinese sentences are parallel and selected from the UN corpus. The "B1" sentence was translated by Google Translate from the original Chinese sentence to an English sentence, then translated back by Google Translate to Chinese using the English sentence. The "B2" sentence is the second back translation, which was translated based on "B1". "B3", "B4" and "B5" were consequently obtained using the same way. Clearly, Chinese speakers

Table 6.1: An example for the five back translations by Google Translate with the original English and Chinese sentences.

Original English	The exercise revealed the high potential and political significance of cooperation in the Balkan region and adjoining countries.
Original Chinese	这次演习揭示了在巴尔干地区和邻国开展合作的巨大潜力和政治意义。
B1: 1st Back Translation	这次演习揭示了在巴尔干地区和周边国家合作的巨大潜力和政治意义。 (*Perfect back translation*)
B2: 2nd Back Translation	这项工作显示出的合作与周边国家在巴尔干地区的巨大潜力和政治意义。 (*Three noun phrases and missing predicate*)
B3: 3rd Back Translation	巨大的潜力和政治表现出的合作与周边国家在巴尔干地区工作的重要意义。 (*Two noun phrases and missing predicate*)
B4: 4th Back Translation	的巨大潜力和政治合作的重要性，表示通过与周边国家在巴尔干地区工作。 (*Syntax error*)
B5: 5th Back Translation	巨大的潜力和政治合作的重要性，通过与邻国在巴尔干的工作表示。 (*Syntax error*)

can easily recognise that "B5" is much worse than the original sentence and not written by humans.

6.4 PPM-based Evaluation Method

In previous chapters, we have investigated a PPM-based method for evaluating translation alignment in parallel corpora. Co-translations in different languages should have the same information and therefore have similar com-

Table 6.2: Corpora using for translation quality evaluation.

Name	Use	Language	Sentences	Size (bytes)
LCMC	Training	Simplified Chinese	37,932	4,547,617
UN	Test	Simplified Chinese	1,000	97,056

pression code lengths. This can also be used for translation quality evaluation and translation performance comparison among different translation systems as follows.

As before, we will use PPMD as the PPM-based evaluation method with a maximum order of 6 for Chinese (2 Chinese characters order in UTF-8 encoding) and 5 for English to train corpora and compress test corpus and back translations. For our method, we will perform repeated back translations for each sentence from the test corpus five times and therefore obtain five back translations to compare. Code length ratios (CR) between the original Chinese sentences and their five back translations will be presented. In addition, as a comparison, we will then compare the performance of CR values with other translation evaluation methods, which will be introduced in the following section.

Table 6.2 shows the training and test corpora we are using in this experimental evaluation. All of them use UTF-8 encoding and have been aligned at the sentence level. The test corpus is manually collected from the UN corpus.

6.5 Bilingual Evaluation Understudy (BLEU)

Bilingual Evaluation Understudy (also called BLEU) is one of the algorithms for evaluation of translation quality which has been translated by machine from one natural language to another (Papineni et al., 2002). Scores of BLEU

are calculated by comparing original text (or a professional human translation) and machine translation to reach an estimate of translation's overall quality. The output of BLEU is always a number between 0 and 1, where 1 means a perfect translation and 0 indicates a poor one. A higher BLEU score indicates that the machine translation has more matched information to the original text (or a professional human translation).

One idea to compare MT output with satisfactory translations is to use statistics of short sequences of Chinese characters (character n-grams), which indicates that the greater number of n-grams that the translation being evaluated shares with the satisfactory translation, the better the translation is judged to be (Doddington, 2002). In this chapter, we will use the simplified basic BLEU method with another three enhanced variants based on the basic BLEU calculation—Modified BLEU, n-gram BLEU and Modified n-gram BLEU.

6.5.1 Basic BLEU

Basic BLEU uses the original BLEU calculation that calculates how many words (or Chinese characters) from a candidate sentence appear in a reference sentence and the formula is shown as follows:

$$P_B = \frac{m}{w_t}$$

where P_B is the Basic BLEU score, w_t is the total number of words (or Chinese characters) in the candidate sentence and m is the number of words (or Chinese characters) from the candidate sentence that appear in the reference sentence. The following is an example for the calculation of Basic BLEU:

Example:

```
Candidate:  The dog barked to the bird on the tree.
Reference:  The dog barked and the bird sang.
```

where we can see that the candidate is obviously not a correct translation from the reference sentence. According to Basic BLEU calculation, there are 7 words (a punctuation is considered as a word for counting purposes) found in the reference sentence and there are 8 words in total (including punctuations) in the reference sentence. Therefore, $P_B = 7/8 = 0.875$.

6.5.2 Modified BLEU

A modified BLEU calculation is used to avoid too many identical words (or Chinese characters) in candidate sentence found fewer times in reference sentence, which m is limited as m_{max}. The calculation is shown as follows:

$$P_{MB} = \frac{m_{max}}{w_t}$$

where P_{MB} is Modified BLEU score and m_{max} indicates a limited total number of words (or Chinese characters) from the candidate sentence that appear in the reference sentence. According to the modified formula, for the example in previous section, m_{max} is 6 because there are only two "the" words in the reference sentence but three "the" words in the candidate, the three "the" words in this case is limited to two. Therefore, $P_{MB} = 6/8 = 0.750$. Compared to the 0.875 of Basic BLEU calculation, 0.750 is perhaps a more reasonable BLEU score because of the incorrect translation.

6.5.3 n-gram BLEU

By n-grams, experiments with BLEU showed that n-grams of length 4 work best (Papineni et al., 2002). However, we choose $n = 2$ for computing BLEU scores for the same example and the formula is as follows:

$$P_{nB} = \frac{m}{w_t - n + 1}$$

where P_{nB} is the n-gram BLEU score and m is the number of all two sequential words (or Chinese characters) combinations from the candidate sentence that appear in the reference sentence. Therefore, $P_{nB} = 3/(8-2+1) = 0.429$, which is significantly lower than Basic BLEU and Modified BLEU values for this example.

6.5.4 Modified n-gram BLEU

Modified n-gram BLEU uses the same modification as Modified BLEU for n-gram BLEU and the calculation is shown as follows:

$$P_{MnB} = \frac{m_{max}}{w_t - n + 1}$$

where P_{MnB} is the Modified n-gram BLEU score, which for the same example is $P_{MnB} = 3/(8 - 2 + 1) = 0.429$.

6.6 Experiment

We use the three translation systems to translate the test corpus of 1,000 Chinese sentences and to use the 1,000 translations (English) to translate back to Chinese to get the 1,000 first back translations. Then we use the 1,000 first back translations to produce the second back translations and so on until we have produced the fifth back translations. These are marked as B1 to B5 in the following figures.

We have obtained 5,000 back translations (from B1 to B5) as well as the 1,000 original corpus. All the codelength values and BLEU scores are calculated based on the 5,000 + 1,000 sentences.

As a first experiment, we used each original sentence as a priming "training corpus" for PPMD compression method to compress its five back translations. The idea was that the best training data for this experiment is the

original sentence itself. Figure 6.1a, Figure 6.1b and Figure 6.1c show the amount of the five back translations which are greater than or equal to the values on the x-axis by the three translation systems.

We can see that many codelength values are higher than 2.0 and there is no obvious indication to show which translation system performed better. Figure 6.1d shows the average codelength values based on the previous three figures.

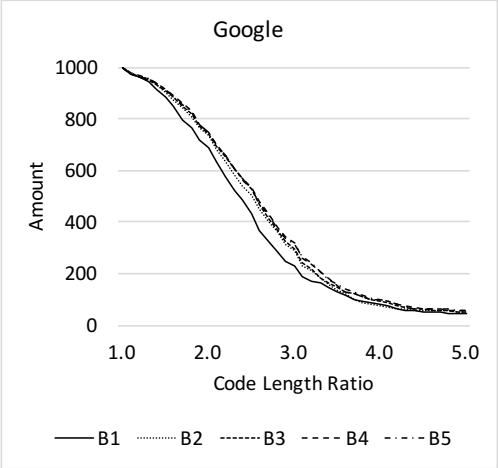
The average CR values of Google Translate in Figure 6.1d shows an increasing trend meaning that the quality of back translations were getting worse with each iteration. However, both Baidu and Youdao translation systems did not show an obviously increasing trend in Figure 6.1d.

As a second experiment, we used the LCMC corpus instead of priming the original sentence to do the same experiment. Figure 6.2a, Figure 6.2b and Figure 6.2c show the results of the experiment. We can see that the data becomes more reasonable and makes more sense. All codelength values are lower than 2.0 and most of them are between 1.0 and 1.1. Compared to Figure 6.2b, Figures 6.2a and 6.2c for Google and Youdao has more translations where the CR values are lower than 1.1.

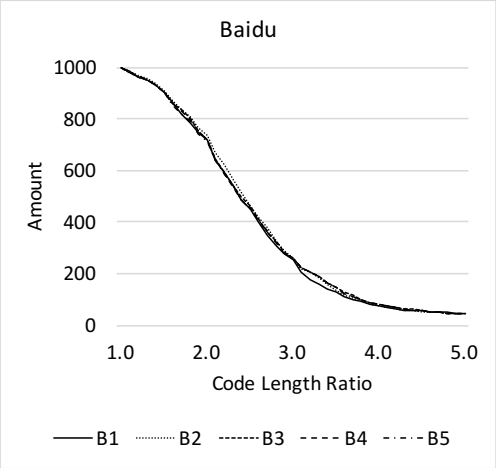
Figure 6.2d presents average CR values for the three translation systems, where we can clearly see that Google Translate has a better overall CR range than the others.

We also used BLEU-based calculation methods to do the similar comparison. Figures 6.3a, 6.3b and 6.3c present basic BLEU scores for the back translations. We can see that Google Translate still performs slightly better than others.

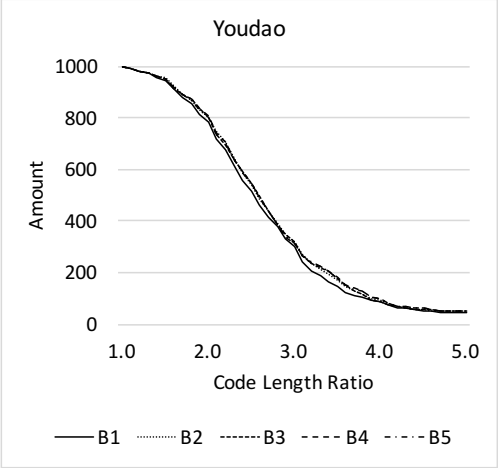
Figures 6.4a, 6.4b and 6.4c show the scores of Modified BLEU and we can see that overall BLEU scores are getting slightly worse.



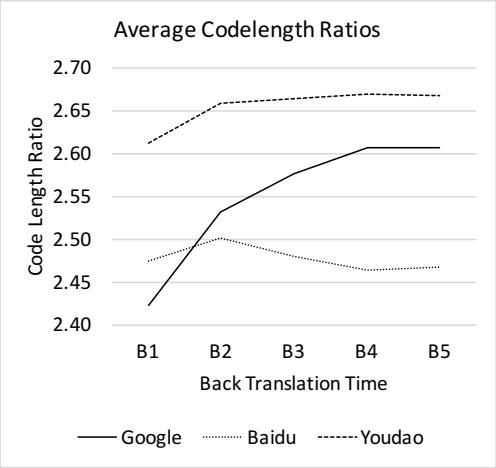
(a) Amount of CR values using Google Translate greater than or equal to x-axis label values.



(b) Amount of CR values using Baidu Translate greater than or equal to x-axis label values.

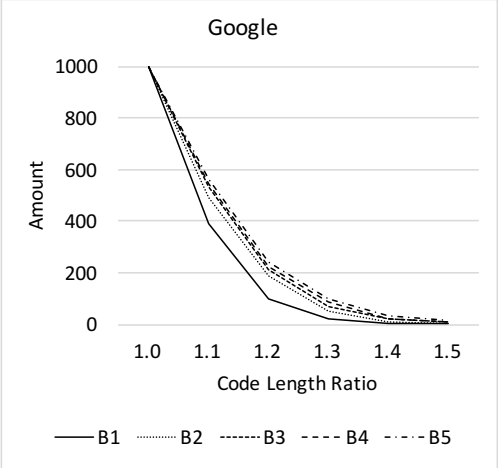


(c) Amount of CR values using Youdao Translation greater than or equal to x-axis label values.

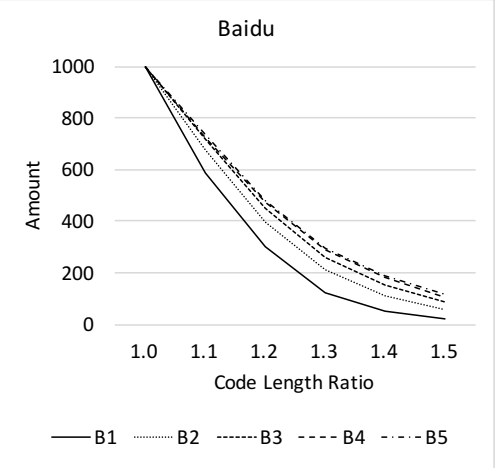


(d) Average CR values for Google Translate, Baidu Translate and Youdao Translation.

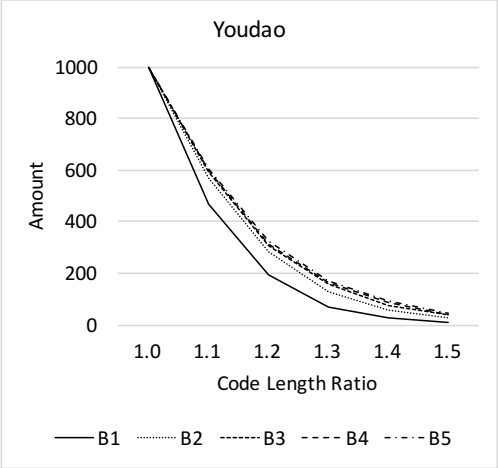
Figure 6.1: Amount of CR values using the original sentences respectively as the priming "training corpora" for Google, Baidu and Youdao translation systems.



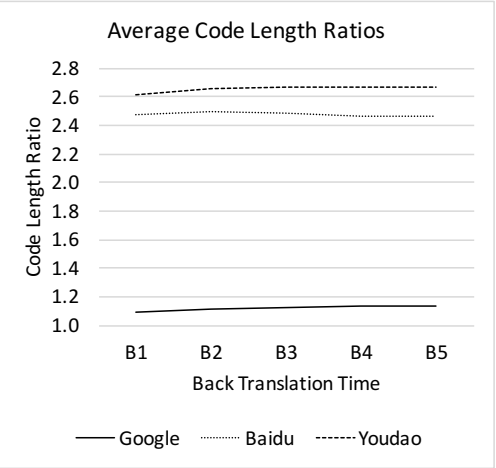
(a) Amount of CR values using Google Translate greater than or equal to x-axis label values.



(b) Amount of CR values using Baidu Translate greater than or equal to x-axis label values.

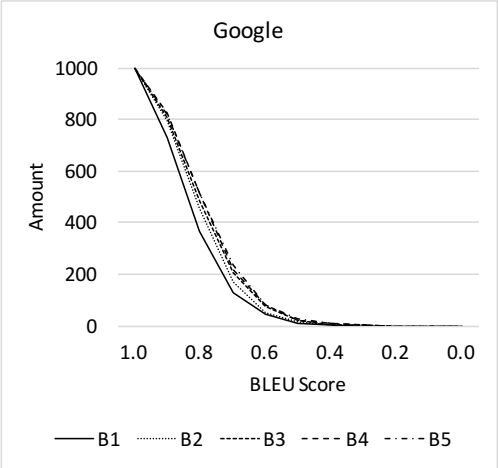


(c) Amount of CR values using Youdao Translation greater than or equal to x-axis label values.

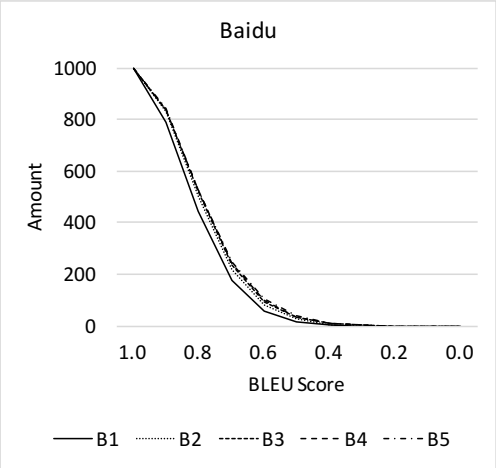


(d) Average CR values for Google Translate, Baidu Translate and Youdao Translation.

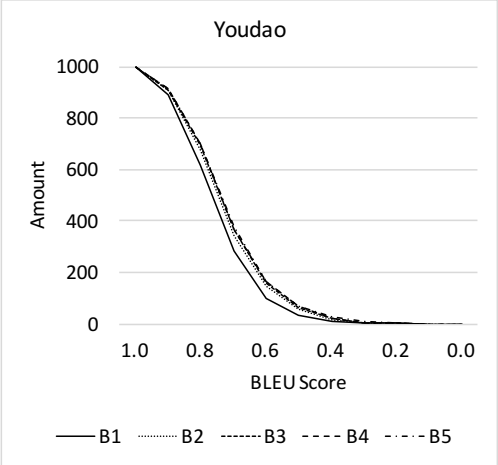
Figure 6.2: Amount of CR values by training the LCMC corpus for Google, Baidu and Youdao translation systems.



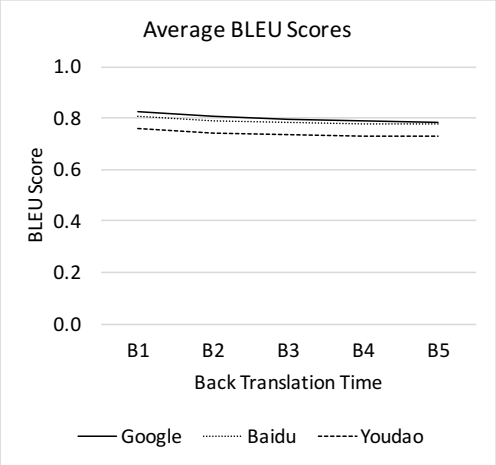
(a) Amount of basic BLEU scores using Google Translate greater than or equal to x-axis label values.



(b) Amount of basic BLEU scores using Baidu Translate greater than or equal to x-axis label values.

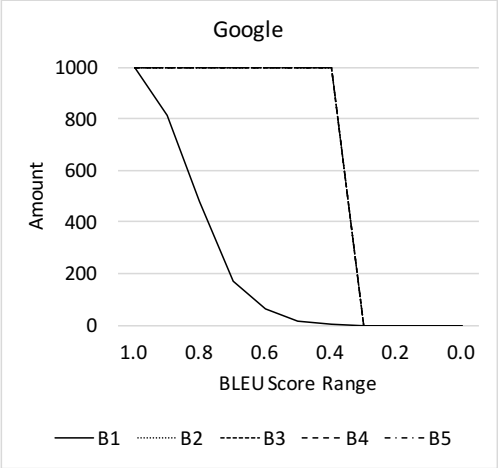


(c) Amount of basic BLEU scores using Youdao Translation greater than or equal to x-axis label values.

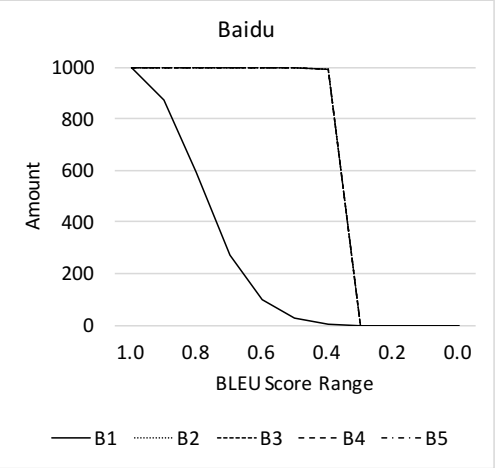


(d) Average basic BLEU scores for Google, Baidu and Youdao translation systems.

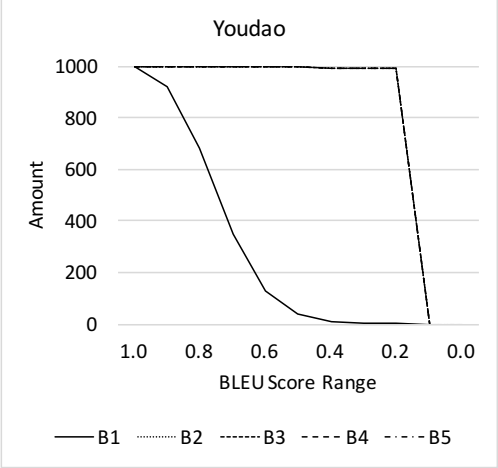
Figure 6.3: Amount of basic BLEU scores for Google, Baidu and Youdao translation systems.



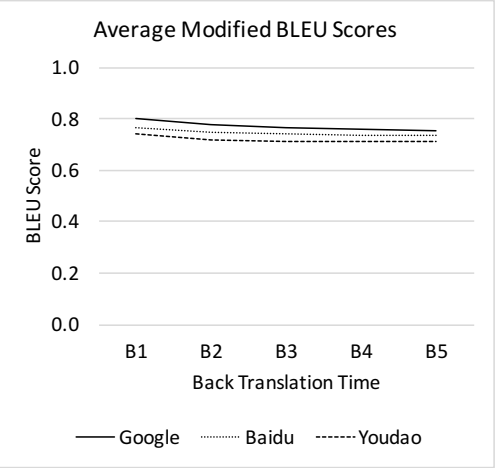
(a) Amount of modified BLEU scores using Google Translate greater than or equal to x-axis label values.



(b) Amount of modified BLEU scores using Baidu Translate greater than or equal to x-axis label values.



(c) Amount of modified BLEU scores using Youdao Translation greater than or equal to x-axis label values.



(d) Average modified BLEU scores for Google, Baidu and Youdao translation systems.

Figure 6.4: Amount of modified BLEU scores for Google, Baidu and Youdao translation systems.

When we used n-gram BLEU calculation method with $n = 4$ as Figures 6.5a, 6.5b and 6.5c showed, most translations presented lower BLEU scores.

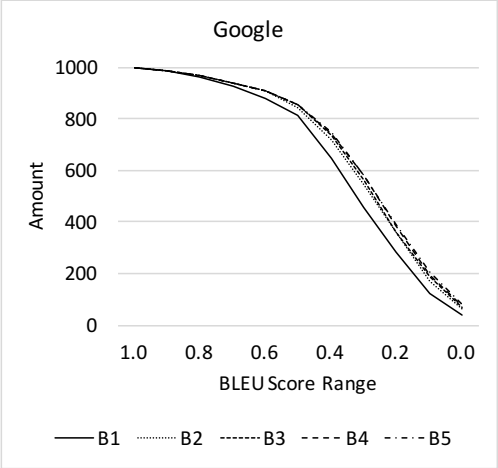
Figures 6.6a, 6.6b and 6.6c are for Modified n-gram BLEU and resulted in almost identical graphs to n-gram BLEU.

Figures 6.7a, 6.7b and 6.7c are statistical bar charts for the three translation systems. Each bar indicates an average BLEU score of the 1,000 same time back translations. The three charts present a comparison among the four BLEU-based metrics for evaluating back translation and translation quality. However, the basic and modified BLEU methods produced much higher scores than the n-gram and modified n-gram BLEU methods and the scores are either too high or to low for the same back translations.

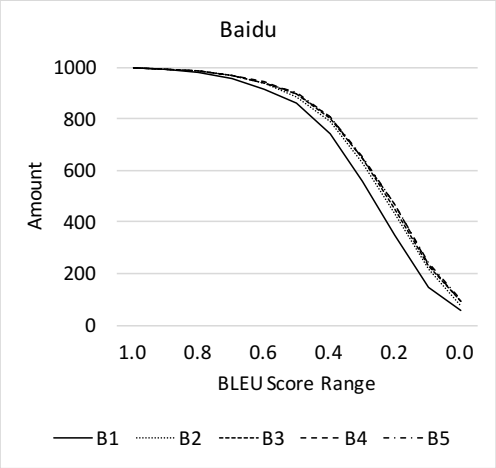
In summary, BLEU scores go up with successive back translations whereas code lengths go down because generally the translation quality improves with each back translation as the sentences start to converge. Compared to PPM-based method shown in Figures 6.1 and 6.2, the advantage of BLEU-based translation evaluation methods is mainly the speed due to there being no training phrase needed.

6.7 Conclusion

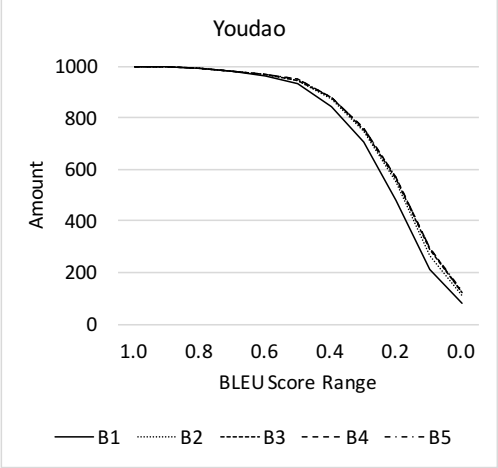
After analysing the previous bar charts, it is clear to see that Google Translate performs better than the others and Baidu Translate performs better than Youdao Translation. According to four different BLEU calculations, we have found that BLEU is not the best translation quality evaluation method. A high BLEU score does not always mean a good translation and vice versa. In contrast, PPMD using the LCMC training corpus provided consistent re-



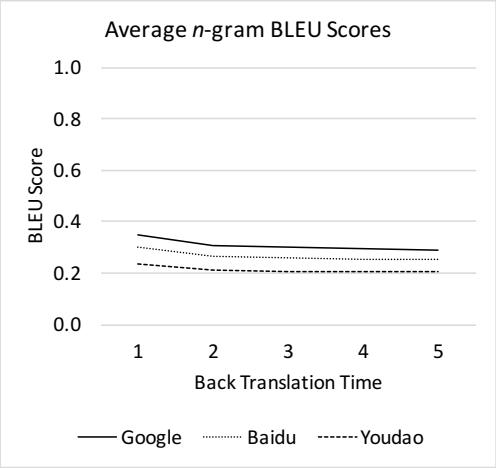
(a) Amount of n-gram BLEU scores using Google Translate greater than or equal to x-axis label values.



(b) Amount of n-gram BLEU scores using Baidu Translate greater than or equal to x-axis label values.

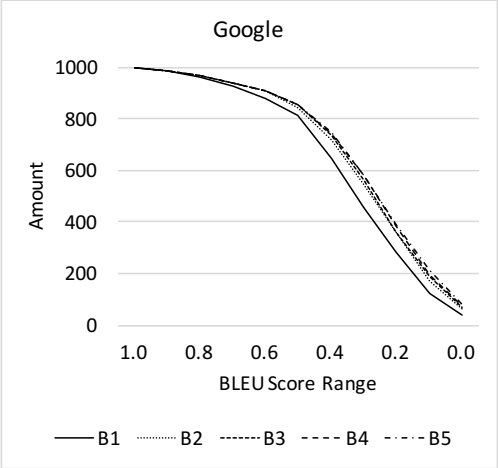


(c) Amount of n-gram BLEU scores using Youdao Translation greater than or equal to x-axis label values.

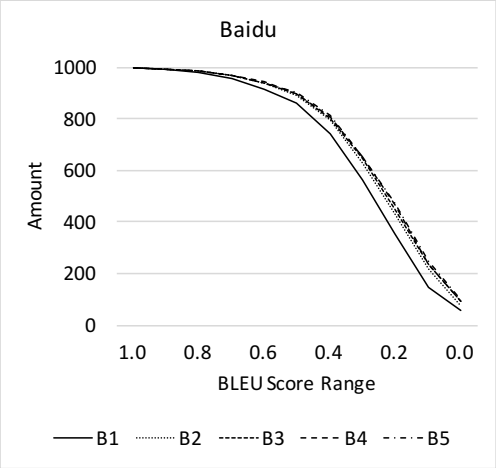


(d) Average n-gram BLEU scores for Google, Baidu and Youdao translation systems.

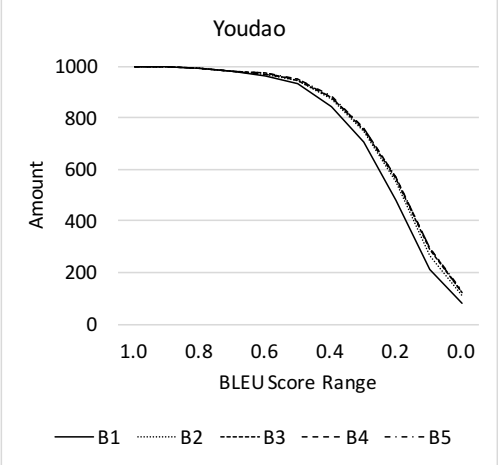
Figure 6.5: Amount of n-gram BLEU scores for Google, Baidu and Youdao translation systems.



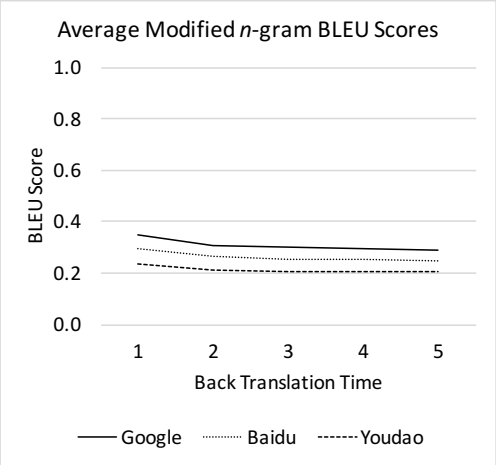
(a) Amount of modified n-gram BLEU scores using Google Translate greater than or equal to x-axis label values.



(b) Amount of modified n-gram BLEU scores using Baidu Translate greater than or equal to x-axis label values.

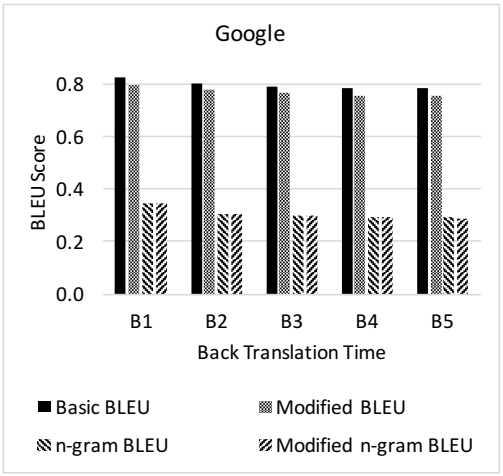


(c) Amount of modified n-gram BLEU scores using Youdao Translation greater than or equal to x-axis label values.

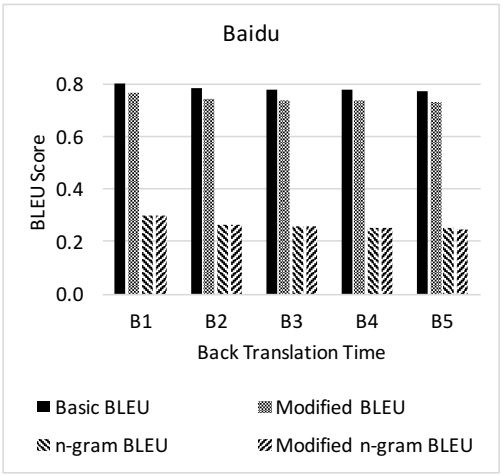


(d) Average modified n-gram BLEU scores for Google, Baidu and Youdao translation systems.

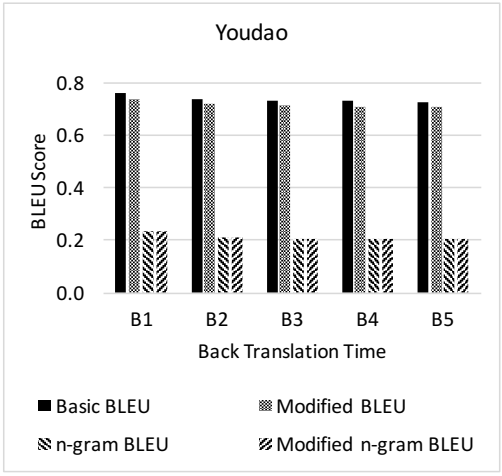
Figure 6.6: Amount of modified n-gram BLEU scores for Google, Baidu and Youdao translation systems.



(a) Precisions for different BLEU calculations by Google Translate.



(b) Precisions for different BLEU calculations by Baidu Translate.



(c) Precisions for different BLEU calculations by Youdao Translation.

Figure 6.7: Comparing BLEU scores for the four BLEU calculation among Google, Baidu and Youdao translation systems.

sults for this experiment and therefore there is reason believe that the PPM-based translation quality evaluation method will work for more samples. The experiment showed that the back translation-based evaluation method was able to present differences between original sentences and their back translations more accurately.

In the future work of evaluation of translation quality, we will extend the comparison to provide stronger evidence that the PPM-based method provides an excellent method for evaluating translation quality.

Chapter 7

Discussion & Conclusions

7.1 Introduction

This chapter generally discusses this research and all experiments, concludes the study and reviews the initial aims and objective. The most important results and conclusions are highlighted in this chapter. In the section of limitations, a number of reasons of disadvantages and some negative discussions are presented. At the end of this thesis, the future work along with personal and professional recommendations is discussed.

7.2 Summary & Conclusions

This research firstly reviewed SMT including the history, important models, basic process as well as the difficulties of SMT and the research direction. Secondly, as the basic approach of this study, Corpus Linguistic has been discussed. Next, we have reviewed a number of existing corpora and the their types. Some useful parallel corpus evaluation methods have also been subsequently discussed, which are essential for evaluating the quality of trans-

lations and parallel corpora. Parallel corpus alignment at different levels has also been discussed. We have then reviewed various PPM models, compared major PPM variants (PPMA, PPMB, PPMC and PPMD) and discussed later variants of PPM. Text encodings for English and Chinese encodings have been reviewed and UTF-8 is the chosen encoding for this study.

In chapter 3, we firstly compared three compression methods which are Gzip, Bzip2 and PPM. The experimental results showed that PPM performed better than the others for compressing natural language text. Then two new distance metrics have been introduced for matching sentences for alignment of parallel corpora. Two of the metrics are based on computing the compression code length of the sentences as this is an accurate measure of the information contained in the text. The best metric for determining sentence alignment was based on absolute compression code length difference between sentence pairs. Absolute difference based metrics (including when using sentence length) were also more effective than using ratio based metrics.

The first experiment of chapter 4 for the evaluation of the DC, HK and UN corpora show that using the PPM compression code length metric is an effective method to evaluate a parallel corpus or compare the quality of two or more parallel corpora. The experimental results were compared in different ways to describe the different features among the three testing corpora. The DC corpus is not with natural sequence but has the fewest unsatisfactory translations or mistranslations, and therefore the DC corpus should be least noisy. The HK corpus is the smallest corpus with natural sequences but is most noisy because there are a high percentage of unsatisfactory translations or mistranslations. The UN corpus is the largest corpus with natural sequence. There are overall the most satisfactory translations in UN Corpus

and a reasonable small percentage of noise. The DC corpus also presented a better quality than the other two corpora with the HK corpus having the lowest quality. The results concluded from the first experiment matched the manual reviews of the three testing corpora. The second experiment used the KDE4 and GNOME corpora which were automatically collected from the Internet. The experiment at results clearly showed that the quality of the KDE4 and GNOME corpora is worse than the DC, HK and UN corpora and they are not satisfactory to use unless any further cleaning is done on them. From the second experiment, we have also seen that PPM-based compression method is effective for recognising unsatisfactory Chinese-English parallel corpora and even unsatisfactory part from the corpora.

Chapter 6 evaluated the UN corpus showed that the PPM code length method is very effective at determining the quality of sub-sections of the corpus. The experimental results indicate that a significant number of erroneous and poor translations in the corpus have been recognised by the PPM code length method. The experiments also indicate that the PPM code length metric is an effective method for filtering out unsatisfactory translations for this purpose. From the experiments it can be seen that a new parallel corpus can be automatically created based on sentence length and PPM code length methods. It is also possible to control the quality of a new parallel corpus creation by removing sentence pairs that exceed code length based thresholds.

Finally, the study compared Google, Baidu and Youdao translation systems in chapter 6 and employed four BLEU-based metrics for evaluating back translations. Experimental results showed that Google Translate performed better than the others and Baidu Translate performed better than Youdao Translation. Comparing the four BLEU-based metrics with PPM code length

method, we have found that BLEU is not the best translation quality evaluation method because a high BLEU score does not always mean a good translation and vice versa. In the experiment, PPM provided consistent results and there is reason believe that the PPM-based translation quality evaluation method will work for more samples. The experiment also showed that the back translation-based evaluation method was able to present differences between original sentences and their back translations more accurately.

7.3 Review of Aim & Objectives

The aim and objectives of this study that have been proposed in Section 1.2 have all been successfully achieved. The novel PPM compression-based method has been applied to the tasks of alignment, automatically creating and evaluating Chinese-English parallel corpora and the results are competitive. PPM as a compression-based method for sentence alignment has been compared and contrasted with Gzip and Bzip2. Therefore, as the chosen compression scheme, PPM code length metrics (CR and CD) have been used for aligning Chinese-English bilingual parallel corpora. Compared with sentence length-based metrics (SLR and SLD) for sentence alignment, PPM achieved higher accuracies than sentence length methods. For evaluating Chinese-English parallel corpora, the results also showed that PPM code length was effective to be a metric for Chinese-English parallel corpus evaluation. After the determination of the best threshold code length ratio for recognising whether the translation is satisfactory or unsatisfactory, the threshold value of 1.5 has been employed and justified the feasibility for the automatic creation of Chinese-English parallel corpus from the Internet. Finally, the PPM-based code length metric has successfully been applied for measuring

translation quality and comparing common translation systems.

Therefore, the specific objectives as detailed in section 1.2 were achieved as follows:

- *Compare and contrast whether PPM performs better than other common compression methods for compressing Chinese and English text.*
  PPM has been compared and contrasted with Gzip and Bzip2 compression schemes and the experimental results showed that PPM performed better the others for compressing Chinese-English bilingual text. The objective was achieved in section 3.4.

- *Determine how well the PPM-based evaluation method works for aligning Chinese-English parallel corpora at the sentence level.*
  The determination of whether PPM-based evaluation method works well for aligning Chinese-English parallel corpora at the sentence level has been achieved in chapter 3.

- *Examine whether PPM code length-based metrics perform better than traditional sentence length-based metrics.*
  The objective of whether PPM code length-based metrics perform better than traditional sentence length-based metrics was achieved in chapter 3, chapter 4 and chapter 5.

- *Evaluate the quality of Chinese-English parallel corpora by using the novel PPM compression code length metric.*
  The quality of Chinese-English parallel corpora by using the novel PPM compression code length metric has been evaluated in chapter 4.

- *Evaluate whether PPM-based compression method works well for automatic creating Chinese-English parallel corpora from the Internet.*

The evaluation of whether PPM-based compression method works well for automatic creating Chinese-English parallel corpora from the Internet was achieved in chapter 5.

- *Investigate the PPM-based evaluation method as a way for measuring and comparing common translation systems and determine whether PPM-based evaluation method works better than BLEU evaluation measurements.*

  Finally, the investigation of the PPM-based evaluation method as a way for measuring and comparing common translation systems and determination of whether PPM-based evaluation method works better than BLEU evaluation measurements were both achieved in chapter 6.

7.4 Limitations

There are some limitations for this research due to various reasons. First of all, as expected, the speed of PPM compression is slower especially when compressing for code length compare to the speed of calculating sentence length. The slower speed for each compression leads to a long time spent calculating code length values for each sentence of a large corpus. As a result, the Python implementation needs to be substantially optimised.

Another main limitation is that depth-limited search algorithm is time consuming. This makes sentence alignment difficult almost impossible when depth is deeper than 10 on a normal computer, which means for a 5-tree, the algorithm has to perform 48,828,125 (5^{11}) calculations for each alignment. In addition, the alphabet size of a language is also a factor of performance. For large alphabet languages (e.g. Chinese), a larger size corpus is more important than small alphabet languages (e.g. English).

7.5 Future Work

Based on this research, there are a number of questions that have arisen and merit further investigation as follows:

- An API for PPM compression that supports other programming languages needs to be implemented, so that the speed performance of calculating code length values can be significantly improved.
- The search algorithm for automatic alignment needs to use dynamic programming methods to find the optimal alignment and also to improve the speed performance.
- The alignment search method should also support M:N models (M$\geq$2, N$\geq$2).
- The use of PPM-based methods to align parallel corpora down to phrase and word levels should also be investigated.
- A statistical machine translation system based on PPM models should also be explored..
- A hybrid approach which combines the code length and sentence length based metrics found effective for Arabic-English (Alkahtani et al., 2014) also should be investigated.

Bibliography

Alkahtani, S. (2015). Building and verifying parallel corpora between Arabic and English. *PhD Dissertation, Bangor University*.

Alkahtani, S., Liu, W., and Teahan, W. J. (2014). Aligning a New Parallel Corpus of Arabic-English. *17th International Conference on Text, Speech and Dialogue*.

Baidu, Inc. (2014). Baidu Translate API Pricing. `http://fanyi.baidu.com`. Accessed: 01-07-2014.

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with Bilingual Parallel Corpora. *Proceedings of the 43rd Annual Meeting of the ACL*, pages 597–604.

Bauer, L. (1993). *Manual of information to accompany the Wellington corpus of written New Zealand English*. Department of Linguistics, Victoria University of Wellington.

Behr, F. H., Fossum, V., Mitzenmacher, M., and Xiao, D. (2003). Estimating and Comparing Entropy across Written Natural Languages Using PPM Compression. *Proceedings of Data Compression Conference*, page 416.

Benoit, G. (2013). Character Encoding. *Simmons College*, pages 1–27.

Bertsekas, D. P. (2011). Dynamic Programming and Optimal Control. *Massachusetts Institute of Technology, 3rd Edition*, 2.

Braune, F. and Fraser, A. (2010). Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora. *Coling 2010: Poster Volume*, pages 81–89.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A Statistical Approach to Machine Translation. *Comput. Linguist.*, 16(2):79–85.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning Sentences in Parallel Corpora. *ACL '91 Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Comput. Linguist.*, 19(2):263–311.

Brunning, J. (2010). Alignment Models and Algorithms for Statistical Machine Translation. *PhD Dissertation, Cambridge University Engineering Department and Jesus College.*

Bzip2 (2012). The Bzip2 Home Page. `http://www.bzip.org`. Accessed: 01-08-2012.

Callison-burch, C. and Osborne, M. (2006). Re-evaluating the role of BLEU in machine translation research. In *In EACL*, pages 249–256.

Chang, Z. (2008). A PPM-based Evaluation Method for Chinese-English Parallel Corpora in Machine Translation. *MSc Dissertation, University of Wales: Bangor.*

Chiang, D. (2005). A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

Church, K. W. (1993). Char_align: a program for aligning parallel texts at the character level. *Proceedings of the 31st annual meeting on Association for Computational Linguistics (ACL)*, pages 1–8.

Church, K. W. and Hovy, E. H. (1993). Good applications for crummy machine translation. *Machine Translation*, 8(4):239–258.

Cleary, J. G. and Witten, I. H. (1984). Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Transactions on Communications*, 32(4):396–402.

Coded Character Set (1986). 7-Bit American Standard Code for Information Interchange. *ANSI X3*, 4.

Collins, P. and Peters, P. (1988). The Australian corpus project. *Kytö et al*, pages 103–121.

Crego, J. M. and Mariño, J. B. (2006). Improving Statistical MT by Coupling Reordering and Decoding. *Machine Translation*, 20(3):199–215.

Creutz, M. and Lagus, K. (2007). Unsupervised Models for Morpheme Segmentation and Morphology Learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34.

Davies, M. (2010). The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and linguistic computing*, page fqq018.

Ding, H., Quan, L., and Qi, H. (2011). The Chinese-English Bilingual Sentence Alignment Based on Length. *Int. Conference on Asian Language Processing*, pages 201–204.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dukes, K. and Habash, N. (2010). Morphological Annotation of Quranic Arabic. In *LREC*. Citeseer.

Dyer, C., Clark, J., Lavie, A., and Smith, N. A. (2011). Unsupervised Word Alignment with Arbitrary Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 409–419, Stroudsburg, PA, USA. Association for Computational Linguistics.

Francis, W. N. (1965). A Standard Corpus of Edited Present-Day American English. *College English*, 26(4):267–273.

Francis, W. N. and Kučera, H. (1967). *Computational analysis of present-day American English*. Brown University Press, Providence.

Gale, W. A. and Church, K. W. (1993). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–90.

Galley, M. and Quirk, C. (2011). Optimal Search for Minimum Error Rate Training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 38–49, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gao, S.-X., An, M.-J., Mao, C.-L., Xian, Y.-T., and Yu, Z.-T. (2015). A Method for Building Naxi Language Dependency Treebank Based on Chinese-Naxi Language Relationship Alignment. In *International Journal of Database Theory and Application*, volume 8(2) of *IJDTA '15*, pages 25–32.

Google, Inc. (2014). Google Translate API Pricing. `https://cloud.google.com/translate/v2/pricing`. Accessed: 01-07-2014.

Google Translate (2014). Machine Translation—Research at Google. `http://research.google.com/pubs/MachineTranslation.html`. Accessed: 01-07-2014.

Greenbaum, S. (1991). ICE: The international corpus of English. *English Today*, 7(04):3–7.

Gzip (2012). The Gzip Home Page. `http://www.gzip.org`. Accessed: 01-08-2012.

Haruno, M. and Yamazaki, T. (1996). High-performance Bilingual Text Alignment Using Statistical and Dictionary Information. *Proceedings of the 34th annual meeting of Association for Computational Linguistics*, pages 131–138.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Mach. Learn.*, 42(1-2):177–196.

Humphreys, L. (2008). Investigation into Machine Translation of English-Welsh Legal Text. *MPhil Dissertation, Bangor University.*

Hundt, M., Sand, A., and Siemund, R. (1998). *Manual of information to accompany the Freiburg-LOB Corpus of British English ('FLOB').* Albert-Ludwigs-Universität Freiburg.

Hundt, M., Sand, A., and Skandera, P. (1999). *Manual of Information to Accompany the Freiburg-Brown Corpus of American English ('Frown').* Albert-Ludwigs-Universität Freiburg.

Hutchins, W. J. and Somers, H. L. (1992). *An introduction to machine translation.* Academic Press.

Johansson, S., Leech, G. N., and Goodluck, H. (1978). *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computer.* Department of English, University of Oslo.

Kaalep, H.-J. and Veskis, K. (2007). Comparing Parallel Corpora and Evaluating their Quality. In *Proceedings of the MT Summit XI*, pages 275–279.

Katz, S. M. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Translations on Acoustics, Speech, and Signal Processing*, ASSP-35(3):400–401.

Kay, M. (1997). The Proper Place of Men and Machines in Language Translation. *Machine Translation*, 12(1):3–23.

Kay, M. and Röscheisen, M. (1993). Text-translation Alignment. *Comput. Linguist.*, 19(1):121–142.

KDE (2015). About KDE ® Community. `https://www.kde.org`. Accessed: 01-06-2015.

Khmelev, D. V. and Teahan, W. J. (2003). A Repetition Based Measure for Verification of Text Collections and for Text Categorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 104–110, New York, NY, USA. ACM.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation . In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kutuzov, A. (2013). Improving English-Russian Sentence Alignment through POS Tagging and Damerau-Levenshtein Distance. *Association for Computational Linguistics*, pages 63–68.

Lamraoui, F. and Langlais, P. (2013). Yet Another Fast, Robust and Open Source Sentence Aligner. Time to Reconsider Sentence Alignment? *XIV Machine Translation Summit*.

Lavie, A. and Denkowski, M. J. (2009). The Meteor Metric for Automatic Evaluation of Machine Translation. *Machine Translation*, 23(2):105–115.

LDC (2014). Linguistic Data Consortium: 1993-2007 United Nations Parallel Text. `https://catalog.ldc.upenn.edu/LDC2013T06`. Accessed: 01-05-2014.

Leech, G. (1992). 100 million words of English: the British National Corpus (BNC). *Language Research*, 28(1):1–13.

Lewis, D. D. (1992). Feature Selection and Feature Extraction for Text Categorization. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 212–217, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lunde, K. (2009). *CJKV information processing*. O'Reilly Media, Inc.

Ma, X. (2006). Champollion: A Robust Parallel Text Sentence Aligner. *Proceedings of the Fifth International Conference On Language Resources and Evaluation (LREC)*.

Macleod, C., Ide, N., and Grishman, R. (2002). The American National Corpus: A Standardized Resources for American English. In *Proceedings of 2nd Language Resources and Evaluation Conference (LREC)*, pages 831–836, Athens, Greece.

McEnery, A. and Xiao, Z. (2004). The Lancaster Corpus of Mandarin Chinese: A Corpus for Monolingual and Contrastive Language Study. In *Proceedings of LREC 2004*, pages 1175–1178.

McEnery, T. and Wilson, A. (2004). *Corpus Linguistics*. Edinburgh University Press, Edinburgh, Scotland, UK, 2nd edition.

Melamed, I. D. (2000). Models of Translational Equivalence among Words. *Computational Linguistics*, 26(2):221–249.

Mifflin, H. (2000). The American heritage dictionary of the English language. *Houghton Mifflin*.

Miller, C. L. (2010). Vocative syntax in Biblical Hebrew prose and poetry: A preliminary analysis. *Journal of semitic studies*, 55(2):347–364.

Moore, R. C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora. *Association for Machine Translation*, pages 135–144.

Moore, R. C. (2004). Improving IBM Word-alignment Model 1. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mújdricza-Maydt, E., Körkel-Qu, H., Riezler, S., and Padó, S. (2013). High-Precision Sentence Alignment by Bootstrapping from Wood Standard Annotations. *The Prague Bulletin of Mathematical Linguistics*, 99:5–16.

Narendra, P. M. and Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transactions on Computers*, C-26(9):917–922.

Nirenburg, S. (1989). Knowledge-based Machine Translation. *Machine Translation*, 4(1):5–24.

NIST (2012). Machine Translation Evaluation. `http://www.nist.gov/itl/iad/mig/mt.cfm`. Accessed: 01-08-2012.

NetEase, Inc. (2014). Youdao Translation API Pricing. `http://shared.youdao.com/www/about.html`. Accessed: 01-07-2014.

Nyberg, E. H., 3rd, Mitamura, T., and Carbonell, J. G. (1994). Evaluation Metrics for Knowledge-Based Machine Translation.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Comput. Linguist.*, 29(1):19–51.

OPUS (2015a). The GNOME Corpus. `http://opus.lingfil.uu.se/GNOME.php`. Accessed: 01-06-2015.

OPUS (2015b). The KDE4 Corpus. `http://opus.lingfil.uu.se/KDE4.php`. Accessed: 01-06-2015.

Ozolins, U. (2009). Back translation as a means of giving translators a voice. *Interpreting & Translation*, 1(2):1–13.

Paegelow, R. S. (2008). Back Translation Revisited: Differences that Matter (and Those that Do Not). *The ATA Chronicle*, pages 22–25.

Papageorgiou, H., Cranias, L., and Piperidis, S. (1994). Automatic Alignment in Parallel Corpora. *Proceedings of 32nd Annual Meeting of Association of Computational Linguistic*, pages 334–336.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th*

Annual Meeting of the Association for Computational Linguistics (ACL), pages 311–318.

Poesio, M. (2004). The MATE/GNOME Proposals for Anaphoric Annotation, Revisited. In Strube, M. and Sidner, C., editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 154–162, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Poesio, M. (2015). The GNOME Corpus. `http://cswww.essex.ac.uk/Research/nle/corpora/GNOME/index.htm`. Accessed: 01-06-2015.

Poplack, S. (1989). The care and handling of a megacorpus: The Ottawa-Hull French Project. *Language change and variation*, 4.

Quirk, R. (1960). TOWARDS A DESCRIPTION OF ENGLISH USAGE. *Transactions of the Philological Society*, 59(1):40–61.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A comprehensive grammar of the English language*. Pearson Longman.

Rama, T. and Borin, L. (2011). Estimating Language Relationships from a Parallel Corpus. A Study of the Europarl Corpus. In *NEALT Proceedings Series (NODALIDA 2011 Conference Proceedings)*, volume 11, pages 161–167.

Ravi, S. and Knight, K. (2010). Does Giza++ Make Search Errors? *Comput. Linguist.*, 36(3):295–302.

Riley, D. and Gildea, D. (2010). Improving the Performance of Giza++ Using Variational Bayes. *Computer Science Department, The University of Rochester*.

Sennrich, R. and Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. *Proceedings of The Ninth Conference of the Association for Machine Translation in the Americas*.

Shastri, S. V. (1988). The Kolhapur Corpus of Indian English and work done on its basis so far. *ICAME journal*, 12(15-26).

Shkarin, D. (2002). PPM: One Step to Practicality. In *Proceedings of the Data Compression Conference*, DCC '02, pages 202–211, Washington, DC, USA. IEEE Computer Society.

Simard, M., Foster, G. F., and Isabelle, P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–81.

Sinclair, J. (1987). *Looking up: An account of the COBUILD project in lexical computing and the development of the Collins COBUILD English language dictionary*. Collins COBUILD.

Sinclair, J. (2004). *Trust the text: Language, corpus and discourse*. Routledge.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas,*.

Teahan, W. J. (1995). Probability Estimation for PPM. *In Proceedings NZCSRSC'95*.

Teahan, W. J. (1998). Modelling English Text. *Ph.D. Dissertation, University of Waikato*.

Teahan, W. J. (2010). *Artificial Intelligence – Agent Behaviour I.* Ventus Publishing ApS, Frederiksberg, Denmark, 1st edition.

Teahan, W. J., Chang, Z., Humphreys, L., and Roberts, D. (2006). A compression-based method for aligning parallel corpora and evaluating machine translation systems. In *Association for Computational Linguistics*.

Teahan, W. J. and Harper, D. J. (2001). Combining PPM Models Using a Text Mining Approach. *Data Compression Conference*, pages 153–162.

Teahan, W. J., Wen, Y., McNab, R., and Witten, I. H. (2000). A Compression-based Algorithm for Chinese Word Segmentation. *Association for Computational Linguistics*, 26(3):375–393.

The British National Corpus (2007). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. `http://www.natcorp.ox.ac.uk/`.

Tian, L., Wong, D. F., Chao, L. S., Quaresma, P., Oliveira, F., and Yi, L. (2014). UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 1837–1842.

Tiedemann, J. (2009). News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In *Natural Language Processing*, volume 5, pages 237–248.

Tiedemann, J. (2010). Context Adaptation in Statistical Machine Translation Using Models with Exponentially Decaying Cache. *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15.

Tiedemann, J. and Nygaard, L. (2004). The OPUS corpus - parallel & free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Trieu, H.-L., Nguyen, P.-T., and Nguyen, L.-M. (2015). A New Feature to Improve Moore's Sentence Alignment Method. In *VNU Journal of Science: Comp. Science & Com. Eng.*, volume 31(1), pages 32–44.

Unicode Staff CORPORATE (1991). *The Unicode Standard: Worldwide Character Encoding.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Uszkoreit, J. and Brants, T. (2008). Distributed word clustering for large scale class-based language modeling in machine translation. In *In ACL International Conference Proceedings*.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2005). Parallel Corpora for Medium Density Languages. *Recent Advances in Natural Language Processing IV*, pages 247–258.

Véronis, J. and Langlais, P. (2000). Evaluation of Parallel Text Alignment Systems. The ARCADE project. *N. Ide and J. Veronis (eds.): Parallel Text Processing: Alignment and Use of Translation corpora. Kluwer Academic Publishers. Chapter 19*, pages 369–388.

Vogel, S., Ney, H., and Christoph, T. (1996). HMM-Based Word Alignment In Statistical Translation. In *International Conference on Computational Linguistics C96-2141*, pages 836–841.

Weidmer, U. (1994). Issues and Guidelines for Translation in Cross-cultural Research. *Institute for Social and Economic Research (ISER)*, 21226–1231:1226–1231.

Wu, D. (1994). Aligning a parallel English-Chinese corpus statistically with lexical criteria. *Computational Linguistics*, pages 80–87.

Wu, D. and Fung, P. (2009). Semantic Roles for SMT: A Hybrid Two-pass Model. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 13–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wu, P. (2007). Adaptive Models of Chinese Text. *PhD Dissertation, University of Wales: Bangor*.

Wu, P. and Teahan, W. J. (2005). Modelling Chinese for Text Compression. In *Data Compression Conference, 2005. Proceedings. DCC 2005*, page 488.

Wu, P. and Teahan, W. J. (2008). A New PPM Variant for Chinese Text Compression. *Nat. Lang. Eng.*, 14(3):417–430.

Xiong, D., Liu, Q., and Lin, S. (2006). Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 521–528, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xu, J., Zens, R., and Ney, H. (2005). Sentence Segmentation Using IBM Word Alignment Model 1. In *Proceedings of EAMT 2005 the 10th Annual Conference of the European Association for Machine Translation*, pages 280–287.

Yu, Q., Max, A., and Yvon, F. (2012). Revisiting Sentence Alignment Algorithms for Alignment Visualization and Evaluation. *LREC Workshop*, pages 10–16.

Appendices

Appendix I

10 examples of satisfactory translations with higher SLR values from the KDE4 corpus:

```
Sentence Number:  21138 / SLR: 1.905 / CR: 1.050
This box is used to specify the element on which calculation is
to be performed.
此框用于列出需要计算的元素。

Sentence Number:  20085 / SLR: 1.608 / CR: 1.000
Check to specify the number of times the alarm should repeat
after each recurrence
选中此处可指定每次重现后提醒的次数

Sentence Number:  54395 / SLR: 1.762 / CR: 1.050
A circle defined by its center and the length of a segment as the
diameter.
由圆心和某线段的长度来构造圆。
```

Sentence Number: 95065 / SLR: 1.762 / CR: 1.000

Calculates the inverse of the matrix.

计算矩阵转置。

Sentence Number: 67408 / SLR: 1.729 / CR: 1.000

Remember this identity, so that it will be used in future composer windows as well.

记住此身份，以后用于撰写器窗口。

Sentence Number: 21278 / SLR: 1.727 / CR: 1.062

Amount of solvent is always specified in terms of volume.

溶剂的量通常以体积计。

Sentence Number: 16695 / SLR: 1.704 / CR: 1.000

You are not authorized to remove this service.

您无权删除此服务。

Sentence Number: 96055 / SLR: 1.653 / CR: 1.080

If the start value is greater than the end value the step must be less than zero.

如果首项比末项大，公差必须小于0。

Sentence Number: 70845 / SLR: 1.788 / CR: 1.000

Use this to close the dialog and return to the application.

关闭对话框并返回程序。

Sentence Number: 66151 / SLR: 1.600 / CR: 1.056

Are you sure you want to empty the trash folder?

您确定要清空废件夹?

Appendix II

10 examples of unsatisfactory translations with lower SLR values from the KDE4 corpus:

Sentence Number: 66690 / SLR: 1.043 / CR: 1.455
Default forward template
默认转发模板12345

Sentence Number: 53132 / SLR: 1.095 / CR: 1.500
New GnuPG Home Location
新的主配置地址

Sentence Number: 6837 / SLR: 1.000 / CR: 2.182
Use < ,< =,:, > = and >.
使用、以及符号。

Sentence Number: 23379 / SLR: 1.034 / CR: 1.818
Create a CTags database file.
创建一个数据库文件。

Sentence Number: 43316 / SLR: 1.083 / CR: 1.800
Path to Kexi database file
数据库文件的路径

Sentence Number: 70388 / SLR: 1.067 / CR: 1.800

Maximum is at x = %1, %2(x) = %3

当123时，可取得最大值

Sentence Number: 57168 / SLR: 1.037 / CR: 1.857

Flalign* -

beginflalign*

无编号左右对齐公式

Sentence Number: 6212 / SLR: 1.048 / CR: 1.857

Simple PHP Application

简单的应用程序

Sentence Number: 38043 / SLR: 1.028 / CR: 1.692

%1 is not a whole number of minutes.

文件\1"不是有效的插件。

Sentence Number: 11058 / SLR: 1.000 / CR: 1.545

IMAP Server via KMail

通过访问服务器

Appendix III

10 examples of satisfactory translations with higher SLR values from the GNOME corpus:

Sentence Number: 61 / SLR: 1.333 / CR: 1.000
This plugin checks applications for accessibility problems and generates a report including the severity and description of the problems. The report links errors to documentation about how to remedy common problems. The plugin is extensible with test schemas that define rules for validation.
插件因访问问题并生成一个包括问题严格描述的报告来检测应用程序。关于怎样补救一般问题的文档的链接错误报告。对于定义确定规则的测试模式插件是可扩展的。

Sentence Number: 663 / SLR: 1.303 / CR: 1.000
The default plugin layout for the top panel
上方面板的默认插件布局

Sentence Number: 664 / SLR: 1.704 / CR: 1.000
A list of plugins that are disabled by default
默认禁用插件的列表

Sentence Number: 668 / SLR: 1.278 / CR: 1.000
The color and opacity of the highlight border.
高亮边框的颜色和不透明。

Sentence Number: 672 / SLR: 1.238 / CR: 1.000

Browse the various methods of the current accessible

浏览当前可访问对象的各种方法

Sentence Number: 1271 / SLR: 1.278 / CR: 1.000

The window width value.

窗口宽度值。

Sentence Number: 1287 / SLR: 1.267 / CR: 1.000

Hotkey combination for related action.

相关动作的热键组合。

Sentence Number: 2317 / SLR: 1.091 / CR: 1.000

What is Accessibility?

什么是辅助功能?

Sentence Number: 2604 / SLR: 1.444 / CR: 1.000

The following sections contain examples of the gestures that you can add to the GDM configuration files.

下面章节包含了您可以添加到配置文件里的手势例子。

Sentence Number: 2693 / SLR: 1.333 / CR: 1.100

Complete the move operation.

完整的移动操作

Appendix IV

10 examples of unsatisfactory translations with lower SLR values from the GNOME corpus:

```
Sentence Number:  63380 / SLR: 1.000 / CR: 2.167
Brasero | %s (Video Disc)
您的项目没有保存。

Sentence Number:  48933 / SLR: 1.000 / CR: 1.556
Unable to read your iPod
为什么要这样呢?

Sentence Number:  13568 / SLR: 1.000 / CR: 1.818
Foundations are built up in suit.
收牌区按花色递增收牌。

Sentence Number:  19875 / SLR: 1.000 / CR: 1.462
Enable HTML tags folding
自动补全当前的词

Sentence Number:  44543 / SLR: 1.000 / CR: 1.455
Tools, Filters and Plug-ins
改进的自由选择工具

Sentence Number:  62003 / SLR: 1.000 / CR: 1.643
The drive has no rewriting capabilities
新光盘在含有源盘的刻录机中
```

```
Sentence Number:  83461 / SLR: 1.000 / CR: 1.533
RSS feed for %(lang.get_name)s
```
已翻译模糊翻译未翻译

```
Sentence Number:  44274 / SLR: 1.000 / CR: 1.429
Copy Files/Folders...
```
切换到分支标签

```
Sentence Number:  93726 / SLR: 1.000 / CR: 1.615
Tag is already attached to a file
```
文件打开时出错: \"。

```
Sentence Number:  98568 / SLR: 1.000 / CR: 1.459
If you alter, transform, or build upon this work, you may
distribute the resulting work only under the same, similar or
a compatible license.
```
自动保存您和联系人的所有文字会话。您可以搜索之前全部的会话或者根据联系人和日期浏览之前的会话。