NAIST-IS-DD1361023

Doctoral Dissertation

Studies on improving two fundamental steps for Chinese natural language processing: word segmentation and spelling check

Fei Cheng

March 15, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Fei Cheng

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Masashi Shimbo	(Member)
Assistant Professor Hiroyuki Shindo	(Member)
Assistant Professor Kevin Duh	(Johns Hopkins University)

Studies on improving two fundamental steps for Chinese natural language processing: word segmentation and spelling check*

Fei Cheng

Abstract

In Chinese, a sentence is written as a sequence of Chinese character without any indicators of word boundaries. Therefore, **Chinese word segmentation** is generally thought as the fundamental step in the Chinese Natural Language Processing (NLP) pipeline. In the meanwhile, **Chinese spelling check** is an automatic mechanism to detect and correct human errors in unsegmented Chinese documents, which can be seen as the prior process before word segmentation. **Chinese word segmentation** and **spelling check** play such crucial roles as to directly affect all the other downstream Chinese NLP tasks such as Part-of-Speech tagging, syntactic parsing and etc. However, both these two tasks are facing some challenges.

In **Chinese word segmentation**, there are two main issues remaining. First, various word segmentation standards keep the existing corpora from being used in combination. Second, highly productive Chinese synthetic words increase out-of-vocabulary words. We believe that both issues can be addressed by analyzing the internal structure of words. For this purpose, we construct a dictionary of synthetic words with the internal structure manually annotated. By taking the dictionary as training data, we propose a machine learning based word structure parser which can automatically analyze the internal structure of input words. We demonstrate that the word segmentation performance can be improved by parsing the internal structure of the words in the training data. Furthermore, we

*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1361023, March 15, 2018.

propose a simple strategy to transform two different Chinese word segmentation corpora into a new consistent segmentation level, in which the two corpora are used in combination. The larger combined training data is verified to be highly consistent by 10-fold cross-validation. With the help of larger training data size and internal structure of words, our segmentation system achieves state-of-the-art performance on the test data of the two corpora.

As for **Chinese spelling check**, spelling errors are hard to be detected without word boundary information and must be considered within a context. Our two-stage spelling check system first enlarges the candidate lists by gathering corrections generated by several individual systems. The second ranking component is to select the most possible correction to balance precision and recall.

Keywords:

Internal Structure, Synthetic Word, Parsing, Word Segmentation, Spelling Check

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deep gratitude to my supervisor, Professor Yuji Matsumoto, whose wisdom, expertise, understanding and patience support me during the period of my master and PhD courses. As a newbie to Natural Language Processing, I continuously receive progressive training by discussing with Professor Matsumoto. Not only in the research, Professor Matsumoto also gave me many supports and private space in life. It's my honor to be Professor Matsumoto's student.

I would also like to thank my previous Assistant Professor Kevin Duh for helping me to construct a correct thought of research. He taught me a detailed framework of research: a new idea, quick iterations, evaluation and writing papers. As a western-educated researcher, he always encouraged us to talk more, ask more. After transferring to John Hopkins University, he still help me with my research and the new submission of my paper.

Also, there are many thanks to Professor Satoshi Nakamura, Associate Professor Masashi Shimbo, Assistant Professor Hiroyuki Shindo for their insight comments and advices on the thesis. Associate Professor Masashi Shimbo and Assistant Professor Hiroyuki Shindo gave me very useful discussions during the Kenkyukai and Benkyokai meetings. As for two previous Assistant Professor Masayuki Asahara and Mamoru Komachi, I would like to thank their help to me for starting my study and life in NAIST.

The final thank is to all the members of Matsumoto-ken, who formed a world-class team. They gave much help to the international students like me. To an OB student Jia Lu, thank you for the help during my first half of a year in NAIST. Special thank to Yuko Kitagawa.

To my family, I would like to appreciate my parents for supporting me to study abroad. Thanks to my wife, I know that both of us were facing lots of difficulties to keep going during the life in Japan. I'm happy to see that you got your PhD degree and found a job. Thanks to my four years old daughter, I wish that you would keep you independent thinking and always be as happy as right now.

Chapter 1

1. Introduction

This chapter sets the general view of the dissertation. Section 1.1 introduce the motivation for improving the current issues in two fundamental prior steps of Chinese natural language processing: Chinese word segmentation and spelling check. Section 1.2 summarizes the contributions of this dissertation. The outline of the whole dissertation is listed in Section 1.3.

1.1 Motivation

Chinese word segmentation (CWS) is commonly thought as the fundamental process in the Chinese natural language processing pipeline, for the reason there is no word delimiter in Chinese such as 'space' in English. Two issues with the conventional pipeline methods involving word segmentation are (1) the lack of a common segmentation standard and (2) the poor segmentation performance on OOV words. These two issues may be circumvented if we adopt the view of character-based parsing, providing both internal structures to synthetic words and global structure to sentences in a seamless fashion. We believe that synthetic word analysis is a potentially important but relatively unexplored problem in Chinese natural language processing. However, the accuracy of synthetic word parsing is not yet satisfactory, due to the lack of research.

In this thesis, we first focus on building a Chinese synthetic word dictionary with internal structures annotated, which is a potential useful resource for other down-streaming Chinese NLP tasks. Then, we develop a synthetic word parser to predict internal structures of words. Finally, a standard CRF-based segmenter is adopted to verify the improvement of CWS performance obtained from internal structure information inside words.

As a consequence of the lack of common segmentation standard, the Chinese word segmentation corpora accomplished by different research groups can hardly achieve cooperation. We further proposed a simple strategy to transform different

CWS corpora to a common segmentation level, in which multiple CWS corpora can be simply combined to achieve larger training data. The extended training data is verified to be highly consistent. We make a detailed investigation of the improvement brought by the extension of training data and additional internal structure information.

Spelling check is an automatic mechanism to detect and correct human errors, which is a common task in every written languages. In Chinese natural language processing, spelling check can be seen as the prior step before word segmentation. However, Chinese spelling check (CSC) is different from that in other alphabetical languages. There are no word delimiters in raw documents and most words are with short lengths (usually one to three characters). Therefore, error detection is very hard in Chinese and must be considered within a surrounding context, not just within a single word (such as spelling check in English). The Seventh SIGHAN Workshop establishes a share task for Chinese spelling check with two sub-tasks: error detection and error correction, as shown in Figure 1.

Two popular CSC approaches are: language model based (LM-based) method and statistic machine translation based (SMT-based) method. LM-based method is to calculate the scores of all possible sentences generated by replacing each character with a new character in its confusion set (a collection of candidates for the spelling error). SMT-based method is another solution, which treats a input sentence with errors as 'source language' and the correct sentence as 'target language'.

In this thesis, we develop a two-stage hybrid CSC approach, which first combines the correction candidates generated by several single models. In the second stage, a Support Vector Machines (SVMs) model is used to rank each candidate list to generate the most possible character. Our proposed model is expected to obtain a large correction candidate list for each character compared to a single model method and context, dictionary or class-based features can be easily incorporated into our SVMs ranking step.

1.2 Contribution

Our contributions are summarized as follows:

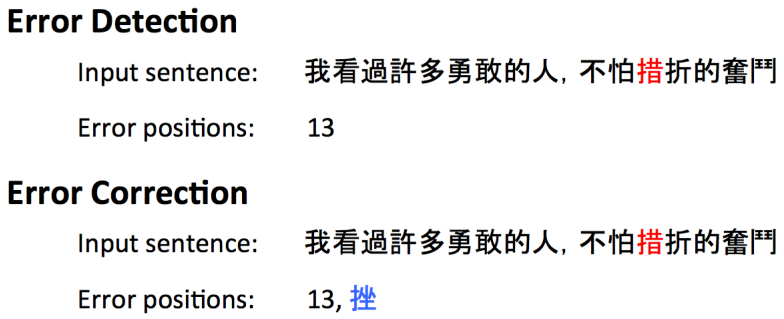

Figure 1. Two common sub-tasks of Chinese spelling check: error detection and error correction.

- We construct a useful dictionary of 31,849 synthetic words with internal structure information annotated.
- We define a character-based dependency framework for analyzing internal structure of Chinese synthetic words and boost parsing performance by extracting additional features from a dictionary and a large-scale unlabeled corpus.
- We propose a method to improve Chinese word segmentation performance by incorporating internal structure information inside Chinese synthetic words. Our system transforms the words in the original training data into a fine-grained segmentation level, and achieves state-of-the-art word segmentation performance.
- We propose a strategy to transform two CWS corpora of the the Second International Chinese Word Segmentation Bake-off data to a consistent segmentation level, in which multiple corpora can be simply combined to extend larger training data. The extension of training data and the flexibility of incorporating internal structure information of our pipeline word segmentation system show significant improvement of the word segmentation performance.
- We propose a two-stage hybrid approach for Chinese spelling check, which contains two key steps: **correction generation** and **correction ranking** Our spelling check approach is simple, effective and consuming low

resources. The final test shows that our approach obtains competitive results of the state-of-the-art systems, which used much more resources.

1.3 Outline of Dissertation

For the lack of available resources for analyzing internal information of Chinese words, we build a synthetic word dictionary with internal tree structure manually annotated.

In Chapter 2, we conduct following processes to reach the goal of analyzing the word structure.

- establish the annotation standard and construct a synthetic word dictionary with the internal structure annotated.
- design a character-based morphological dependency framework to represent different structure types of Chinese synthetic words.
- a graph-based dependency parser is implemented to perform the parsing work and several types of features extracted from a dictionary and a large-scale unlabeled are incorporated to boost our parser.

In Chapter 3, we introduce our word segmentation model, which is enhanced by using the internal structure information inside words. The basic idea is to:

- use the synthetic word parser described in Chapter 2 to parse the internal structure of the words in current word segmentation training data to convert the data into a fine-grained level.
- use a CRF-based segmenter to predict the combined position label of each character, which contains both the original annotated and new fine-grained information.

In Chapter 4, we introduce our pipeline word segmentation system, which is benefited from the extension of larger training data and the flexibility of incorporating internal structure information.

- propose a strategy to transforms multiple CWS corpora to a common segmentation level for a easy extension of larger training data.

- the common segmentation data is flexible to introduce internal structure further.

In Chapter 5, we introduce a two-stage hybrid Chinese spelling check system with two key steps: correction generation and correction ranking. The basic idea is to:

- enlarge the correction candidate list by merge the results from a LM-based model and a SMT-based model.
- adopt a SVMs model to predict a confidence score for each correction in a candidate list of a character in sentences. Each candidate list is ranked by the score and the top character is treated as the most possible correction.

In Chapter 6, we summarize this dissertation and discuss the future direction of this work.

Chapter 2

2. Analyzing the Internal Structure of Chinese Synthetic Words

In this chapter, we construct an useful dictionary of Chinese synthetic words with the internal structure manually annotated and propose a character-based dependency framework for analyzing the word structure. In Section 2.1, we introduce the motivation and background of Chinese synthetic word parsing. We define the classification of Chinese words in Section 2.3. The annotation standard is carefully established and the detailed annotation work is introduced in Section 2.4. In Section 2.5 , we introduce a character-based dependency representation of the word structure and several feature types extracted from a dictionary and a large-scale unlabeled corpus are incorporated into our parser. In Section 2.6, a series of experiments are conducted to evaluate the performance of our synthetic word parser and the usefulness of features. The summarization of this chapter is at the last part.

2.1 Introduction

Unlike Indo-European languages, such as English, a sentence in Chinese is written as continuous characters without distinct word boundaries. Chinese word segmentation (CWS) is commonly treated as the first step, before part-of-speech (POS) tagging, parsing, and other components in the natural language processing (NLP) pipeline. The dominant approaches treat word segmentation as a character-based sequential labeling problem. Conditional Random Fields (CRFs) is the common learning model applied in this task. This method offers both robust performance and flexibility to incorporate features.

Unfortunately, there is not a clear and intuitive notion of 'word' in Chinese. A highly controversial part is that Chinese synthetic words have a quite complex structure and could be represented by several segmentation levels as shown in Figure 2. The immediate consequences are two issues: the variety of word

Figure 2. The variety of the segmentation levels of a sample synthetic word.

segmentation standards and highly productive out-of-vocabulary (OOV) words. Thus, one research line is to find more useful statistic or class-based features to improve segmentation performance. On the other hand, both two issues indicate one common solution, which is to investigate internal information inside Chinese words. However, less available resources can be found for analyzing internal structure of Chinese words.

In this work, we propose a series of processes for the purpose of analyzing the internal structure information of words. Constructing a resource with internal information annotated is the first step for parsing word structure by using the supervised learning methods. We first introduce the classification of 'word' in Chinese. Based on this classification, we carefully assign the annotation standard and the annotation work is completed by four students. Then, we design a character-based dependency relation framework for jointly analyzing segmentation and structure parsing, which is implemented by a graph-based dependency model. For boosting the parsing performance, several statistic and cluster features extracted from a dictionary and a large-scale unlabeled corpus are incorporated into our parsing model. Finally, a 10-fold cross validation is used to evaluate the performance of our character-based dependency framework, compared to the traditional pipeline method. Furthermore, we make a detailed investigation of the improvement brought by different features.

2.2 Related Work

Recently, some work on using the internal structure of words to improve Chinese process show promising results on different tasks. Li [18] claimed the importance of word structures. They proposed a new paradigm for Chinese word segmentation in which not only flat word structures are identified but with internal structures are also parsed in a sentence. They aimed to integrate word structure information to improve the performance of word segmentation, parsing or other NLP tasks on sentences. Zhang et al. [46] manually annotated the structures of 37,382 words, which covered the entire Chinese tree bank 5 (CTB5). then, they built a shift-reduce parser with the customized actions designed to jointly perform word segmentation, Part-of-speech tagging and phrase-structure parsing. Their system significantly outperformed the state-of-the-art word-based pipeline methods on CTB5 test.

However, these work reply on prior knowledge of internal structure information on CTB5, which is provided by the manual annotation processes. Our work aims to construct a automatic mechanism to analyzing the structure of a word accurately, which is helpful to any tasks in the Chinese NLP pipelines. For instance, given a synthetic word parser, we can easily reconstruct a character-level CTB5 corpus to benefit the parsing task in a similar way as Zhang et al. [46] without relying an additional annotation work. Although the result might not be as good as the performance based on the gold annotation data, a robust synthetic word parser can reach a close performance to manual annotation and adapt to any other word-based resources.

Our character-based word parsing model is inspired by the work [24, 47]. Lu et al. [24] described the semantic relations between characters. They proposed a structure analysis model for three-character Chinese words. Zhao [47] presented a character-level unlabeled dependency scheme as an alternative to linear representation of sentences for word segmentation task. Their results demonstrated that the character-based dependency framework can obtain comparable performance compared to the state-of-the-art word segmentation models.

Instead of adopting a traditional pipeline method with word segmentation and parsing to analyze the word structure, we extends these previous work by proposing a character-based morphological dependency framework to represent

the internal tree structure of words. In the experiments, our framework significantly outperforms the pipeline method without relying any extra resources. In additional, we further boost our word structure parser by extracting statistic and cluster features from a large-scale unlabeled corpus and a dictionary. Our word structure parser is expected to provide reasonable performance to convert the segmentation standards of the existing word segmentation corpora to a fine-grained level (Section 3.3.1).

2.3 Definition of 'Word' in Chinese

In Chinese, **word** is generally considered as an ambigous definition, because a clear delimiter **space** does not exist. For Chinese speakers, a word is a lexical entry, representing a whole meaning. In this chapter, we adopt the simple classification of Chinese word proposed by Lu et al. [24], which divided Chinese words into the following two types.

- **Single-morpheme Word**: Those words only have one morpheme inside them and cannot be segmented further. It means that the meanings of the individual parts do not indicate the meaning of the original word. The following are three sub-types of single-morpheme words:

  - **One-character single-morpheme word**:
    人 (human), 睡 (sleep), 热 (hot)
  - **Multi-character single-morpheme word**:
    葡萄 (grape), 徘徊 (wander), 彷徨 (hesitate)
  - **Transliterated word**: Those words are usually translated from other languages on pronunciation.
    麦当劳 (McDonald's), 瓦伦西亚 (Valencia), 大卫・贝克汉姆 (David Becham)

- **Synthetic Word**: These words are composed of two or more single-morpheme words and represent a new meaning which can be indicated from the internal constituents. Figure 3 is the internal structure of the synthetic word 总 / 工程 / 师 (chief engineer):

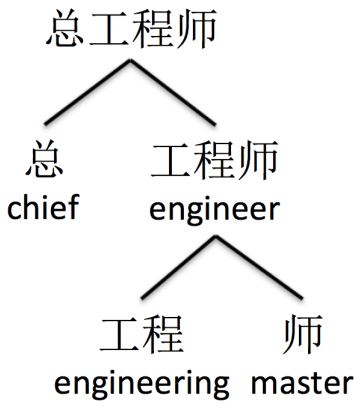

Figure 3. An example of the internal tree structure of a Chinese synthetic word.

> Given the internal tree structure of the word 总 / 工程 / 师 , the meaning can be inferred as 'chief master of engineering'.

A special case in Chinese is some two-character words, such as 工程 (engineering) can be further segmented into two single character 工 (craft) and 程 (process). In this thesis, we treat this type of two-character words in the same way as single-morpheme words, which are the minimal units (leaf) in a parse tree.

2.4 Annotation

A serious challenge for internal structure analysis of synthetic words is the lack of available resources. Therefore, we decide to manually annotate the internal structure of Chinese synthetic words in Cradle [25], which is a lexicon management system with friendly interfaces to annotate and present internal tree of words (Figure 4).

2.4.1 Annotation Standard

Given a list of Chinese synthetic words, we carefully establish the following annotation based on the definition of Chinese word in Section 2.3.

- Determine whether the current word is a synthetic word or not. If it is a single-morpheme word, the annotator skips to the next word in the list.

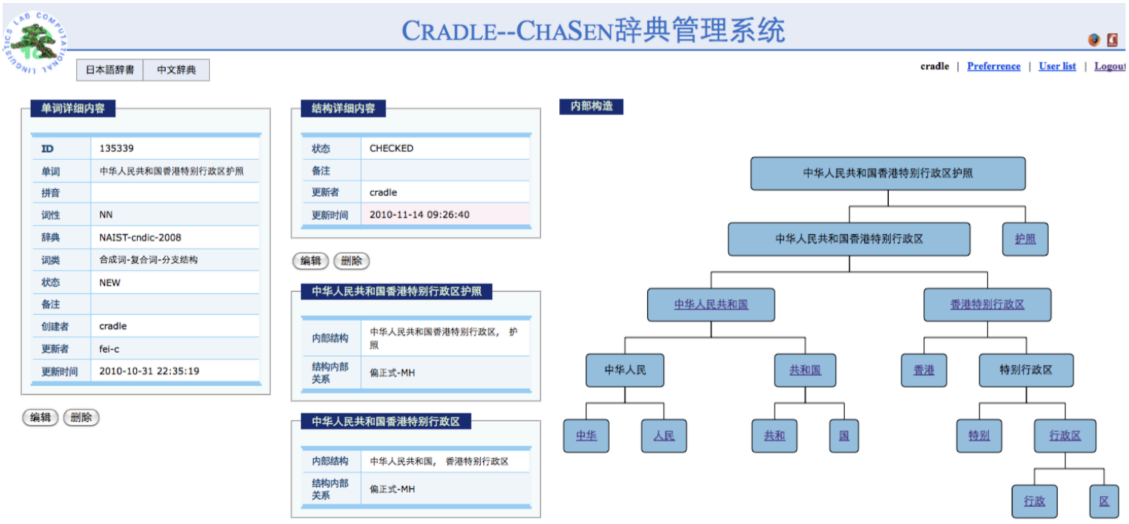

Figure 4. A tree structure example of a synthetic word in Cradle.

- Split the target word into meaningful parts (normally two parts on each level) from top to bottom.
- Stop annotating until that all the split parts are single-morpheme words.

2.4.2 Annotation Data

Chinese Wikipedia is a rich extending resource, which contains plenty of Chinese synthetic words. An article title in Chinese Wikipedia generally takes a independent meaning, such as a place, a historical event or a technical term. There are 826,557 article titles in our 2012 crawl of Chinese Wikipedia. According to our annotation standard, four students randomly annotated 31,849 words [1] with the length distribution shown in Table 1. Each student's annotation is revised by another student.

For investigating the quality of our annotation, we required two of the students to annotate additional 200 words. We evaluate the annotation agreement in two levels. They first do word segmentation on the input words. Secondly, they annotate brackets on the gold segmented words. The Kappa-coefficient on

[1] `https://github.com/racerandom/chcomparser`

character-Length	Number of Words
3-character	21848
4-character	2292
5-character	1838
6-character	1516
7-character	1433
$\geq$ **8-character**	2992

Table 1. The word length distribution of our annotated data.

the word boundary between characters in the first step is 0.947. The Kappa-coefficient on matching the brackets is 0.921.

Although the definition of 'word' in Section 2.3 is clear to follow, there are still some vague cases in the middle of single-morphine words and synthetic words. For instance, the word 工程 (engineering) is categorized as a single-morphine word in the modern Chinese. However, 工程 is derived from the meaning of the characters 工 (craft) and 程 (process). These words are paraphrased from western terms (e.g. 社会 society, 纪律 discipline), by compounding two Chinese characters based on their meanings. It suggests that the internal structure of Chinese words can be annotated into a deeper character level in the further studies. Certainly, the annotation requires more linguistic knowledge of Chinese characters.

2.5 Character-based Dependency model

Parsing is a common NLP task for analyzing the syntax structure of sentences. Dependency grammar [12] is a modern syntactic framework, which presents the sentence structure as all dependency relations (each dependency points to the a single parent, from the modifier). Intuitively, the internal tree structure of words can be seen as a small-scale analog of the sentence structure and dependency parsing is a natural method to reach our goal.

However, correctly parsing Chinese synthetic words is challenging, not only because word segmentation step exists, but also for the reason that standard part-of-speech (POS) tags provide limited information. For instance, 中国 NN / 国 际 NN / 广播 NN / 电台 NN contains a sequence of identical NN tags, giving little clue about their internal branching structure. Our work is concerned

with parsing Chinese synthetic words into a parse tree without replying on POS tagging.

For these reasons, we design a character-based dependency model for predicting internal word structure. Instead of using a traditional pipeline method with word segmentation and word level parsing processes. Our model allows joint word segmentation and internal structure parsing. Each dependency arc in our model represent a morphological relation between two characters. The label set of the model is introduced in Table 2.

Label	**Relation**
B	Branching relation (external)
C	Coordinate relation (external)
WB	Beginning inside a single-morpheme word (internal)
WI	Other part inside a single-morpheme word (internal)

Table 2. The label set of character-level morphological dependency relations.

These four morphological dependency relations can be further classified into two categories: internal and external. **B** and **C** are used to present the external relation between two words. **WB** and **WI** are used to present the internal relation of two character inside a single-morpheme word. However, the direction of the internal relation is ambiguous to be assigned, even for native speakers. In this work, all the single-morpheme words take a flat structure, which means that the modifier always points to the head from left to right for all the internal relations. For instance, an example of the flat structure of the transliteration (single-morpheme) word 奥林匹克 (Olympic) is shown in Figure 5.

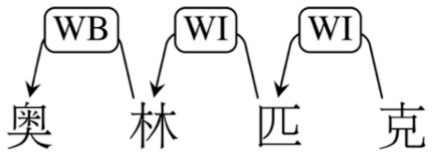

Figure 5. An example of the flat structure of a transliteration word

2.5.1 Internal Structure with Character-level Dependency Representation

Before illustrating the internal structure of words by our dependency relation label set, we classify two morphological structure types of Chinese words as follows.

Branching is the most common morphological relation connecting two internal parts. The branching structure of a tree-character word **ABC** can be enumerated as three sub-types: **A+BC**, **AB+C** and **A+B+C**. For instance in Figure 6, 副总统 (vice president) is an **A+BC** type synthetic word composed by two single-morpheme words 副 (vice) and 总统 (president). The word 联系人 (contact person) is composed by two single-morpheme words 联系 (contact) and 人 (person). 中日韩 (China, Japan and Korea) is an **A+B+C** type word composed by three single-morpheme words 中 (China), 日 (Japan) and 韩 (Korea) with coordinate relations between each other.

Merging is the other morphological phenomenon in Chinese. It means two semantically related words sharing a common internal part, can be merged into one word by removing one of the common parts. The Merging structure of a word **ABC** can be also classified into three sub-types: **AB+AC**, **AC+BC** and **AB + BC**. For instance in Figure 7, a word **ABC** 国内外 (domestic and overseas) can be composed by **AB** 国内 (domestic) and **AC** 国外 (overseas), which are sharing the common **A** 国 (country). The example 动植物 (animal and vegetation) is consisted by 动物 (animal) and 植物 (vegetation) sharing the common right character 物 (object). 干电池 (dry cell) takes the **AB + BC** structure consisted by two words 干电 (dry power) and 电池 (battery) sharing the common middle character 电 (electricity).

The internal structure of a long synthetic word ’Olympic Games’ can be represented as a character-level dependency tree as shown in Figure 8. 奥林匹克 with the labels **WB WI WI** represents a single-morpheme transliterated word ’Olympic’. 运动会 (sports competition) is composed by two single-morpheme words 运动 (sports) and 会 (competition). 奥林匹克 and 运动会 take a external branching relation between them.

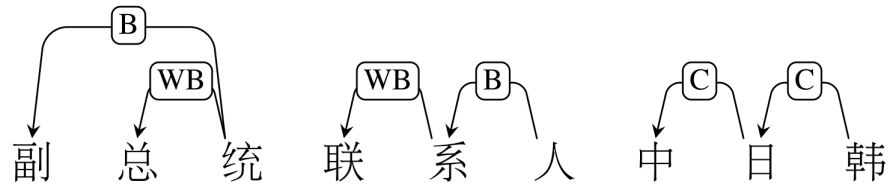

Figure 6. Branching structures of three-character synthetic words

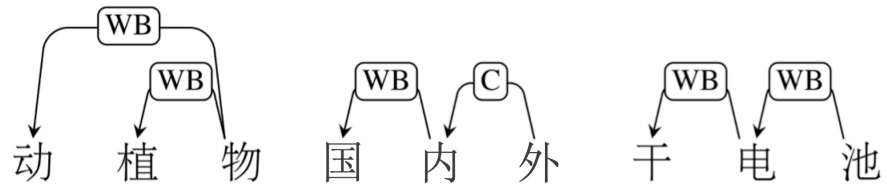

Figure 7. Merging structures of three-character synthetic words

2.5.2 Graph-based Dependency Parsing

Graph-based dependency parsing algorithm [12, 26] defines the score of a dependency graph as the sum of the scores of all the arcs $s(i, j, l)$ it contains. Here, $s(i, j, l)$ is the arc between words i and j with label l. This problem is equivalent to finding the highest scoring directed spanning tree in the complete graph over the input sentence. It is represented by:

$$s(x, y) = \sum_{(i,j,l) \in y} s(i, j, l) \tag{1}$$

Second order sibling factorization [26, 4] is proposed as an extension of the original graph-based dependency parsing, which shows the significant improvement compared to first order parsing. The score of a tree for second order parsing is

$$s(x, y) = \sum_{(i,k,j,l) \in y} s(i, k, j, l) \tag{2}$$

which is the sum of adjacent edge score in y. The new score function contains k as the middle modifier between i and j, which can be easily reduced to standard first-order model by ignoring k.

The implementation of graph-based parsing model adopted in this work is

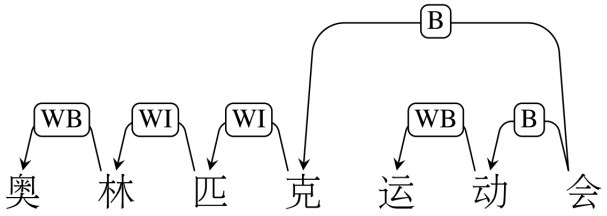

Figure 8. An example tree structure of a long word with our dependency representation

MSTParser[2] [26].

2.5.3 Feature Types

Our task can be seen as a joint prediction for position information (the word segmentation task) and structure information (the parsing task). Both tasks are inspired by the improvement from statistic features [13, 35] or cluster-based features [23, 16] in the past decade. Therefore we plan to boost our parser by incorporating the features extracted from two existing resources: a dictionary (Naist Chinse dictionary [3] and a large-scale unlabeled corpus (Chinese Gigaword second edition [4]).

For easily demonstrating the feature types used in our graph-based synthetic word parser, we denote a character c_i to a character c_k and c_j with their context $[...c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}...]$, $[...c_{k-2}, c_{k-1}, c_k, c_{k+1}, c_{k+2}...]$ and $[...c_{j-2}, c_{j-1}, c_j, c_{j+1}, c_{j+2}...]$. Table 3 is the template of all the context character sequence features including both first order and second order types.

We further classifier the into three categories: **Local**, **Long**, **2-order**. **Local** features include the uni-gram c_s to 4-gram $c_s c_{s+1} c_{s+2} c_{s+3}$ character sequence surrounding c_i or c_j. Considering the difference of our character-based word structure parsing from the normal word-based parsing, **Local** plays an important role to detect the internal relations (Table 2), which are designed to indicate the position of a character inside a word. **Long** features are responsible to detect

[2]https://sourceforge.net/projects/mstparser/

[3]https://cl.naist.jp/index.php?%B8%F8%B3%AB%A5%EA%A5%BD%A1%BC%A5%B9%2FNCD

[4]https://catalog.ldc.upenn.edu/LDC2005T14

Type	Feature Template
Local	c_s $(i-2<s<i+2 \quad and \quad j-2<s<j+2)$
	$c_s c_{s+1}$ $(i-3<s<i+2 \quad and \quad j-3<s<j+2)$
	$c_s c_{s+1} c_{s+2}$ $(i-4<s<i+3 \quad and \quad j-4<s<j+3)$
	$c_s c_{s+1} c_{s+2} c_{s+3}$ $(i-5<s<i+4 \quad and \quad j-5<s<j+4)$
Long	$c_i + c_j, c_i + c_{j-1}c_j, c_i + c_{i-2}c_{i-1}c_i$
	$c_{i-1}c_i + c_j, c_{i-1}c_i + c_{j-1}c_j, c_{i-1}c_i + c_{j-2}c_{j-2}c_j$
	$c_{i-2}c_{i-1}c_i + c_j, c_{i-2}c_{i-1}c_i + c_{j-1}c_j, c_{i-2}c_{i-1}c_i + c_{j-2}c_{j-2}c_j$
2-order	$c_k + c_j, c_{k-1}c_k + c_j, c_k + c_{j-1}c_j$
	$c_i + c_k + c_j, c_{i-1}c_i + c_k + c_j$
	$c_i + c_{k-1}c_k + c_j, c_{i-1}c_i + c_{k-1}c_k + c_j$
	$c_i + c_{k-1}c_k + c_{j-1}c_j, c_{i-1}c_i + c_{k-1}c_k + c_{j-1}c_j$

Table 3. The character sequence feature template for the synthetic word parser. '+' denotes the combination of two character sequences. For Instance, the Brown Cluster feature of the sequence c_i+$c_{j-1}c_j$ is the combination of two Brown Clusters of c_i and $c_{j-1}c_j$.

whether a external relation should be established between two head characters. **2-order** features help our parser to more accurately predict sibling cases. For instance, 副总统 in Figure 7 takes a most common **Merging** structure in which two modifiers 副 and 总 point to a common head 统 with two different external **B** and internal **WB** relations.

The feature types extracted from the extra resources are list as follows:

- **Dictionary Feature**: If a context character sequence in the feature template (Table 3), exists in the NAIST Chinese dictionary (with 129,560 entries), the existing POS tags in the dictionary of the sequence are used as features. It is possible for one word to correspond to multiple POS tags in the dictionary. For instance, 稳定 contains two possible POS tags: 'NN' (noun) and 'VV' (verb) in the dictionary.

- **Brown Cluster Feature**: Koo et al. [16] trained a dependency parser in English and Czech and used Brown clusters [2] with different lengths as additional features. we use our basic CRF-based segmenter (Section 3.3.2) to do word segmentation on Chinese Gigaword second edition. Then we

conduct a word-level Brown clustering on the segmented corpus. If a context character sequence in the feature template (Table 3), exists in the word list of the segmented corpus, its corresponding Brown cluster id is used as a feature. '$k = 100$' reaches highest parsing performance in the cross-validation experiments.

- **Accessor Variety Feature**: Feng et al. [13] first introduce the accessor variety (AV) to identify meaningful Chinese words. The number of the distinct occurrence (accessor variety) of character types before or after a target character is a important statistic indicator to evaluate how likely word boundaries surround it. The access variety value AV of a character sequence s defined as following formula [48] is adopted as statistic feature into their segmenter.

$$AV(s) = min\{L_{av}(s), R_{av}(s)\} \tag{3}$$

where the left and right AV values $L_{av}(s)$ and $R_{av}(s)$ are defined, respectively, as the numbers of its distinct predecessor and successor characters. Since AV is a statistic number of distinct character types, the sparse data problem exists. Zhao and Kitalleviate [48] narrow down the feature representation to alleviate the sparse problem by the new feature function for the $AV(s)$ of a character sequence s as follows.

$$f(s) = t, \quad if\ 2^t \leq AV(s) < 2^{t+1}, \tag{4}$$

where t is an integer to logarithmize the score. The AV score of a character sequence is proportion to the possibilities of the left and right word boundaries surrounding it, which indicates s is a meaningful word or not. In this work, all the AV scores are calculated from Chinese Gigaword second edition in advance. The scores of all the character sequences in the feature template (Table 3) are adopted as features.

2.6 Experiments

2.6.1 Experiment Setting

For investigating the performance of our synthetic word parser [5], we perform a 10-fold cross-validation on the whole 31,849 synthetic words (obtained in Section 2.4). Since the total 31,849 words are not repeated, the cross-validation results can be seen as the analysis performance of our synthetic word parser on the OOV words.

Intuitively, a pipeline method with word segmentation and parsing is a reasonable baseline for comparison to evaluate the performance of our system. For the pipeline method, we implement a word segmenter (Section 3.3.2) and a word level MST parser with the default features [27, 28] to complete this task as the baseline.

For evaluating the performance of our synthetic word parser, the evaluation metric of CoNLL 2006 shared task[6] is adopted in character-level, which includes unlabeled attachment score (UAS), unlabeled complete match (UCM), labeled attachment score (LAS) and labeled complete match (LCM).

2.6.2 Main Results

In Table 4, we present the main performance of our synthetic word parser. 'Baseline' denotes the pipeline method (Section 2.6.1). 'MST-2' denotes our graph-based parser with only first-order features (the local and long features in Table 3). 'MST' denotes our graph-based parser including the second-order features. 'Dict', 'Brown' and 'AV' denote those feature types (introduced in Section 2.5.3) separately incorporated into the parser. 'all' denotes that all features are used.

The results of 'MST' and 'MST-2' clearly suggest that our character-based dependency framework significantly outperforms the pipeline method 'Baseline' without relying extra resources. Second order sibling factorization brings a 0.58% improvement on LAS. 'MST-2' significantly improves the parsing performance by around 4.5% on LAS and 12.2% on LCM, compared to the 'baseline'. The

[5]The synthetic word parser described in Section 2.5 and the synthetic word dictionary described in Section 2.4 are released in `https://github.com/racerandom/chcomparser`

[6]http://ilk.uvt.nl/conll/

	UAS	UCM	LAS	LCM
Baseline	93.65	80.19	90.32	71.06
MST	97.96	93.01	94.3	81.72
MST-2	98.07	93.26	94.88	83.31
MST-2 + Dict	98.15	93.58	96.34	87.64
MST-2 + Brown	98.17	93.56	96.41	87.73
MST-2 + AV	98.2	93.75	96.27	86.82
MST-2 + all	98.25	93.87	96.66	89.09

Table 4. The main parsing results of our synthetic word parser.

observation of this improvement can attribute to our well-designed character-based dependency framework.

We further boost the parsing performance by separately incorporating different feature types. 'Dict' and 'Brown' are two word-level features carrying both word boundary and semantic meaning information of words. Both 'Dict' and 'Brown' are proved to be helpful to boost the parsing performance with improvements around 1.5% on LAS, compared to 'MST-2'. As a statistic type of feature, 'AV' provides distinctive evidences of word boundaries, which shows an obvious overall improvement on F-score from 94.88 to 96.27, especially the highest UCM 93.75% with each single feature. The final combination of all three feature types further achieves the best parsing performance with improvement around 1.8% on LAS and 5.8% on LCM, compared to the 'MST-2'. The extra resources for feature extraction conduct main efforts in these improvements, based on the same character-based parsing framework.

2.6.3 Additional Results

Although in Table 4, our best parsing model reaches reasonable performance with 96.66% LAS and 89% LCM, we are concerned about that three-character words take a high proportion around 68.6% of the whole 31,849 words (Table 9). An overall evaluation is far from enough to make a detailed observation of the real performance of our system. The original motivation of synthetic word parsing is to analyzing the internal structure of the words in the existing word segmentation

corpora. An import view is to evaluate the parsing performance on the words with different character lengths. Figure 9 presents the LAS performance of our best system ('MST2-all') against different character lengths of the words. As we expected, our parser reaches a very high LAS score 98.63% on 3-character words. As the word length increases, the LAS performance starts to drop. A special case is that for 6-character words, our system obtains even a slightly higher LAS 95.85% than 95.82% on 5-character words. For the words with 8 characters or more, our system obtains 94.34% LAS, which has dropped by 4.2% compared to 3-character words.

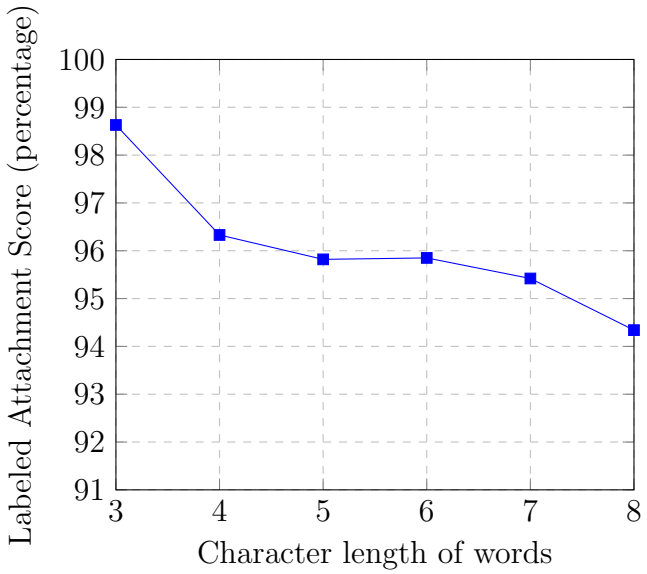

Figure 9. Labeled attachment score against character lengths of words. The character length equal to 8 means greater or equal to 8.

The current state-of-the-art word segmentation systems reach 96 to 97 F-score on Bakeoff-2015 data. For the purpose of enhancing the word segmentation performance, our system is also expected to perform well on labeled complete match (LCM). Table 10 shows the LCM performance of our system against character lengths of words. For 3-character words, the system achieves an ideal result 96.4%. As the word length increases, the LCM drops quickly. The LCM on the words with 8 characters or more has dropped to 64.93%. It is hard to determine whether the performance is enough or not, because the word length distributions

of different CWS corpora are various. We will discuss the parsing performance in two different CWS corpora in the next Chapter.

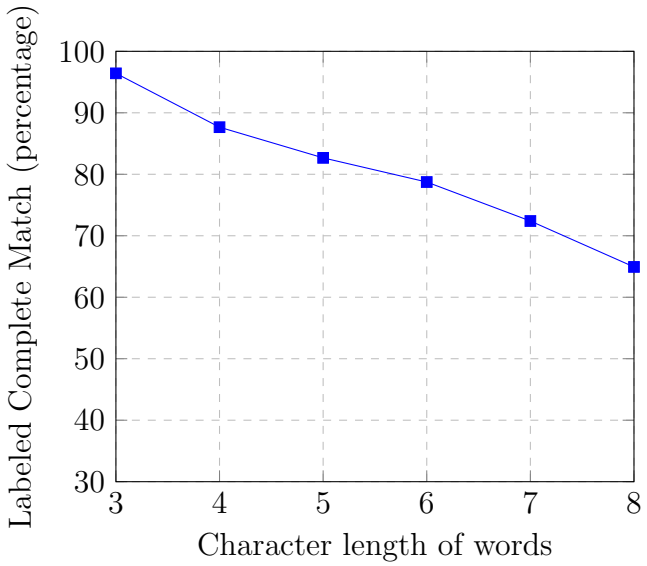

Figure 10. Labeled complete match against character lengths of words. The character length equal to 8 means greater or equal to 8.

2.7 Summary

In this paper, we claim that synthetic word parsing is an important but overlooked problem in Chinese NLP. Our first contribution is the annotation of 31,849 Chinese synthetic words, which is potentially useful to other Chinese NLP tasks. The data is distributed as free available data[7]. In the second step, we propose a well designed character-based dependency parsing framework, which significant outperforms the traditional pipeline method. Furthermore, we highly boost the performance of our synthetic word parser by extracting features from a large-scale unlabeled corpora and a dictionary. We believe that this is a first-step toward a more robust character-based processing of Chinese that does not require explicit word segmentation.

[7]https://github.com/racerandom/chcomparser

Chapter 3

3. Enhancing Chinese Word Segmentation with Internal Structure Information

In this chapter, we illustrate how to enhance the Chinese word segmentation performance by using internal structure information of words. We first introduce the background (Section 3.1) and related work (Section 3.2) about Chinese word segmentation. In Section 3.3, we present our enhanced word segmentation system with two steps of processes: 1) Conversion of the word segmentation training data to a fine-grained level (Section 3.3.1) by our synthetic word parser, which is presented in Chapter 2. 2) Using a word segmenter to predict a combined position label of two segmentation levels. (Section 3.3.2). The results of the main experiments of our word segmentation system are shown in Section 3.4.2. The summary is made at last.

3.1 Introduction

Since Chinese has no spaces between words to indicate word boundaries, word segmentation is a task to determine word delimiters in Chinese sentences. In recent years, Chinese word segmentation has progressed significantly, with the state-of-the-art performance around 96 to 97 F-score on the the Second International Chinese Word Segmentation Bakeoff (Bakeoff-2005) data. However, issues still remain and we summarize two as follows.

- **1. The variety of word segmentation standards**

  Due to the difficulty of defining 'word' in Chinese, the resources are annotated in compliance with the different specified 'rules' of the providers. In Bakeoff-2005, Peking university supplied a very specific segmentation guideline with 19 pages for Peking University (PKU) corpus . For instance, one clause in the guideline indicates that most of the community, institution and

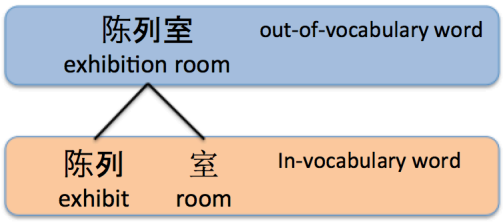

Figure 11. An example of a OOV word with all the internal parts as in-vocabulary words.

organization names are synthetic words, which are to be segmented. Therefore, the word 北京国安队 is segmented into 北京 (Beijing) / 国安 (Guoan) / 队 (club) in PKU. On the other hand, according to the guideline of the Microsoft Research (MSR) corpus, a Named Entity like 北京国安队 is to be treated as a single word. The ambiguity on word segmentation standards not only makes it hard to share annotated resources among different research groups, but also has an adverse impact on other downstream Chinese NLP tasks.

- **2. Low recall of Out-of-vocabulary (OOV) words**

  Frequent OOVs are another crucial issue that causes low accuracy in word segmentation. Li and Zhou [19] defined those words that are OOVs but consisting of frequent internal parts (In-vocabulary words, called IV words) as pseudo-OOVs and estimated that over 60% of OOVs are pseudo-OOVs in five common Chinese corpora. For instance (Figure 11), PKU corpus does not contain the word 陈列室 (exhibition room), even though the word 陈列 (exhibit) and 室 (room) appear hundreds of times.

Goh et al. [14] also claimed that most OOVs are proper nouns taking the form of Chinese synthetic words. These previous works suggest that analyzing the internal structure of the synthetic words brings two improvements on word segmentation: 1) The segmentation standard is more consistent. For instance, two words 中国国 际广播 电台 and 中央/广播/电台 are inconsistently annotated in MSR, because 中国国 际广播 电台 is treated as a Named Entity. A fine-grained

中国/国 际/广播/电台 structure obvious makes the corpus more consistent inside. 2) Parts of pseudo-OOVs are converted into in-vocabulary words (IVs). By running a synthetic word parser on each of the words in a existing word segmentation training data, we can generate a fine-grained segmentation standard which is more consistent inside and with lower OOVs rate. Since the current Conditional Random Fields (CRFs) word segmenters [39, 35] perform well on IVs, this converting process can conceivably improve the handling of pseudo-OOV words.

In this chapter, we propose a pipeline word segmentation system to address the segmentation standard and OOVs issues. Our system first converts the original training data to a fine-grained segmentation level by parsing the words with a synthetic word parser. Each original character-level position label is combined with the label derived from the new fine-grained standard. Then a CRF-based word segmenter is trained on the training data with new combined labels. Our system is evaluated on the Bakeoff-2005 data to verify the usefulness which the internal structure information of Chinese words brings.

3.2 Related Work

Xue et al. [42] first proposed this method which treated Chinese word segmentation as a character-based sequential labeling problem and exploite several discriminative learning algorithms. Tseng et al. [39] adopted the CRFs model as the learning method and obtained the best results in Bakeoff-2005. One research line is to find effective feature types. Sun and Xu [35] attempted to extract statistical information from large unlabeled data to enhance the CWS performance. Recently, Liu et al. [23] introduced character representations as a new cluster-based feature for the CRF-based segmenter. These feature types successfully improved current CWS performance.

Recently, studies that explored the use of the internal structures of words to improve Chinese processing have shown promising results. Sun et al. [37] presented a joint model for Chinese word segmentation and OOVs detection. Their models achieved fast training speed, high accuracies and increase on OOV recall. Sun [33] proposed a similar sub-word structure to our work, which is generated by merging the segmentations provided by different segmenters (a word-based segmenter, a character-based segmenter and a local character classifier).

However, their model does not actually analyze the sub-words structure of all the synthetic words in a corpus, but only those words with different segmented results of the three segmenters. Her work maximizes the agreement of different models to improve CWS performance. Different from their work, we aim to provide a simple and unified way to incorporate internal morphological information of the synthetic words into the CWS task or other Chinese NLP tasks.

Li and Zhou [19] claimed the importance of word structures. They proposed a new parsing paradigm, in which the internal structures of words are identified. Zhang et al. [46] manually annotated the internal structures of 37,382 words, which covers the entire Chinese TreeBank 5 (CTB5). Then, they constructed a shift-reduce parser with customized actions to jointly perform word segmentation, part-of-speech tagging, and parsing. Their system significantly outperformed current pipeline methods. However, these studies relied on prior knowledge of internal structure information, which is manually annotated. In this work, we employ an automatic parsing mechanism to analyze the internal structures of words to improve word segmentation performance.

Some other researches focus on adapting different segmentation standards of multiple corpora. Jiang et al. [15] presented a simple strategy to train a source classifier a source corpus, which is used to label the target corpus and results in a “source-style” annotation of the target corpus. Then the final segmenter was trained on the target corpus with the “source-style” prediction as additional features. Their method was similar to some ideas in domain adaptation [11, 10]. Unlike the 'source-style' prediction, our work intends to create a fine-grained segmentation prediction on the original corpus by a synthetic word parser, which is expected to be more consistent inside and with lower OOVs rate. Our work doesn't rely on another annotated corpus. The fine-grained level prediction attributes to the internal structure information inside words.

3.3 Word Segmentation System Enhanced by Internal structure of Word

The framework of our word segmentation system includes two key components as shown in Figure 12.

- **Synthetic word parser** is used to analyze the internal tree structure of the words in the original word segmentation training data. Then we convert training data to a fine-grained segmentation level according to the flat segmentation representations of the internal trees of the words.
- **CRF-based word segmenter** is adopted to predict a character-level combined position label of the original annotation and new fine-grained standard.

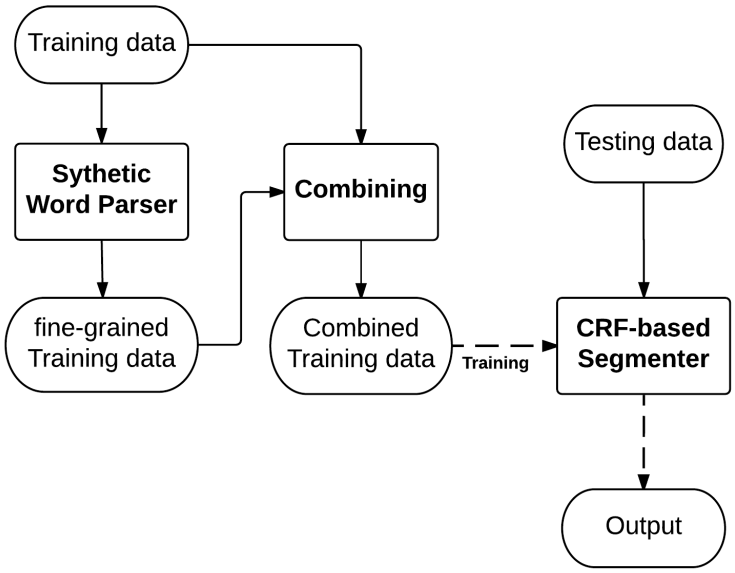

Figure 12. The framework of our word segmentation system.

3.3.1 Conversion to Fine-grained Segmentation Level

Intuitively, the internal structure information inside words is helpful to improve word segmentation performance. In Chapter 2, we introduced our character-based synthetic word parser, which achieves high performance by incorporating different types of features extracted from a dictionary and a large-scale unlabeled corpus. We use our synthetic word parser to parse the internal tree structure of the words in the training data of Bakeoff-2005 (PKU and MSR). Each word in

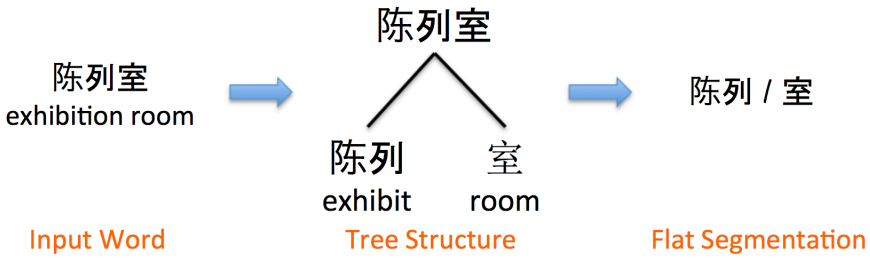

Figure 13. The tree structure and flat segmentation of a sample word.

Original Standard	国家大剧院 / 将 / 作为 / 历史性 / 建筑 / 被 / 保留 / 下来 national grand theater / will / as / historic / building / be / preserved / down
Internal trees	国家大剧院 历史 大剧院 大 剧院 历史性 历史 性
Fine-grained Level	国家 / 大 / 剧院 / 将 / 作为 / 历史 / 性 / 建筑 / 被 / 保留 / 下来 National / grand / theater / will / as / historic / nature / building / be / preserved / down

Table 5. The conversion of an example sentence from the original standard to the new fine-grained level.

the original training data is converted to a flat segmentation result of its parse tree, as shown in Figure 13.

For native Chinese speakers, single character and 2-character words are usually treated as the basic units. In this step, all the words longer than 2-character in the original training data are parsed by the synthetic parser. With the flat segmentation information inside the words, the original training data are converted into a fine-grained level as Table 5.

3.3.2 Word Segmenter for a Combination of Two Segmentation Labels

The Character-based labeling method is a dominate approach for Chinese word segmentation. A 4-label set $\{B, I, E, S\}$ is widely used to represent the position information $(begin, intermedia, end, single)$ of a character inside a word. Table 6 shows an example sentence 在激昂 奋进的音 乐声中辞旧迎新 (ring the old year out and welcome the new year in this passionate music) labeled with the 4-label

Characters	在	激	昂	奋	进	的	音	乐	声	中	辞	旧	迎	新
Labels	S	B	E	B	E	S	B	I	E	S	B	I	I	E
Segmentation	在/	激昂/	奋进/	的/	音乐声/	中/	辞旧迎新							

Table 6. An example sentence labeled with the $\{B, I, E, S\}$ set.

set.

In this work, the fine-grained segmentation provided by the previous step brings new segmentation labels for the characters in the training data. We make a simple combination of the original label and new fine-grained label for each character, as shown in Table 7. We extend the 4-label set into a combined label set as:

$$\textbf{\textit{Combined label = Original label + Fine-grained label}}. \tag{5}$$

Conditional random fields (CRFs) are a class of statistical sequence modeling framework first introduced into language processing in Lafferty et al. [17]. The probability model and feature function is defined given a set $H \times T$, where H is a set of context features predefined and T is a possible tag in the tag set. The feature function is defined as follows,

$$f(h,t) = \begin{cases} 1, \text{ if } h = h_i \text{ and } t = t_j \\ 0, \text{ otherwise} \end{cases}, \text{ where } h_i \in H \text{ and } t_j \in T \tag{6}$$

CRFs perform well in the Chinese word segmentation task and many researches adopt CRFs as the baseline segmenter. For predicting the new combined position label of each character, we adopt a CRF-based model as the baseline segmenter. Unlike common feature selection work on word segmentation (different features are separately added into the baseline segmenter to observe the changes of the system performance), our segmenter includes all the features as shown in the next subsection to perform an idea baseline result to observe, whether the new combined segmentation labels can further enhance the baseline to state-of-the-art performance.

3.3.3 Feature Types for the CRF-based Word Segmenter

To demonstrator the features easily, we denote a current character to be labeled c_i with a context $[...c_{i-2}c_{i-1}c_ic_{i+1}c_{i+2}...]$. $c_{[s:e]}$ expresses a character sequence that starts from c_s and ends at c_e.

- **Character Context**
  - Character uni-gram: $c_s \quad (i-3 < s < i+3)$
  - Character bi-gram: $c_sc_{s+1} \quad (i-3 < s < i+2)$
  - Whether c_s and c_{s+1} are identical, for $(i-2 < s < i+2)$
  - Whether c_s and c_{s+2} are identical, for $(i-4 < s < i+2)$
- **Dictionary**
  - The identity of the character sequence $c_{[s:i]} \quad (i-5 < s < i)$, if it matches a word in Naist Chinese dictionary.
  - The identity of the character sequence $c_{[i:e]} \quad (i < e < i+5)$, if it matches a word in Naist Chinese dictionary.
- **Accessor Variety** The definition for calculating AV score is as same as in Section 2.5.3. The accessor variety features are included in our model as follows,
  - The AV score of the character sequence $c_{[s:i]} \quad (i-5 < s < i)$
  - The AV score of the character sequence $c_{[i:e]} \quad (i < e < i+5)$
- **Vector-based Character Sequence Representation** focuses on embedding a word or character n-gram as a low-dimensional real-valued vector. The goal is to place similar character n-grams into nearby points in the vector space. Liu et al. [23] introduced vector-based character representations as a new type of feature. Given a sentence $[c_0, c_1, c_2, c_3, c_4, ...]$, they generate its Bi-gram chunks $[c_0c_1, c_1c_2, c_2c_3, c_3c_4, c_4c_5...]$ and tri-gram chunks $[c_0c_1c_2, c_1c_2c_3, c_2c_3c_4, c_3c_4c_5, c_4c_5c_6...]$. Next, they learn the vector-based representations of all the Uni-gram, Bi-gram, Tri-gram character sequences

Original Standard	国家大剧院 / 将 / 作为 / 历史性 / 建筑 / 被 / 保留 / 下来 national grand theater / will / as / historic / building / be / preserved / down
Original Labels	*B I I I E S B E B I E B E S B E B E*
Fine-grained Level	国家 / 大 / 剧院 / 将 / 作为 / 历史 / 性 / 建筑 / 被 / 保留 / 下来 National / grand / theater / will / as / historic / nature / building / be / preserved / down
Fine-grained Labels	*B E S B E S B E B E S B E S B E B E*
Combined Labels	***B-B I-E I-S I-B E-E S-S B-B E-E B-B I-E E-S B-B E-E S-S B-B E-E B-B E-E***

Table 7. An example of the way to obtain the new combined segmentation labels of a sentence in the training data.

from a large-unlabeled corpus. Finally, their cluster-based features come from the K-Mean method applied on these vector-based representations. In this work, we do a similar process and use the following cluster-based features:

- Uni-gram clusters: $c_s \quad (i-3 < s < i+3)$
- Bi-gram clusters: $c_s c_{s+1} \quad (i-3 < s < i+2)$
- Tri-gram clusters: $c_s c_{s+1} c_{s+2} \quad (i-4 < s < i+2)$

3.4 Experiments

3.4.1 Experiment Setting

In Chapter 2, we constructed a 31,849 synthetic word dictionary with internal structure annotated. However, a number of transliteration words (e.g. 奥林匹克 Olympic) exist in the Chinese corpora, our synthetic word parser should perform well not only on synthetic words but also on transliteration words. We further extract 6,574 transliteration words from the NAIST Chinese Dictionary and automatically assign the flat structure (each modifier character points to its successor head character from left to right) for these words. As a result, we obtain 38,423 words as the training data for our parser.

CRFSuite[8] is a speed oriented implementation of CRFs for labeling sequential data. We incorporate the dictionary (NAIST Chinese Dictionary) and access

[8]http://www.chokkan.org/software/crfsuite/

variable feature (Chinese Gigaword Second Edition) in the same manner described in (Sun and Xu [35]). Based on (Liu et al. [23]), we use word2vec[9] to train the vector representations of the unigram, bigram and trigram character sequences of Chinese Gigaword Second Edition. The cluster-based results with K=100 are treated as the features for the segmenter. The segmenter also provides the baseline results for comparison.

The second international Chinese word segmentation Bakeoff-2005 provides two annotated simplified Chinese corpora: PKU and MSR. We conduct all the word segmentation experiments on these two corpora.

In this paper, OOVs are defined as words not seen in the training set; thus, even if a word is in the NAIST dictionary, it could be OOV with respect to the training set. In PKU, there are 2404 OOVs and among them 1097 are seen in the dictionary. In MSR, there are 1960 OOVs and 259 are seen in the dictionary.

3.4.2 Word Segmentation Results

Table 8 summarizes the word segmentation results on PKU and MSR corpora. "Bakeoff-2005" denotes the best results of the second international Chinese word segmentation bakeoff-2005 on two corpora. Since we use extra resources and our proposed method replies on the synthetic word parser trained on an dictionary with internal structure annotated, the results might not be directly compared with the state-of-the-art systems. For comparison, we give a baseline result by training a CRF word segmenter on the original PKU and MSR data sets. Although the baseline achieves very ideal performance by inluding the state-of-the-art features, our proposed system is expected to improve the word segmentation performance, especially on OOVs recall without relying on any new feature types.

Compared to the baseline, the proposed method improves 0.2 points F-score on both PKU and MSR corpora. Especially, the proposed system obtains the improvements of OOV recall from 81.9 to 83.6 on PKU and 72.5 to 74 on MSR. Our proposed system achieves the highest F-score with 0.962 on PKU and 0.974 on MSR. Our proposed method obtains a overall higher performance, compared to the results of Zhang et al [45] who extracted dynamic statistical features from both in-domain and out-domain corpus. The OOV recall of our system significantly

[9]https://code.google.com/archive/p/word2vec/

System	PKU				MSR			
	P	**R**	**F**	**R_{oov}**	**P**	**R**	**F**	**R_{oov}**
Baseline	96.3	95.7	96.0	81.9	97.1	97.4	97.2	72.5
Proposed method	**96.5**	**95.9**	**96.2**	**83.6**	**97.3**	**97.5**	**97.4**	**74.0**
Zhang 2013	**96.5**	95.8	96.1	73.1	-	-	**97.4**	-
Sun 2009	95.6	94.8	95.2	77.8	**97.3**	97.3	97.3	72.2
Bakeoff-2005	95.3	94.6	95.0	63.6	96.2	96.6	96.4	71.7

Table 8. Comparison of the Proposed Method to the Baseline and Previous works on PKU and MSR Corpora.

Corpus	**3-char**		**4-char**		**5-char**		**6-char**		**longer**	
	Count	**Rate**	**Count**	**Rate**	**Count**	**Rate**	**Count**	**Rate**	**Count**	**Rate**
PKU	11,320	20.5%	6,812	12.3%	1,746	3.2%	611	1.1%	887	1.6%
MSR	17,081	19.4%	12,545	14.2%	6,879	7.8%	5,103	5.8%	10,195	11.6%

Table 9. The character length distribution of the words with three characters or more in PKU and MSR corpora. 'Count' denotes the number of the word types with a specific character length. 'Rate' denote the number of the word types with a specific character length against the total word types in a corpus.

outperforms theirs with a 10.9 points lead. In MSR, we obtain higher OOV recall and slightly higher F-score than the state-of-the-art system [38], which adopted a latent variable CRF model. Our system also outperforms their system in PKU. In both corpora, our proposed system outperforms the best "Bakeoff-2005" results.

3.4.3 Additional Experiments

In Table 9, we investigate the character length distribution of the words with three characters or more in PKU, MSR. MSR is obviously a coarse standard data compared to PKU with higher rates in almost all the character lengths. Especially, the words with 7-character or more account for around 11.6% of all the word types in MSR, while the rates are only 1.6% in PKU.

We are interested in how much parsing accuracy is needed for improving the word segmentation step. We conduct a 10-fold cross validation on the whole 38,423 words with the best parsing model in Table 4. Instead of using LAS, we use the more restrict metric LCM to observe the parsing performance against the

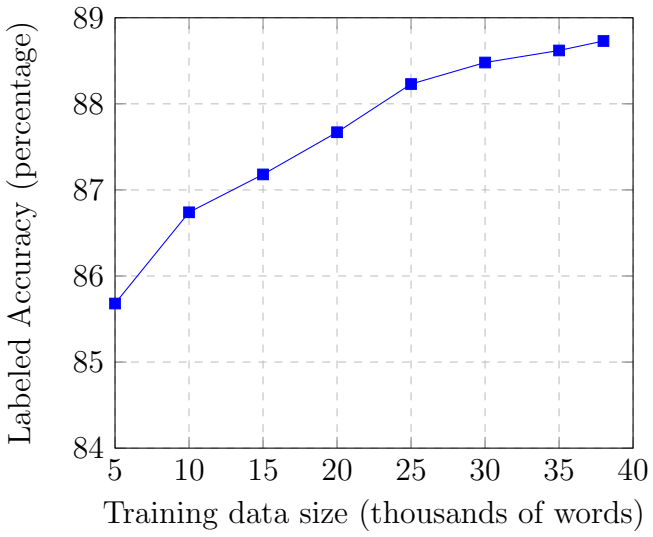

Figure 14. Labeled Complete Match parsing performance against the training data size.

training data size in Figure 14. As we mentioned, 3-character words still take a high proportion of the whole 38K words. The system reaches 85.68% LCM with only 5k training data. The reason is that the LCM of 3-character words is much higher and the features also performance well. As the size of the training data increases, the overall LCM increases quickly. The finally parsing performance reaches 88.73% with the total 38K words for training. we can expect that the 3% gain of LCM mainly comes from the improvement of the long words parsing performance. Considering the character length distribution in Table 9, we believe that the overall 88.73 LCM is a reasonable parsing performance.

We also evaluate the LCM performance on the different word lengths of the whole 38,423 words in Figure 15. The results are similar to the results on 31,849 synthetic words Figure 10. For 3-character words, the system achieves a reasonable result 96.29%. As the word length increases, the LCM drops quickly. The LCM of the words with 8 characters or more has dropped to 64.91%. Considering that the words with 5 character or more barely exist in PKU (Table 9), our parser perform really well on 3-character or 4-character synthetic words. On the other hand, MSR is a coarse level corpus with 11.6% words longer than 6 characters. However, the parsing performance is not the only factor to influence the overall

segmentation result. The fine-grained level brings two improvements: more consistent inside a corpus and lower OOVs. As we mentioned in Section 2.6.1, the results of the cross-validation on the unduplicated 38,423 words can be seen as the parsing performance on OOVs. Therefore, OOV rate is another key element to connect the parsing performance to the final word segmentation performance. Our model still achieves 0.2% improvement to 97.4 segmentation F-score on MSR in Table 8. One clue is that the OOV rate of MSR is 2.6%, which is lower than 5.8% of PKU. Another suggestion is that more inconsistencies exist in MSR.

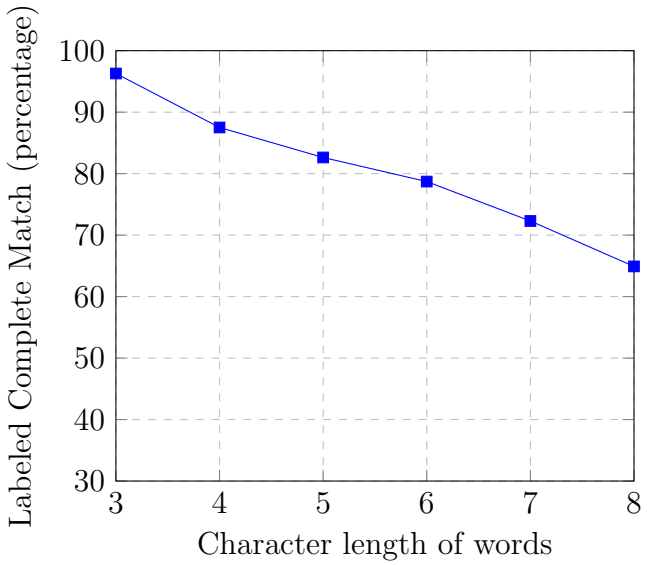

Figure 15. Labeled complete match against character lengths of words on the whole 38,423 words. The character length equal to 8 means greater or equal to 8.

Figures 17, 16 display the OOV recall results of our word segmentation system when the synthetic word parser is trained with different size of training data. As the data size increases, our word segmentation system obtains consistent gains of OOV recall on both corpora. On the whole 38K words training data, our system reaches the highest OOV recall. An interesting observation is that the OOV recall on MSR is more sensitive on data size increasing. The main reason is the different annotation standard of the two corpora. PKU is a correspondingly fine-grained annotated corpus with shorter average word length than MSR (Table 9).

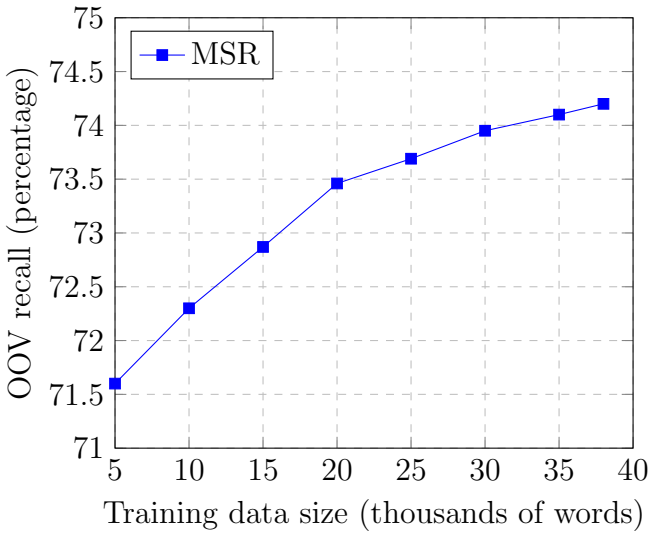

Figure 16. The OOV Recall Evaluation against the training data size on MSR

Our synthetic word parser reaches high parsing accuracy on short length words (three-character and four-character words) even with a small training data size. With the increase of word length, the parser needs more training data. These factors cause that our system reaches high OOV recall on PKU starting from a small training data size and obtains more OOV recall gains on MSR when increasing the training data size.

3.4.4 Analysis

As we expected, the proposed method obtains overall improvement, especially on OOV recall. In both corpora, we observed a number of OOVs are segmented correctly. For instance, 管理法 (management law) is an OOV word in PKU corpus. In this word, 管理 (management) appears frequently and 法 (law) is a common suffix in Chinese synthetic words, such as 行政法 (administrative law) or 国际法 (international law). This type of pseudo-OOVs share a major contribution to improvement the system performance.

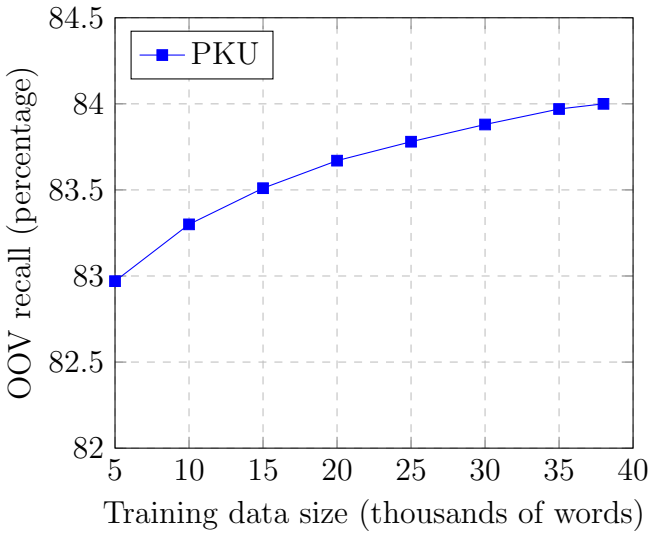

Figure 17. The OOV Recall Evaluation against the training data size on PKU

3.5 Summary

In this chapter, we present a simple yet two-stage word segmentation system. Our system first do a fine-grained conversion of the segmentation standard on two common word segmentation corpora by a synthetic word parser (in Chapter 2). The conversion makes the corpora more consistent inside and reduces OOVs. In the second step, our CRF-based segmenter is enhanced by predicting combined segmentation labels with the original and new fine-grained level information, without relying on any new feature types. Our proposed method achieves state-of-the-art F-score and OOV recall on two common corpus PKU and MSR. However, note that we only exploit the flat segmentation of internal word structure here. As future work, we plan to exploit the full tree structure of synthetic words to improve not only word segmentation but also additional down-streaming tasks, such as sentence parsing.

Chapter 4

4. Extending Training Data in a Consistent Segmentation Level across Multiple Corpora

In this chapter, we propose a pipeline word segmentation system that adapts two different corpora, i.e., PKU and MSR into one consistent segmentation level and improves segmentation performance on each individual corpus. The proposed model is based on two basic components: synthetic word parser (Section 2) and CRF-base word segmenter (Section 3.3.3). In Section 4.2.1, we explain how the proposed method maps two different word segmentation corpora to a consistent segmentation level (to extend data). We further explain how we achieve finer-grained segmentation (to reduce OOVs) using a synthetic word parser (Section 4.2.2), and how we transform the segments back to the PKU and MSR standards (Section 4.2.3). In Section 4.3.3, we compare the final segmentation results obtained by the proposed system to a baseline and state-of-the-art systems.

4.1 Introduction

Due to the lack of a common word segmentation standard, the Chinese word segmentation corpora accomplished by different research groups can hardly share different corpora in combination. One solution is to find a suitable segmentation level that is consistent across multiple corpora and where a part of OOVs are naturally segmented into IV words. Here, segmentation level is defined as any middle segmentation standard between the most fine-grained (character) and most coarse-grained standard (original corpora). A consistent level is expected to be not only more consistent across different corpora, but also more consistent inside each individual corpus. For instance, as shown in Figure 18, two words 石油天然气集 团公司 (Oil and Gas Corporation) and 天然气 (natural gas) inconsistently exist in Standard 1, because 石油天然气集 团公司 is a named entity that is treated as a single word. In the consistent level, 石油天然气集 团公司 takes a more natural standard 石油 / 天然气 / 集 团公司 close to 天然气.

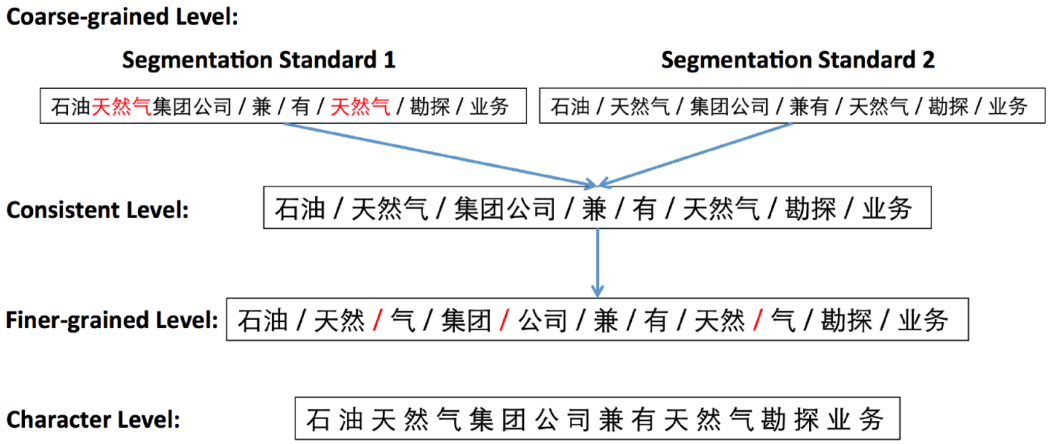

Figure 18. An example of a sentence in several different segmentation levels.

Even in the consistent level, many synthetic words, such as 天然气 and 集团公司, still exist. A synthetic word parser is used to analyze the internal structure of 天然气 (a possible OOV word) and generate the flat sub-word segmentation 天然 / 气 (natural / gas). Our goal is to create a strategy to find a consistent level automatically and extend training data using heterogeneous data. Then, the internal word structure information helps convert the extended data to a finer-grained level (to reduce OOVs).

In this chapter, we further propose a simple strategy to transform two different Chinese word segmentation (CWS) corpora into a new consistent segmentation level, which enables easy extension of the training data size. The extended data is verified to be highly consistent by 10-fold cross-validation. In addition, we use a synthetic word parser to analyze the internal structure information of the words in the extended training data to convert the data into a more fine-grained standard. Then we use two-stage Conditional Random Fields (CRFs) to perform fine-grained segmentation and chunk the segments back to the original PKU or MSR standard. The extension of the training data and reduction of the OOV rate in the new fine-grained level significantly improve the segmentation performance of the recall and F-score on the PKU and MSR corpora.

This method involves the research topic about using heterogeneous data to improve word segmentation performance. Jiang et al. [15] presented a simple strategy to train a source segmenter to segment the target corpus. Then the proposed segmenter is trained on the target corpus including 'source-style' predictions as guide-features. Their method is similar to the ideas in domain adaptation [11, 10]. Chao et al. [6] proposed a coupled Conditional Random Fields model to exploit multiple heterogeneous data to improve segmentation performance on Weibo data. Although our proposed pipeline method contains a similar "source-style" prediction step, the following strategy to find a consistent segmentation level differs from simply treating predictions as guide-features in their work. Our work aims to do a more detailed investigation on the conversions between different segmentation levels. In addition, our idea of a consistent segmentation level is friendly to introduce synthetic words parsing to boost the segmentation performance further.

4.2 Word Segmentation System Involving Different Segmentation Levels

4.2.1 Consistent Segmentation Level for PKU and MSR Corpora

In the proposed model, the first step is to find a consistent segmentation level for multiple CWS corpora. In Figure 19, the raw sentences in the training data of the MSR corpus (MSR train) are segmented by a segmenter trained on PKU. We refer to the output as PKU-level MSR train. Our strategy finds a new segmentation level by including word boundaries that appear in either the original MSR annotation or the PKU-level MSR train (Figure 20). The same process is performed on the PKU side. The new segmentation level maximizes the number of word boundaries based on the annotation standards of the original corpus and the other corpus, which is expected to be fine-grained compared to the two original standards. We hypothesize that the new segmentation level MSR and PKU training data are consistent and can be easily combined into a larger training dataset.

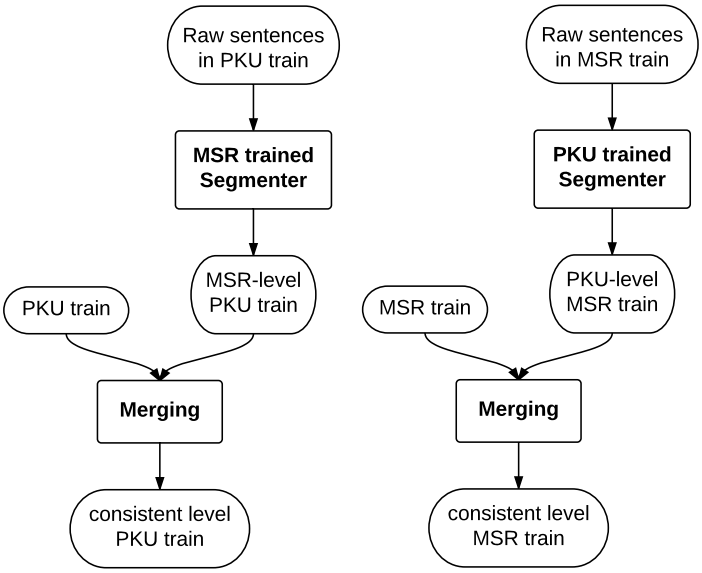

Figure 19. Workflow of the proposed method to find a consistent segmentation level of multiple CWS corpora. 'PKU train' denotes the original annotated PKU training data and 'MSR train' denotes the original annotated MSR training data.

Raw sentence	成 都 铁 路 局 开 行 行 包 专 列
Original MSR	成 都 铁 路 局 / 开 / 行 / 行 / 包 / 专 列
PKU-level MSR	成 都 / 铁 路 局 / 开 行 / 行 包 / 专 列
Consistent Level	成 都 / 铁 路 局 / 开 / 行 / 行 / 包 / 专 列

Figure 20. Example of the strategy to find a new consistent segmentation level.

Consistent Level	成 都 / 铁 路 局 / 开 / 行 / 行 / 包 / 专 列
	Chengdu / railway bureau / start / drive / luggage / package / special train
Internal tree	铁路 局
	railway bureau
Fine-grained	成 都 / 铁 路 / 局 / 开 / 行 / 行 / 包 / 专 列

Figure 21. Example of a sentence with the consistent segmentation level converted to a finer-grained level.

4.2.2 Finer-grained Conversion using Synthetic Word Parser

As mentioned in Section 3, we demonstrated that a fine-grained segmentation level improves word segmentation performance due to the morphological infor-

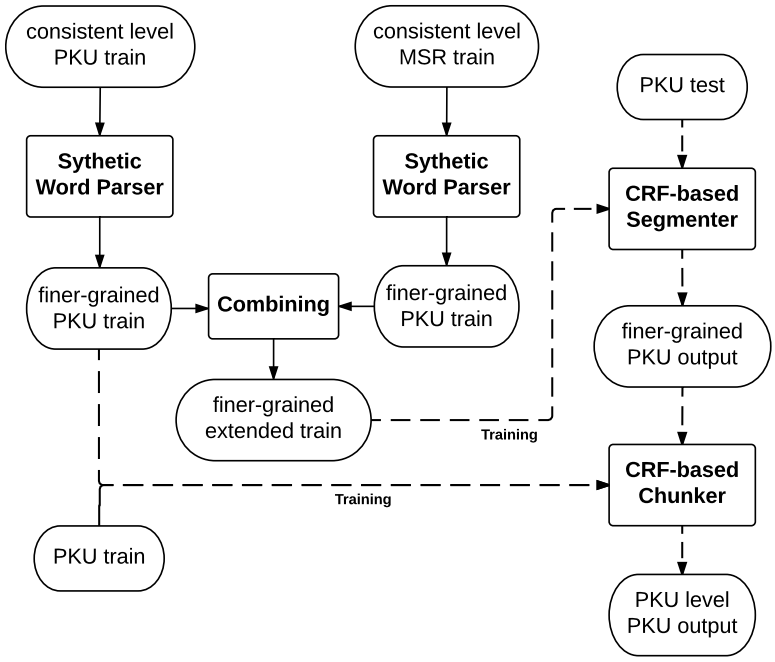

Figure 22. Workflow of the two-stage word segmenter.

mation and low OOV rate. Intuitively, the consistent segmentation level data can be further converted into a finer-grained level using a synthetic word parser. In this work, we train a graph-based parsing model on 38,432 synthetic words with annotated internal structures to perform the conversion.

In this work, the words longer than two characters in the new consistent level data provided by the previous stage are parsed by the synthetic word parser. With the flat sub-word segmentation of each word, the consistent level data are converted to a finer-grained level (Figure 21).

4.2.3 Finer-grained Word Segmentation and Chunking to Original Segmentation Level

After obtaining finer-grained PKU and MSR data, we combine the data into a larger training dataset. Then, the first-stage CRF-based segmenter predicts the fine-grained output of the test data. The second-stage CRF-based chunker is used to recover the fine-grained output to the original segmentation level. The workflow of this step is shown in Figure 22.

Fine-grained Level | 歌剧 / 院 / 合唱 / 团 / 共 / 200 / 人 / 。
Chunking Tags | B / E / B / E / S / S / S / S
Original Level | 歌剧院 / 合唱团 / 共 / 200 / 人 / 。

Figure 23. Chunking tags of an example sentence.

	Precision	**Recall**	**F-score**	$\mathbf{R_{oov}}$
10-fold cross-validation	98.25	98.03	98.14	70.21

Table 10. 10-fold cross-validation results on new extended data.

The training data of the chunker is constructed on the fine-grained and original standard training data. An example of the chunking tags of a sentence is shown in Figure 23. In this sentence, two words 歌剧 (opera) and 院 (house) are chunked to the synthetic word 歌剧院 (opera house). 合唱 (chorus) and 团 (group) are chunked to the word 合唱团 (chorus group). Note that the same features (Section 3.3.3) of the first-stage segmenter are used in the second-stage chunker and the basic units of the features for the chunker are words rather than characters.

4.3 Experiments

4.3.1 Settings

The proposed methods contains the same two components (synthetic word parser and CRF-based word segmenter) as the described in Section 3.4.1. In this section, we adopt the previous settings to construct the exact same synthetic word parser and CRF-based word segmenter.

The basic CRF-based word segmenter also provide a baseline to investigate the improvement brought by the extension of training data and internal structure information.

4.3.2 Consistency of New Segmentation Level

An important hypothesis is the process described in Section 4.2.1 on two different corpora may reach a new consistent segmentation level. To prove this, we first

Corpus	3-char		4-char		5-char		6-char		longer	
	Count	**Rate**	**Count**	**Rate**	**Count**	**Rate**	**Count**	**Rate**	**Count**	**Rate**
PKU	11,320	20.5%	6,812	12.3%	1,746	3.2%	611	1.1%	887	1.6%
MSR	17,081	19.4%	12,545	14.2%	6,879	7.8%	5,103	5.8%	10,195	11.6%
Extended	13,706	17.90%	9,053	11.8%	3,689	4.8%	1,465	1.9%	1,536	2%

Table 11. Character length distribution of words in PKU and MSR corpora. 'Count' denotes the number of the word types with specific character length; 'Rate' denotes the number of word types with specific character length against the total word types in the corpus.

combine the consistent level PKU and MSR training data into a single extended dataset. Then we randomly shuffle the order of the sentences and divide the data into 10 equal pieces. We use the basic CRF-based segmenter trained on 90% data do segmentation on the other 10% (10-fold cross-validation). The results in Table 10 show that the new extended data is highly consistent inside.

Table 11 shows the character length distribution of the words in PKU, MSR, and the extended data. Since the synthetic word parser is performed on the words with three or more characters, we just ignore the 1-char and 2-char words (generally considered as the smallest units in Chinese) in this table. MSR is obviously coarse standard data compared to PKU with higher rates for nearly all character lengths. Particularly, words with seven or more characters account for approximately 11.6% of all word types in MSR, while the rates of the other two corpora are only 1.6% and 2%. The extended data have the most fine-grained level among the three corpora.

The highly different character length distribution of words in two corpora also prevent us from build a natural baseline of directly predicting the test data of one corpus with a model trained on the other corpus and vice versa. The previous researches of incorporating heterogeneous data have included such baselines of very low performance. For these reasons, we do not include these results for comparison.

System	PKU				MSR			
	Precision	**Recall**	**F-score**	**R_{oov}**	**Precision**	**Recall**	**F-score**	**R_{oov}**
Baseline	96.34	95.69	96.01	81.87	97.13	97.35	97.24	72.5
Internal Structure Information								
+ bpe ($10K$)	96.21	95.37	95.79	81.58	97.08	97.36	97.22	72.61
+ bpe ($20K$)	96.21	95.55	95.88	81.33	97.12	*97.45	97.28	72.43
+ bpe ($30K$)	96.28	95.76	96.02	81.17	97.12	*97.47	97.29	72.32
+ bpe ($40K$)	96.30	95.81	96.05	81.13	97.13	*97.51	97.32	72.47
+ bpe ($50K$)	96.30	95.81	96.05	81.13	97.10	*97.48	97.29	72.12
+ leftmost	96.43	*95.87	*96.15	*83.43	*97.24	*97.45	*97.34	*73.89
+ synthetic words	*96.45	*95.92	*96.19	***83.57**	***97.28**	*97.46	***97.36**	***74.03**
Heterogeneous Data								
Jiang et al. [15]	***96.57**	*95.96	*96.26	*82.38	97.16	97.33	97.25	*73.5
Proposed-1	96.26	*96.18	*96.22	*82.2	97.03	*97.46	97.23	*73.25
Heterogeneous + Sub-word								
Proposed-2	96.36	***96.27**	***96.31**	*83.07	97.21	***97.52**	***97.36**	*73.38

Table 12. Comparison of the proposed methods to the state-of-the-art Chinese word segmenter using heterogenous data and on PKU and MSR corpora. Proposed-1 can be directly compared to Jiang et al. [15] because they use the same baseline segmenter and heterogeneous data. Proposed-2 finally combines heterogenous data and synthetic word parsing. * denotes significance at $p < 0.05$, compared to the baseline.

4.3.3 Main Results

Table 12 summarizes the main segmentation results obtained by the proposed methods. Here, we specify two different settings **Proposed-1** and **Proposed-2** for our model. In the first setting, the system simply uses the consistent level extended training data (Section 4.2.1) to train the first-stage segmenter (the synthetic word parser process is omitted). **Proposed-1** helps us estimate the benefit of larger extended training data (including heterogeneous data). In the second setting, the extended training data are converted to a finer-grained standard by synthetic word parsing. The improvement of **Proposed-2** compared to **Proposed-1** indicates the benefit of including the internal structure information of words.

Since analyzing the internal structure information of words is a general component in our pipeline framework, the synthetic word parser can be flexibly alternated by other sub-word analysis algorithms. In the 'Sub-word Information' parts, we first investigate the benefits provided by three different sub-word seg-

mentation analyzers. In Table 12, 'synthetic words' denotes the segmentation system based on the baseline segmenter with all the words in the training data transformed into sub-word segmentation by a synthetic word parser, 'bpe' denotes the system with sub-word segmentation predicted by byte-pair encoding[10] (Sennrich et al. [32]) with different vocabulary settings, and 'leftmost' denotes the system with sub-word segmentation predicted heuristically by a leftmost dictionary match (using the NAIST Chinese Dictionary). 'synthetic words', and 'leftmost' demonstrates improvements on both PKU and MSR, and our synthetic word parser achieves the most overall gains, particularly in OOV Recall.

The results of 'bpe' are not stable. 'bpe' does not show drop on MSR when the vocabulary size is $10K$; compared to the very low performance on PKU. As vocabulary size increases from $10K$ to $50K$, the recall of 'bpe' increases. However, OOV recall is dropping continuously. When the vocabulary is $40K$, 'bpe' obtains the highest F-score, which is slightly better than the baseline. Although 'leftmost' obtains F-score and OOV recall that are close to 'synthetic words', we find some issues such as 1) 总 / 司令 (chief/commander) is incorrectly split into 总司 (Japanese given name) / 令 (command). 2) the inconsistency of 太仓 / 县 (Taicang/county) and 宣汉县 (Xuanhan county) caused by the absence of 宣 汉 in the dictionary. Moreover, the synthetic words parser is designed to predict real tree structure rather than such flat sub-word segmentation.

Although Jiang et al. [15] reported a F-score improvement +0.8 on Chinese TreeBank 5.0 (CTB5) (guided by People's Daily), the results of re-implementation obtained on PKU and MSR are surprisingly lower. This can be attributed to two points: 1) the difference in the segmentation standards between PKU and MSR is large (as shown in Table 11), which brings a big barrier to benefit each other. 2) the large difference in the sizes of the two corpora ($19K$ and $87K$ sentences). It is difficult for a large corpus to obtain improvements considering the additional loss in the pipeline processing.

Both Jiang et al. [15] and Chao et al. [6] intended to improve word segmentation performance on a small data with large heterogeneous data. As Chao et al. [6] showed in the paper, their coupled CRF obtains an F-score improvement of +0.5 on Chinese TreeBank 7.0 (CTB7, $50K$) with the help of PD ($280K$) while

[10]https://github.com/rsennrich/subword-nmt

the guide-feature method obtains an improvement of +0.34. In our experiments, we also investigate another case, i.e., a large corpus (MSR, $87K$) guided by a small corpus (PKU, $19K$). Jiang's method obtains an improvement of +0.21 on PKU (guided by MSR) and +0.02 on MSR (guided by PKU), while **Proposed-1** shows slightly lower F-scores. The results of the Jiang's method and **Proposed-1** on MSR suggest that the large corpus can hardly obtain an obvious improvement guided by a small corpus, considering the additional loss from different segmentation standards and chunking. However, our consistent level data are friendly to incorporate synthetic word parsing to further boost performance.

Parsing synthetic words (**Proposed-2**) contributes an additional +0.1 F-score on both PKU and MSR, based on **Proposed-1** (with only heterogeneous data). **Proposed-2** finally obtains +0.3 on PKU and +0.12 on MSR, compared to the baseline, which outperforms Jiang et al. [15] and **Proposed-1**.

4.3.4 Analysis

Although we decompose words into the consistent level standard (similar to the PKU standard shown in Table 11), many synthetic words still exist in this data. Synthetic word parsing helps analyze the internal information of these words, which provides major improvement compared to **Proposed-1** and [15]. We manually detect some difference between the results of **Proposed-1** and **Proposed-2**. OOV words such as 紫团 / 山 (Zituan/mountain) and 二 / 进制 (binary/numeral system) are correctly identified with the help of the internal information of words because other mountain names and 十 / 进制 (decimal/numeral system) exist in the training data. The internal information of synthetic words also helps identify IV words such as 数学 / 家 (mathematic/ian), 有 钱 / 人 (rich/people), because they get more consistency with other words like 教育 / 家 (educat/or). We also observed that some polysemous characters result in ambiguous errors. For instance, 非 can be a prefix 'non-' in 非 / 军事 (non-/military) or an auxiliary verb 'must'.

4.4 Summary

In this section, we have further proposed a method to find a new consistent segmentation level across two different corpora. This consistent level makes it possible for multiple corpora to be extended easily. Using a synthetic word parser, we converted the consistent level extended data to a finer-grained level, in which our first-stage segmenter is expected to provide more accurate prediction of both IV and OOV words. Although the second-stage chunking brings an additional loss, the proposed system achieves state-of-the-art recall and F-score results on both PKU and MSR. We also perform additional investigations of the benefits of three different sub-word analysis algorithms: synthetic word parsing, byte-pair encoding and leftmost dictionary match. A further idea for improvement is that part-of-speech information may offer important clues for the second-stage chunking prediction.

Chapter 5

5. A Hybrid Approach for Chinese Spelling Check

Chapter 5 is organized as follows. First, we briefly introduce the background of Chinese spelling check in Section 5.1 and the related work in Section 5.2. A overview of our framework is presented in Section 5.3, including the **Candidate Generation** component (Section 5.3.1 and 5.3.2) and the **Candidate Ranking** component (Section 5.3.3). In Section 5.4, we discuss the experiment setting and results. The conclusion is made at the last section.

5.1 Introduction

Spelling check, which is an automatic algorithm to detect and correct human spelling errors in every written language, has been an active research area in Natural Language Processing (NLP) [36, 22, 8]. As a fundamental process taking raw documents with spelling errors inside as inputs, Chinese spelling check can be seen as a prior step before word segmentation and other down-streaming Chinese NLP tasks. The study of such human spelling errors can help both native speaker and second language learners [20], as well as language processing systems, such as web search engines [36].

Unlike Indo-European languages, Chinese has a non-alphabetic writing system without word boundary indicator, such as space. Its graphic unit **character**, do not represent phonemes, but rather morphemic syllables. Furthermore, the average length of a Chinese word is very short: usually one to three characters. Because of such differences, Chinese Spelling errors can not be detected inside a single word like English, but in a larger range with more context information. For these reason, Chinese spelling check requires more researches.

Current research shows that 97% of Chinese spelling errors are due to **phonological** and **visual** similarity between the correct character and spelling error. [8, 21]. Phonologically similar Chinese characters have similar pronunciation, which involves the nucleus and the tone. For instance, the four phonologically

similar Chinese characters of 挫 (cuo4) are: 1) 措 (cuo4), with same nucleus and same tone, 2) 撮 (cuo1), with same nucleus and different tone, 3) 啜 (chuo4), with different nucleus and same tone and 4) 戳 (chuo1), with different nucleus and different tone. On the other hand, the visually similar Chinese characters share the same partial component, such as the Chinese character 挫 mentioned before is similar to the Chinese character 銼by sharing the part 坐. Based on these evidences, Chinese Spelling Check can be treated as a process to reduce the scope of a large-scale phonological and visual error candidates.

In this paper, we propose a novel hybrid framework to deal with Chinese Spelling errors and perform our experiments on the data of Chinese Spelling Check shared task of the Seventh SIGHAN Workshop (SIGHAN 7). Our framework includes two main parts: **candidate generation** and **candidate ranking**. For the candidate generation step, our effort is to generate as many as possible correction candidates in the confusion set provided by the shared task. In the ranking step, we select the most possible characters to correct the errors in the given sentence. Additionally, to address the scarceness of resources, we further generate around 2 million artificial training sentences by using the Chinese character confusion sets.

5.2 Related Work

As we mentioned that 97% of Chinese spelling errors are made due to **phonological** and **visual** similarity to the correct characters. One important research topic is the generation of **confusion sets**, whichn are collections of spelling error candidates for each Chinese character. In the early work of Chang [5], the confusion sets were manually edited from 4 viewpoints, i.e., shape, pronunciation, meaning and input keystroke sequence. Then by substituting each character in the input sentence with the characters in the corresponding confusion set, they used a language model to generate a plausibility score to evaluate each possible substituted sentence. Because of the importance of confusion sets, some researchers attempted to automatically extend confusion sets by using different Chinese input methods. Intuitively, the characters with similar input key sequences are similar in shape. Furthermore, similar input sequence is a main factor of misspelling errors in electronic documents. Zhang [44] proposed a method to automatically

generate confusion sets based on the Wubi method by replacing one key in the input key sequences of a certain character. Lin et al. [20] used the Cangjie input method to extend confusion sets automatically.

One main approach for Chinese spelling check is Language Model based (LM-based) method [22, 43, 7], which is usually trained on a large-scale corpus to evaluate the correctness of all the possible sentences with correction candidates replaced each time. Another approach is the Statistical Machine Translation based (SMT-based) model [22, 9, 40]. Given a input sentence with or without errors as **'source'**, a SMT-based method is trained to 'translate' it to the best correction sentence as **'target'**. However, the shortcoming is that large bitexts are required to train a good translation model.

Our hybrid framework intends to enlarge the size of the correction candidates by gathering the corrections generated by several different models. Furthermore, an SVMs classifier is employed to evaluate the correctness of each corrections in a candidate list for more accurate performance.

5.3 Our Hybrid Framework for Chinese Spelling Check

The processing steps of our hybrid system are shown in Figure 24, which includes two key components: **Candidate Generation** and **Candidate Ranking**.

- **Candidate Generation** is the process to collect the correction candidates generated from our LM-based model and SMT-based model, which is expected to obtain larger size candidate lists compared each single model.
- **Candidate Ranking** is the process to select the most possible correction in each candidate list (generated by the previous step), which is expected to obtain more accurate results compared to each single model.

5.3.1 Candidate Generation with Language Model

The confusion set, which lists likely phonological and visual character confusions, is a very valuable resource in generating likely correction candidates. However, it does not include context information, so it may over-generate candidates if applied blindly to a sentence. Therefore we propose a method that combines the

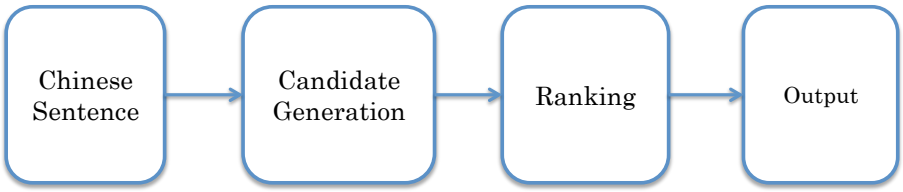

Figure 24. Our hybrid framework for Chinese spelling check.

confusion sets with language models (LMs), which effectively handle context, to efficiently generate correction candidates.

The LM based candidate generation includes three steps: 1) word segmentation, which breaks the Chinese sentence into character chunks, called words; and then 2) candidate lattice construction based on the confusion sets, which are a set of Chinese characters with corresponding similar shape and similar pronunciation characters; 3) generation of the k-best most likely candidates by using the forward algorithm. Next, we will describe the three steps in detail as follows.

Chinese word segmentation, which breaks the Chinese sentence into words, is one of the fundamental parts of the Chinese language processing. In this study, we use the character-based Conditional Random Fields (CRFs) model for Chinese word segmentation [42] by using the open source CRFsuite[11] which tends to perform well in out-of-vocabulary recall. The model is trained on the Academia Sinica corpus, released under the Chinese word segmentation bake-off 2005[12] and the feature templates are the same as in Sun [34].

Given a Chinese sentence with spelling errors, the word segmentation results near the spelling error character are divided into two categories:

- spelling error character is in a multiple characters word.
  For instance, in a segmented word sequence (where words are delimited by slash "/" and red character indicates error): 想必/他們/很/煩腦/吧/！" the error character 腦 (brain) is in a multiple character chunk 煩腦. Since it is not a word, it is not included in the Chinese dictionary.

[11]The CRFsuite package: http://www.chokkan.org/software/crfsuite/
[12]http://www.sighan.org/bakeoff2005/

- spelling error character is in a single character word. For instance, in the segmented sequence: "學校/的/課 桌椅/大/不/分/在/上面/都/有/許多/的/塗鴉/。" the spelling error character 不 (no) is segmented into a single character word.

Based on such evidence, which is also reported by [40, 43], we construct the candidate lattice by the following rules:

- If a word only contains a single Chinese character, add all the candidates in the confusion set.
- If a word contains more than one Chinese character and it is not in the dictionary, then replace all the characters in the word with candidates in the confusion set. If the generated word is in the dictionary, add it as a candidate.
- If a word contains more than one Chinese character and it is in the dictionary, do nothing.

The previous sub-sequence of Chinese sentences are built as shown in Figure 25. For instance, the Chinese word 煩腦, is not in the Chinese dictionary, however, by replacing the candidates from confusion sets, we find that the word 煩惱 (worry or bother) in the Chinese dictionary, and we add it into the lattice. Since for a single character word, there is no way to reduce the candidates in the confusion sets, we add all the candidates in the confusion sets as nodes in the lattice. For instance, the single character word 很 (very, quite), we add all the candidates in the confusion sets, like 恨 (hate), as nodes into the lattice.

Finally, the forward algorithm [31] is used to find the k-best sentences, where the score for each sentence is computed by

$$P(X_1X_2...X_N) = \prod_i P(X_i|X_{i-n}...X_{i-1}) \tag{7}$$

here, $X_1X_2...X_N$ denotes a sequence of Chinese characters (a.k.a, a sentence), N is length of the sentence and n is order of language model. We estimated the conditional probabilities of the language model on Chinese Gigaword by using the open source SRILM[13] package. From our experiments on both the training

[13]http://www.speech.sri.com/projects/srilm/

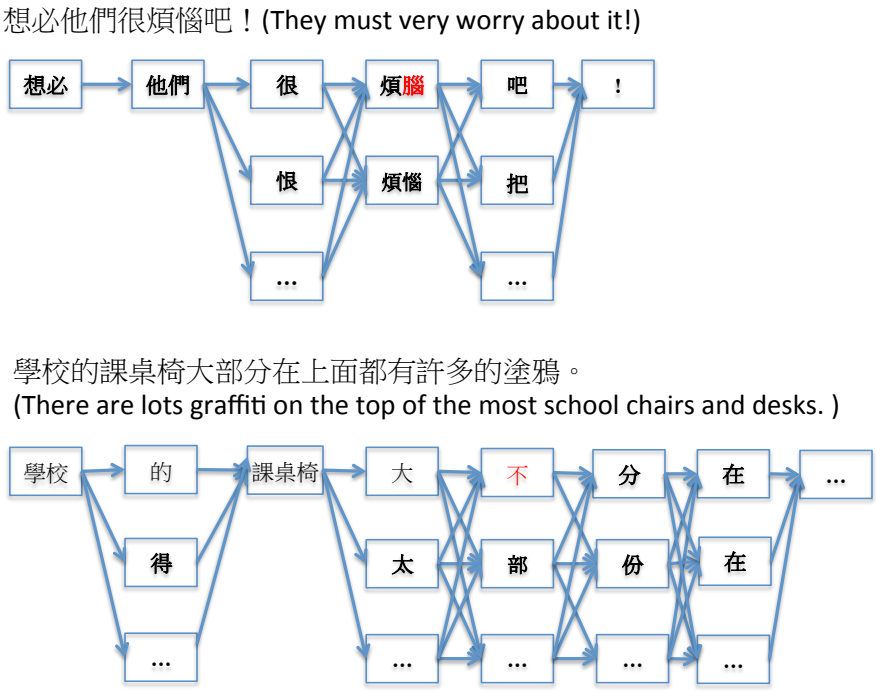

Figure 25. Examples of generated candidate lattices. Note that we first perform the word segmentation and then build lattices. The highlight characters are the spelling errors.

data and the dry run data provided by SIGHAN 7, when the order of language model is larger than 5, the accuracy does not change any more. Therefore, the parameter, the order of the language model, is set to 5 in our experiment.

5.3.2 Candidate Generation with Statistic Machine Translation

As an alternative, we also employ a statistical machine translation model [2] as a way to detect and correct character errors [40].

Given a sentence with errors treated as a source language, our goal is to find the best correction sentence. Formally, given a sentence S which might contain error characters in it as a source sentence, the output is the sentence $\hat{C}$ in the target language with the highest probability of different replacement C.

Symbolically, it is represent by:

$$\hat{C} = \arg\max_{c} p(C|S) \tag{8}$$

Using the Bayes Rule, we can rewrite Formula 8 as:

$$\begin{aligned} \hat{C} &= \arg\max \frac{p(S|C)p(C)}{p(S)} \\ &= \arg\max p(S|C)p(C) \end{aligned} \tag{9}$$

Here, $p(S|C)$ is called "error model", which is the chance that a Chinese character is wrongly written; while $p(C)$ is the language model which evaluates the quality of the corrected Chinese sentence. Traditionally, $p(S|C)$ can be estimated by using the word alignment models [2, 29] from the "error-correct" sentence pairs.

Unlike the language model based candidate generation model, the SMT models detect and correct the spelling errors by incorporating both the "error model" and language model. If we have a large training corpus to estimate a better "error model", which is treated as how likely a Chinese character is wrongly written, we can obtain better results. However, due to the scarcity of training data for the "error model", it is difficult to estimate the true parameters of "error model". To deal with this problem, we generate 2 million of artificial training data by replacing each character in the provided 700 sentences in the training set with candidates in the confusion sets, as shown in Fig 26. Given a sentence, we traverse each character c in the sentence, and replace c by its all candidates $\hat{c}$ in the confusion sets. For example, we replace Chinese character 想 at the beginning of sentence by its candidates in confusion sets 享 and 香; and then we treat the generated sentence (享必他們很煩腦吧!) with the gold standard (想必他們很煩惱吧!) as a training pair for SMT. These generated training data are very important for training "error model" SMT, because most candidates, not observed in training data, have zero probabilities. Empirically, we conduct a set of experiments and observe that without generated data, SMT can not generate any candidates. One of important assumption under this algorithm is that the probabilities of error candidates in the confusion sets tend to be same, in other words, it is a uniform distribution; and by observing more spelling errors, the probabilities of such candidates increase while other candidates, which could not

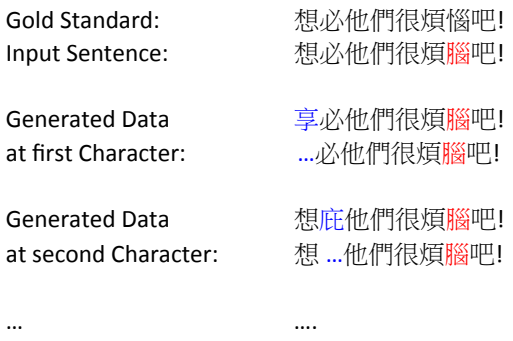

Figure 26. An example of generating the training data for SMT.

be observed, decrease. Therefore, comparing with LM model, SMT model can generate more candidates.

To train our "error model", we adopt the IBM 4 model with default iteration number ($1^5 3^3 4^3$) by using the GIZA++ toolkit, which is an open source package[14] for word alignment. For the language model, we use the SRILM package, mentioned before, with Kneser-Ney smoothing algorithm. The *order* of LM is set to 5, same as the LM generation method.

In the decoding step, we employ the Moses toolkit[15] to find the best translations. Since there is no character re-ordering in Chinese Spelling check, we disable the distortion, setting it to 0, in all the experiment. For parameter optimization, we tried Minimum Error Rate training (MERT) in the Moses toolkit, however, the results only changed slightly. Therefore, we only tune the language model factor, denoted as α, of SMT model in the following experiment.

5.3.3 Candidate Ranking with Support Vector Machines

Although any ranking algorithm can be a ranking component in our framework, for simplicity yet good performance, we adopt the Support Vector Machines (SVMs) in our system, which are supervised learning models used for classification and regression analysis [3]. The goal of **spelling error detection** is to detect whether there are any errors in a given sentence, which can be treated as

[14]https://code.google.com/p/giza-pp/
[15]http://www.statmt.org/moses/

a series of binary classification problems[16]: if the current character is a spelling error, the result is 0, otherwise the result is 1.

Pitler et al. [30] use the SVMs-derived confidence score[17] to determine the brackets between two words in English noun phrases. It inspired us to use such confidence score to determine how likely the current character is a spelling error. By merging the original input character and the error candidates of the previous models, the system creates a candidate list for each character in the input text. And then candidates in the list will be ranked based on the confidence score computed by the SVMs classifier. Finally the character with the highest confidence score will be treated as the correct character of our system. Besides improving precision, this approach also allows us to perform error detection task and the correction task simultaneously. In Figure 27, we give an example to show how our ranking component works. For instance, for the fourth character 有 in the input sentence 他們擁有不怕固難勇与面對的心, the SMT 1-best model predicts a candidate 由. Then we merge the input character and the predicted candidate and create a candidate list containing two characters: 有 and 由. Finally, we rank them based on the confidence score computed by the SVMs classifier and pick the candidate with the highest score as the output.

The features for our SVMs classifier are defined as follows: we denote a character token c_0 with a context sequence: $...c_{-2}c_{-1}c_0c_{+1}c_{+2}...$ and $c_{s:e}$ as a character sequence that starts at the position s and ends at position e. Our system extracts the following features for each candidate:

- **Character features**: c_{-3}, c_{-2}, c_{-1}, c_0, c_{+1}, c_{+2}, c_{+3}, $c_{-1:0}$, $c_{0:+1}$, $c_{-1:+1}$, $c_{-2:-1}$, $c_{+1:+2}$.
- **Pointwise mutual information** [1] between two characters: $PMI(c_{-2}; c_0)$, $PMI(c_{-1}; c_0)$, $PMI(c_0; c_{+1})$, $PMI(c_0; c_{+2})$.
- **Dictionary & N-gram features**: Identity of the character sequence if

[16]It can be treated as a sequence labeling problem as well, where we have two tags: 0 (not a spelling error) and 1 (a spelling error). However, the results were very bad by using such kind of model, such as Conditional Random Fields (CRFs) model. It is likely due to the unbalance label problem (we only have average less than 2 spelling errors in each sentence).

[17]http://www.csie.ntu.edu.tw/ cjlin/libsvm/faq.html Following the instruction of Q06: probability outputs.

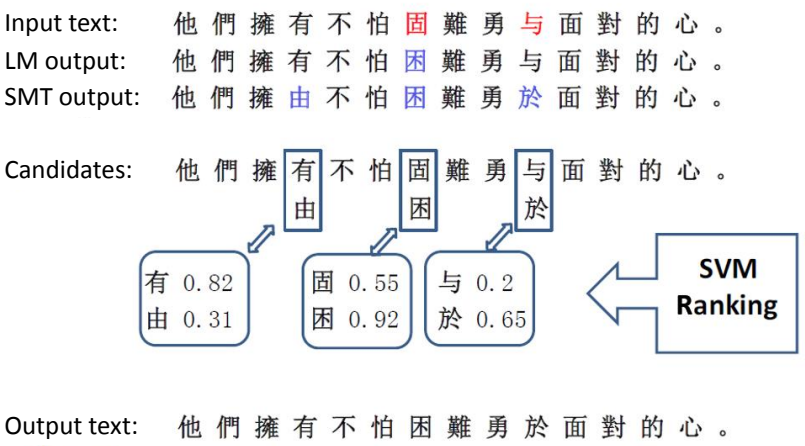

Figure 27. An example of SVMs ranking. The SVMs is applied independently at each candidate position.

it exists in the dictionary and the n-gram word list. For instance: 2-character window $c_{-1:0}$, 3-character window $c_{-2:0}$, 4-character window $c_{-3:0}$, 5-character windows $c_{-4:0}$

We adopt the LIBLINEAR toolkit, an open library for SVMs[18], to train our linear classifier with the L2-loss function. We tune the penalty parameter of the error term C, by using the 5-fold cross-validation on the training data and the dry run data as well.

Further, we additionally applied a down-scaling factor to the confidence scores of the generated candidates in order favor the original input character and prevent excessive correction. Empirically on the dry run data, we found that multiplying the confidence scores of the generated candidates (but not the original input character) by a factor of 0.625 gave good precision and recall.

To obtain more training data for the classifier, we generated 100-best artificial sentences using the LM procedure described in Section 3.1. Correct candidates are labeled as positive and incorrect candidate characters are labeled as negative.

[18]http://www.csie.ntu.edu.tw/ cjlin/liblinear/.

In our data set, the number of training sentences is small and the average number of spelling errors in each sentence is one, and we found that such a large k-best is helpful for ensuring that the SVMs observes the corrections.

5.4 Experiments

5.4.1 Datasets

To train our models, we use the following data sets:

1. **Confusion Set:** Confusion sets are sets of Chinese characters including 5,401 common Chinese characters with similar shape, same and similar pronunciation candidates, which is further divided into five categories: 1) same nucleus and same tone; 2) same nucleus and different tone; 3) different nucleus and same tone; 4) different nucleus and different tone; 5) same number of stokes. For more details, please refer to the Bakeoff 7 [41, 21].

2. **Dictionary:** CC-CEDICT, a free traditional Chinese dictionary released by Creative Commons Attribution-Share Alike 3.0 License is used in our experiment.[19] It total includes around 71,886 Chinese words which contains two or more Chinese characters.

3. **Bakeoff 7 Data:** The statistics of the data set, including the training, dry run and test data, are shown in Table 13. The training data only contains 700 sentences, including 350 sentences without spelling errors which are pretty small to train a better statistical model. Furthermore, comparing with test data, the error rate and error distributions are totally different between the training data and dry run data.

4. **Generated Artificial Data:** From the Table 13, we can see that the size of the training data is insufficient to estimate a better "error model" for training the SMT model. To handle with this problem, we generated around 2 million sentences from the training data by replacing each character in the provided 700 sentences with candidates in the confusion set, as shown in Fig 26.

[19] http://www.mdbg.net/chindict/chindict.php?page=cedict

Data	Task	#sentence	#sentence with errors	#total errors	#sentence with N errors		
					N = 1	N = 2	N ≥ 3
Training	Sub1&2	700	350	350	350	0	0
DryRun	Sub1	50	10	15	5	5	0
	Sub2	50	50	74	36	9	5
Test	Sub1	1,000	300	376	238	49	13
	Sub2	1,000	1,000	1,266	788	168	44

Table 13. **Statistics of Training, Dryrun and Test Data**. Here, Sub1 and Sub2 indicate Chinese Spelling Error Detection Task and Chinese Spelling Error Correction Task, respectively.

5. **Chinese Segmentation Data:** Academia Sinica corpus in Bake-off 2005 for traditional Chinese word segmentation is used to train the segmenter, CRF-based Chinese word segmentation tools, in our experiment. The corpus includes a total of 5,449,698 words and 8,368,050 Chinese characters[20].

6. **Chinese Gigaword:** We use the whole traditional Chinese documents, acquired from Central News Agency of Taiwan, in Chinese Gigaword Second Edition, which is released by the Linguistic Data Consortium (LDC)[21]. The traditional Chinese corpus includes 1,769,953 documents and 792 millions words.

5.4.2 Evaluation Metrics

In the Chinese Spelling Check shared task, there are two sets of evaluation metrics for **Error Detection** (sub-task 1) and **Error Correction** (sub-task 2) [41], which are introduced in detail as follows.

For the **error detection** sub-task, the shared task adopts sentence level metrics for performance evaluation, defined as:

$$False-Alarm\,Rate\,(\mathbf{FAR}) = \frac{\#sentences\_with\_false\_positive\_errors}{\#testing\_sentences\_without\_errors} \quad (10)$$

$$Detection\,Accuarcy\,(\mathbf{DA}) = \frac{\#sentences\_with\_correctly\_detected\_results}{\#all\_test\_sentences} \quad (11)$$

[20]http://www.sighan.org/bakeoff2005/

[21]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId =LDC2005T14

$$Detection\ Precision\ (\mathbf{DP}) = \frac{\#sentence\_with\_correctly\_detected\_errors}{\#test\_sentences\_with\_true\_errors} \quad (12)$$

$$Detection\ Recall\ (\mathbf{DR}) = \frac{\#sentence\_with\_correctly\_detected\_errors}{\#test\_sentence\_with\_errors} \quad (13)$$

$$Detection\ F1\ (\mathbf{DF1}) = \frac{2 * DP * DR}{DP + DR} \quad (14)$$

$$Error\ Location\ Accuarcy\ (\mathbf{ELA}) = \frac{\#sentences\_with\_correct\_error\_locations}{\#all\_test\_sentences} \quad (15)$$

$$Error\ Location\ Precision\ (\mathbf{ELP}) = \frac{\#sentences\_with\_correct\_error\_locations}{\#sentences\_with\_true\_errors} \quad (16)$$

$$Error\ Location\ Recall\ (\mathbf{ELR}) = \frac{\#sentences\_with\_correct\_error\_locations}{\#test\_sentences\_with\_errors} \quad (17)$$

$$Error\ Location\ F1(\mathbf{FLF1}) = \frac{2 * ELP * ELR}{ELP + ELR} \quad (18)$$

For the **error correction** sub-task, the shared task adopts similar sentence level evaluation metrics, which is defined as follows:

$$Location\ Accuracy\ (\mathbf{LA}) = \frac{\#sentences\_correctly\_detected\_error\_location}{\#all\_test\_sentences} \quad (19)$$

$$Correction\ Accuracy\ (\mathbf{CA}) = \frac{\#sentences\_correctly\_corrected\_error}{\#all\_test\_sentences} \quad (20)$$

$$Correction\ Precision\ (\mathbf{CP}) = \frac{\#sentences\_correctly\_corrected\_error}{\#sentences\_corrected\_by\_system} \quad (21)$$

However, the sentence-level evaluation metric provided by the shared task can not make a accurate measurement on the sentences with errors more than one. A reasonable character-level evaluation metric is expected to help us to tune our **Candidate Generation** and **Candidate ranking** component. Therefore, we further employ new metrics to evaluate the accuracy of Chinese Spelling Check in the character-level. The additional evaluation metrics are:

$$Precision_{sub1} = \frac{\#correct\_error\_location\_detected}{\#predicted\_error\_location} \tag{22}$$

$$Recall_{sub1} = \frac{\#correct\_error\_location\_detected}{\#true\_error\_location} \tag{23}$$

$$F - score_{sub1} = \frac{2 * Precision_{sub1} * Recall_{sub1}}{Precision_{sub1} + Recall_{sub1}} \tag{24}$$

$$Precision_{sub2} = \frac{\#correct\_error\_candidate\_detected}{\#predicted\_error\_candidate} \tag{25}$$

$$Recall_{sub2} = \frac{\#correct\_error\_candidate\_detected}{\#true\_error\_candidate} \tag{26}$$

$$F - score_{sub2} = \frac{2 * Precision_{sub2} * Recall_{sub2}}{Precision_{sub2} + Recall_{sub2}} \tag{27}$$

In the next two sub-sections, we will use our character-level metric to evaluate the true qualities of our candidate generation and candidate ranking components.

5.4.3 Candidate Generation Quality

An investigation of the candidate generation quality is very import, because the candidate generation step determines the upper bound of the recall of our system and the missing corrections can not be recovered by the second ranking step. In this subsection, we investigate the performance of our candidate generation component, which includes the language model based approach and statistical machine translation method. We are interested in knowing how many k-best candidates we need to generate to achieve good recall. Note that in our paper, we only concatenate all the candidates from different systems together. Although, there are different ways to make LM and SVM work together, such as intersection

the candidates from both systems, it may loss the true candidates, which cannot be recovered in the ranking part. Our strategy is obtaining a higher recall in candidates generation step and obtaining a higher precision in ranking part. In the following experiments, we perform them on the dry run data set to tune and to select the parameters.

Systems	k-Best Oracle Recall Computed by Eq 23					
	K=1	K=2	K=5	K=10	K=15	K= 20
LM	0.6	0.73	0.9333	1	1	1
SMT	0.6	0.6	0.6	0.6	0.6	0.6
SMT,$\alpha = 0.7$	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
SMT,$\alpha = 0.8$	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
SMT,$\alpha = 0.9$	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
LM + SMT,$\alpha = 0.9$	**0.7333**	**0.7333**	**0.9333**	**1**	**1**	**1**

Table 14. **Comparison of different candidate generation approaches on the error detection (sub-task 1).** Note that k-best denotes the k-best candidate sentences; LM + SMT denotes concatentation of both LM and SMT; and α indicates the language model factor in Moses.

Systems	k-Best Oracle Recall Computed by Eq 26					
	K=1	K=2	K=5	K=10	K=15	K= 20
LM	0.4865	0.5946	0.6351	0.6622	0.7297	0.7568
SMT	0.2568	0.2703	0.2838	0.2838	0.2838	0.2838
SMT,$\alpha = 0.7$	0.3378	0.3514	0.3514	0.3514	0.3514	0.3514
SMT,$\alpha = 0.8$	0.3784	0.3784	0.3784	0.3784	0.3784	0.3784
SMT,$\alpha = 0.9$	0.3919	0.3919	0.3919	0.3919	0.3919	0.3919
LM + SMT,$\alpha = 0.9$	**0.5946**	**0.6757**	**0.7027**	**0.7162**	**0.7568**	**0.7838**

Table 15. **Candidate generation results on the error correction (sub-task 2).** Note that k-best denotes the k-best candidate sentences; LM + SMT denotes concatentation of both LM and SMT; and α indicates the language model factor in Moses.

Table 14 shows the character level recall on error detection task, also known

as the sub-task 1. We show the recall when picking the best possible (oracle) candidate in the k-best list. Note that k refers to the number of candidates generated, which is added to the original input sequence; So, the column k=1 refers to the oracle recall of a list of 2 sentences, the column k=2 refers to the oracle recall of a list of 3 sentences, and so on. Comparing the SMT model with the default language model factor parameter ($\alpha = 0.5$), the LM obtained higher recall. Furthermore, with the number of k-best increasing, the LM outperforms the SMT at all the different settings. We hypothesize this is because (1) the word segmentation module is very effective at reducing the number of incorrect candidates, and (2) even with the artificial training data, the SMT is not able to generate sufficiently diverse k-best when k is relatively small.

Nevertheless, the advantage of a hybrid model can be see, by concatenating the candidate of both the LM and SMT. It outperforms either the single language model or SMT because the candidates generated by the LM and SMT are different. In other words, the candidates generated by the LM and SMT are complementary to each other. The same observation can be obtained from the Table 15, which shows the results for subtask 2 (Error correction). Also, note that the recall on the error detection task (sub-task 1), which is 1 in the 20 best list, is much better than the recall on the error correction task (sub-task 2), which is 0.7838 in the 20 best list. We hypothesize that the reason is the inconsistency of the error distribution and the error rate, which is shown in Table 13.

Figure 28 shows an example of how we catch the oracle candidates, given a real instance in the test data. First, our LM and SMT models generate the 2-best candidates, respectively. Note that these two k-best candidate lists usually contain different error candidates. Last, we merge those all the candidates for ranking.

5.4.4 Candidate Ranking Quality

To estimate the importance of the SVMs ranking step, we perform a set of experiments and tune parameters on the dry run data set. The evaluation metrics are the character-level metrics, introduced in Subsection 5.4.2.

Feature selection plays a crucial role in machine learning community and natural language processing. We are interested in which kinds of features should

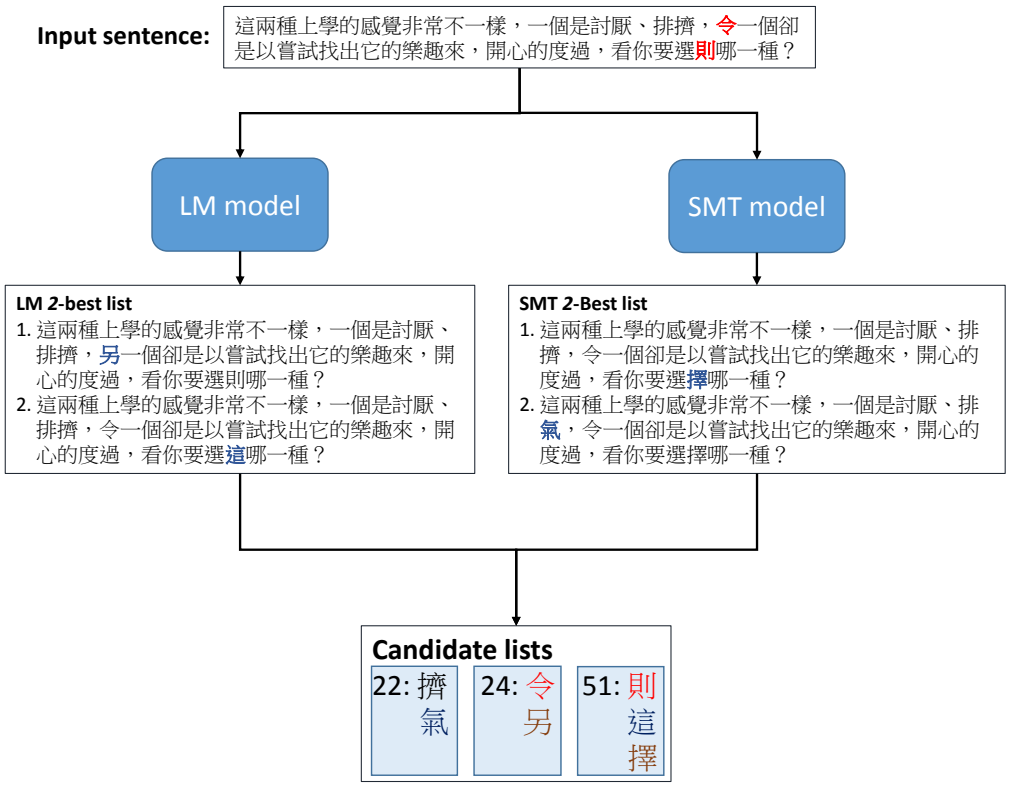

Figure 28. An example of catching the oracle candidate by merging two 2-best list of LM and SMT models. The red characters denote the true errors in the sentence. The blue characters denote the error candidates generated by LM or SMT models. The brown characters denote the true corrections of the gold test data. The number 22, 24, 51 denote the character locations inside the sentence. In this example, our system finally catch all the true corrections in the candidate lists.

be used in our ranking component to obtain a good performance.

As shown in Tables 16 and 17, comparing the SVMs ranking model, which only uses the local context character features, the model which uses all feature types including the local character feature, the dictionary feature, the n-gram language model feature and the pointwise information feature, significantly improved the F-score in both sub-tasks. Comparing to the original LM output, our SVMs

	Precision	**Recall**	**F-score**
LM-based	0.18	**0.6**	0.2769
SVM	0.5	0.2667	0.3478
SVM + DICT	0.625	0.3333	0.4348
SVM + N-GRAM	0.5385	0.4667	0.5
SVM + PMI	0.7	0.4667	0.56
SVM + all	**0.75**	**0.6**	**0.6667**

Table 16. **Comparison of feature selection on error detection (sub-task 1) in the dry run data set.** Here, SVM denotes the SVMs ranking performance with only character features; DICT, N-GRAM and PMI indicate extra dictionary features, the n-gram language model features and the pointwise information features. Please refer to Section 5.3.3 for detail.

	Precision	**Recall**	**F-score**
LM+based	0.72	**0.4865**	**0.5806**
SVM	0.75	0.1216	0.2093
SVM + DICT	**0.8846**	0.3108	0.46
SVM + N-GRAM	0.8235	0.3784	0.5185
SVM + PMI	0.875	0.3784	0.5283
SVM + all	0.8611	0.4189	0.5636

Table 17. **Comparison of feature selection on error correction (sub-task 2) in the dry run data set.** Here, SVM denotes the SVMs ranking performance with basic character features; DICT, N-GRAM and PMI indicate extra dictionary features, the n-gram language model features and the pointwise information features. Please refer to Section 5.3.3 for detail.

model with all feature types obtains significant improvement in the precision from 0.18 to 0.75 without dropping the recall on the error detection task. In the error correction task, although the recall dropped, the SVMs with all features obtains a competitive F-score, which is a trade-off between the precision and recall. One interesting observation is that the LM model obtains a higher F-score compared to our model with the SVMs ranker. The reason is mainly because of the extremely unbalanced error distribution in the training and dry run data as shown in Table 13. On the other hand, it implies that we can choose a larger number of k-best in the candidate generation step to improve the recall, which will be reported in the next subsection.

	Precision	**Recall**	**F-score**
LM	0.18	0.6	0.2769
SMT	0.04	0.6667	0.0755
LM 1-best with SVM ranking	**0.75**	0.6	0.6667
LM 2-best with SVM ranking	0.6471	0.7333	**0.6875**
LM 5-best with SVM ranking	0.52	0.8667	0.65
LM 10-best with SVM ranking	0.518	**0.9333**	0.6667
SMT 1-best with SVM ranking	0.5556	0.6667	0.6061
SMT 2-best with SVM ranking	0.5556	0.6667	0.6061
SMT 5-best with SVM ranking	0.5556	0.6667	0.6061
SMT 10-best with SVM ranking	0.5556	0.6667	0.6061
LM + SMT 1-best with SVM ranking	0.5789	0.7333	0.6471
LM + SMT 2-best with SVM ranking	0.55	0.7333	0.6286
LM + SMT 5-best with SVM ranking	0.4643	0.8667	0.6047

Table 18. **Comparison of SVM ranking on error detection (sub-task 1) in the dry run data.** Note that k-best denotes the number of generated error candidates; these are ranked together with the original input sentence. LM + SMT denotes concatentation of both LM and SMT.

Table 18 shows how the accuracy is affected by the ranking component on the dry run data set for the error detection task (sub-task 1). One observation is that comparing the systems without ranking, our proposed approach with ranking reduces wrongly generated candidates and improves the precision score with a small sacrifice on the recall, however, the F-score is improved. For example, in the LM model of 1-best setting, the precision increased to 0.75 from 0.18 by using the SVMs ranker. One more interesting observation is that the LM outperforms the SMT due to the lack of training data to estimate a better error model, which is introduced in Subsection 5.3.2. Similar observations can be obtained from the Table 19, which is shown the results of sub-task 2. All these evidences demonstrate the importance of our ranking component. To avoiding the bias on the small dry run data, we also conduct a serial experiments on training data by using 5-fold validations and can have similar observation as on the dry run data.

	Precision	Recall	F-score
LM	0.72	0.4865	0.5806
SMT	0.1381	0.3919	0.2042
LM 1-best with SVM ranking	**0.8611**	0.4189	0.5636
LM 2-best with SVM ranking	0.7778	0.473	0.5882
LM 5-best with SVM ranking	0.7059	0.4865	0.576
LM 10-best with SVM ranking	0.6066	0.5	0.5481
SMT 1-best with SVM ranking	0.6111	0.2973	0.4
SMT 2-best with SVM ranking	0.5946	0.2973	0.3964
SMT 5-best with SVM ranking	0.5789	0.2973	0.3929
SMT 10-best with SVM ranking	0.5789	0.2973	0.3929
LM + SMT 1-best with SVM ranking	0.7059	0.4865	0.576
LM + SMT 2-best with SVM ranking	0.7037	**0.5135**	**0.5938**
LM + SMT 5-best with SVM ranking	0.6667	**0.5135**	0.5802

Table 19. **Comparison of SVM ranking on error correction (sub-task 2) in the dry run data.** Note that k-best denotes the number of generated error candidates; these are ranked together with the original input sentence. LM + SMT denotes concatentation of both LM and SMT.

5.4.5 Main results

In the final test, we use the standard test data sets provided by the shared task. Note that we use the best setting, which is empirically suggested by Section 5.4.4. These data sets contain 1000 sentences for each sub-task: the error detection sub-task and error correction sub-task.

In Table 20, FAR denotes the false-alarm rate computed by Eq. 10; DA, DP, DR and DF indicate detection accuracy, detection precision, detection recall and detection f-score, computed by the Eq. 11, Eq. 12, Eq. 13 and Eq. 14, respectively; ELA, ELP, ELR and ELF denote error location accuracy, error location precision, error location recall and error location f-score, computed by Eq. 15, Eq. 16, Eq. 17 and Eq. 18, respectively. Note that SIGHAN7 best1 and SIGHAN7 best2 were the two best systems reported in the shared task, however, they used more resources. The Sinica&NTU1 system used similar resources as ours, however, in their systems of Sinica&NTU2 and Sinica&NTU3, they further used information obtained from a web search engine (Baidu). Our system outperformed the Sinica&NTU1, which used similar resources as ours, and achieved lower false alarm rate, as

Systems	FAR	DA	DP	DR	DF1	ELA	ELP	ELR	ELF
Our result	0.1557	0.8070	0.6646	0.7200	0.6912	0.7350	0.4431	0.4800	0.4608
Sinica&NTU1	0.4471	0.6540	0.4603	0.8900	0.6068	0.5490	0.2793	0.5400	0.3682
Sinica&NTU2	0.1414	0.8350	0.7027	0.7800	0.7393	0.7460	0.4354	0.4833	0.4580
Sinica&NTU3	0.1414	0.8360	0.7036	0.8833	0.7413	0.7490	0.4431	0.4933	0.4669
SIGHAN7 Best1	0.0514	0.8610	0.8455	0.6567	0.7392	0.8200	0.6695	0.5200	0.5854
SIGHAN7 Best2	0.0957	0.8560	0.7690	0.7433	0.7559	0.8050	0.5931	0.5733	0.5830

Table 20. Comparison of final results on error detection (sub-task 1) in the standard test data.

well as higher detection F-score and error location F-score. Furthermore, our system even exceeded the Sinica&NTU2 and Sinica&NTU3 systems in performance, which used additional information from a web search engine (Baidu) [7]. However, our system was beaten by the two best systems reported in the shared task. Comparing our system, these two best systems in shared task used more resources, such as the POS tagging information, a considerable larger dictionary and an large idiom dictionary. We strongly believe that such resources have a great contribution to improve the Chinese Spelling Check System, and they can be flexibly incorporated into our proposed system if available.

We can obtain similar results in the error correction task (sub2) shown in Table 21. Here, LA, CA and CP denote location accuracy, correction accuracy and correction precision, computed by Eq. 19, Eq. 20 and Eq. 21,respectively. Note that SIGHAN7 best1 and SIGHAN7 best2 were the two best systems reported in the shared task, however, they used more resources. The Sinica&NTU1 system used similar resources as ours, however, in their systems of Sinica&NTU2 and Sinica&NTU3, they further used information obtained from a web search engine (Baidu). Our system also outperformed all the three systems of Sinica&NTU, which used the same resource as ours. However, the two best systems, reported in the shared task, also obtained the best performance in the error correction task. To recap, our system significantly outperformed the three systems of Sinica&NTU, which is the best system that used similar resources in the shared task. Even comparing with the state-of-the-art systems, which used more resources than ours, our results are still competitive.

Systems	LA	CA	CP
Our result	0.498	0.457	0.621
Sinica&NTU1	0.507	0.467	0.467
Sinica&NTU2	0.489	0.445	0.445
Sinica&NTU3	0.487	0.450	0.450
SIGHAN7 Best1	0.663	0.625	0.703
SIGHAN7 Best2	0.370	0.356	0.705

Table 21. Comparison of final results on error correction (sub-task 2) in the standard test data.

5.4.6 Error Analysis

Chinese spelling check is a hard problem because the error detection must be done within a context. The situation is that the context information is also ambiguous. We found that our system is quite effective if the local context provides sufficient information, but fails otherwise. The main error types produced by our system are:

- **Semantic error:** In the sentence with id "0003" in the error detection task (sub-task 1), our system detected the 真 (really) as an error in the context 力量真大 (the force is really big.) and correct it to 增 (increase) with the context 力量 增大 (the force increases.). In a local context, both characters 真, and 增, can be regarded as correct in meaning, but our correction is incorrect when considering the semantics of the whole sentence.

- **Pronoun agreement error:** In the sentence with id "0008" in the error detection task (sub-task 1), our system corrected 他 (he) to 它 (it). Unless we can the find its pronoun agreement with the person 貝多芬 (Beethoven) in previous text, it is difficult to determine which is correct.

5.5 Summary

We proposed a simple and effective framework for Chinese Spelling Check which includes two key components: candidate generation and candidate ranking. Firstly, we generated the candidates by using the LM and SMT to achieve large recall.

Then, to improve the precision, we employed an SVMs classifier to rank the generated candidates and give the most likely correction. In this paper, we examined in depth issues such as what type/number of candidates are most effective in improving recall, and what ranking features are best for improving precision. We also proposed a simple approach to improve the SMT model by replacing the characters in the training data with all the candidates in the confusion set, to generate many artificial samples. Our final test results reveal that our framework outperforms other systems, which adopted the same or similar resources as ours in the SIGHAN 7 shared task; even comparing with the state-of-the-art systems, which used more resources, such as a considerable large dictionary, an idiom dictionary and other semantic information, our framework still obtains competitive results.

Chapter 6

6. Conclusion and Future Work

6.1 Conclusion

In this thesis, we improve two fundamental steps before Chinese NLP pipeline: Chinese word segmentation and spelling check.

Our hypothesis of the word segmentation task is that the main segmentation standard and OOVs issues can be addressed by analyzing the internal information of Chinese words. However, due to the lack of existing resources, the first challenge becomes to create an effect way to automatically analyze the word structure.

For this purpose, we carefully build an annotation standard and a synthetic word dictionary with manual annotation. Instead of adopting a traditional pipeline method, we design a novel character-based morphological dependency framework for representing the internal structure of words, which can jointly perform word segmentation and parsing work. Furthermore, our synthetic word parser is flexible to be boosted by several feature types, which are extracted from a dictionary and a large-scale unlabeled corpus.

Our word segmentation system is composed by two-stage processes. The existing word segmentation corpora are first converted to a fine-grained segmentation level by our synthetic word parser. Then, a CRF-based segmenter with the state-of-the-art features is used to predict a new label of each character, which combines both the original and the fine-grained level information. The experiment results show a significant improvement of our segmentation system compared to the baseline segmenter without relying on any new feature types.

We further propose a strategy to transform CWS corpora to a consistent segmentation level, in which multiple corpora can be easily combined to extend larger training data. The extended training data is verified to be highly consistent by cross-validation. Due to the extension of larger training data and flexibility of incorporating internal structure information, our proposed word segmenter

achieves a significant improvement compared to the baseline and state-of-the-art system using heterogeneous data.

For Chinese spelling check, we propose a hybrid spelling check framework with two key components: candidate generation and candidate ranking. Candidate generation intends to enlarge the size of candidate lists by gathering the results from two single spelling check system. Candidate ranking adopts a SVMs classifier to provide a confidence score for each candidate for ranking. The ranking step is designed to provide high precision of the system performance without losing too much recall. Our framework is flexible to extend the correction candidate lists by adding more single spelling check systems and substitute any ranking techniques for the SVMs classifier. Our spelling check system achieves competitive results with less resources consuming, compared to the state-of-the-art systems in SIGHAN 7.

6.2 Future Work

Most modern Chinese words are derived from the Chinese characters, each of which contains an independent meaning. It is a straightforward thinking that Chinese synthetic words can be represented in the form of fully character-level internal trees in the future studies. Rich information can be integrated into this new direction such as Part-of-Speech tag of each character, directed morphological dependency between two characters, etc. It can reinforce the current word structure analyzer (i.e. synthetic word parser) by overcoming two short-comings, i.g. relying on the minimal granularity definition of single-morphine words and undirected dependencies in word-level.

In this work, we demonstrate that the automatic word structure analyzer can improve the performance of Chinese word segmentation. A further extension is to apply our word structure analyzer into another downstream task, i.e. syntactic parsing. It is natural to consider that whether word is necessarily the minimal unit of Chinese, since each Chinese character takes a independent meaning. Combining morphological analysis and syntactic parsing into one parsing model is an attractive goal for the Chinese NLP community.

In our current spelling check system, the language model (LM) method conducts main effort for the candidate generation step. However, a obvious problem

is that the LM replies on the prior word segmentation process. It means that if an error character is segmented into a different word from the correct segmentation, this error is likely to be undetectable for our LM method. There is the space to improve the LM method by including more complex cases of different word segmentation results or even jointly perform the spelling check and word segmentation processes.

In recent years, deep learning has swept across the NLP community. In the task of Chinese word segmentation, conventional CRF models are beaten by various recurrent neural networks. The modern neural networks rely on the combination with lookup tables of words with random initialization or pre-trained process, which capture word representations based on context information. In contrast to this, our internal structure of words provides independent information from words themselves, which can be seen an additional information source to be integrated into word representations in the future.

References

[1] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, pages 31–40, 2009.

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

[3] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[4] Xavier Carreras. Experiments with a higher-order projective dependency parser. In *EMNLP-CoNLL*, pages 957–961, 2007.

[5] Chao-Huang Chang. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium (NLPRS95)*, volume 95, pages 278–283. Citeseer, 1995.

[6] Jiayuan Chao, Zhenghua Li, Wenliang Chen, and Min Zhang. Exploiting heterogeneous annotations for weibo word segmentation and pos tagging. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 495–506. Springer, 2015.

[7] Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. A study of language modeling for chinese spelling check. In *Sixth International Joint Conference on Natural Language Processing*, page 79, 2013.

[8] Yong-Zhi Chen, Shih-Hung Wu, Ping-Che Yang, and Tsun Ku. Improve the detection of improperly used chinese characters in students' essays with error model. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(1):103–116, 2011.

[9] Hsun-wen Chiu, Jian-cheng Wu, and Jason S. Chang. Chinese spelling checker based on statistical machine translation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 49–53,

Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[10] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[11] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[12] Jason M Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics, 1996.

[13] Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93, 2004.

[14] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. Machine learning-based methods to chinese unknown word detection and pos tag guessing. *Journal of Chinese Language and Computing*, 16(4):185–206, 2006.

[15] Wenbin Jiang, Liang Huang, and Qun Liu. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging: a case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 522–530. Association for Computational Linguistics, 2009.

[16] Terry Koo, Xavier Carreras Pérez, and Michael Collins. Simple semi-supervised dependency parsing. In *46th Annual Meeting of the Association for Computational Linguistics*, pages 595–603, 2008.

[17] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence

data. In *Proceedings of the 18th International Conference on Machine Learning*, volume 951, pages 282–289, 2001.

[18] Zhongguo Li. Parsing the internal structure of words: A new paradigm for chinese word segmentation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1405–1414. Association for Computational Linguistics, 2011.

[19] Zhongguo Li and Guodong Zhou. Unified dependency parsing of chinese morphological and syntactic structures. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1445–1454. Association for Computational Linguistics, 2012.

[20] Yih-Jeng Lin, Feng-Long Huang, and Ming-Shing Yu. A chinese spelling error correction system. In *Proceedings of the Seventh Conference on Artificial Intelligence and Applications (TAAI)*, 2002.

[21] C-L Liu, M-H Lai, K-W Tien, Y-H Chuang, S-H Wu, and C-Y Lee. Visually and phonologically similar characters in incorrect chinese words: Analyses, identification, and applications. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):10, 2011.

[22] Xiaodong Liu, Fei Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. A hybrid chinese spelling correction system using language model and statistical machine translation with reranking. In *Sixth International Joint Conference on Natural Language Processing*, page 54, 2013.

[23] Xiaodong Liu, Kevin Duh, Yuji Matsumoto, and Tomoya Iwakura. Learning character representations for chinese word segmentation. In *NIPS 2014 Workshop on Modern Machine Learning and Natural Language Processing*, 2014.

[24] Jia Lu, Masayuki Asahara, and Yuji Matsumoto. Analyzing chinese synthetic words with tree-based information and a survey on chinese morphologically derived words. In *IJCNLP*, pages 53–60, 2008.

[25] Jia Lu, Masayuki Asahara, and Yuji Matsumoto. *Chinese Synthetic word Analysis using Large-scale N-gram and an Extendable lexicon Manage- ment System*. PhD thesis, Nara Insitute of Science and Technology, 2011.

[26] Ryan McDonald. *Discriminative learning and spanning tree algorithms for dependency parsing*. PhD thesis, University of Pennsylvania, 2006.

[27] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 91–98. Association for Computational Linguistics, 2005.

[28] Ryan T McDonald and Fernando CN Pereira. Online learning of approximate dependency parsing algorithms. In *EACL*, 2006.

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

[30] Emily Pitler, Shane Bergsma, Dekang Lin, and Kenneth Church. Using web-scale n-grams to improve base np parsing performance. In *Proceedings of the 23rd International Conference on Computational Linguistics(COLING10)*, pages 886–894. Association for Computational Linguistics, 2010.

[31] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[32] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[33] Weiwei Sun. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1385–1394. Association for Computational Linguistics, 2011.

[34] Weiwei Sun. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT11)*, pages 1385–1394, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[35] Weiwei Sun and Jia Xu. Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 970–979. Association for Computational Linguistics, 2011.

[36] Xu Sun, Jianfeng Gao, Daniel Micol, and Chris Quirk. Learning phrase-based spelling error models from clickthrough data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL10)*, pages 266–274. Association for Computational Linguistics, 2010.

[37] Xu Sun, Houfeng Wang, and Wenjie Li. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 253–262. Association for Computational Linguistics, 2012.

[38] Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. A discriminative latent variable chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64. Association for Computational Linguistics, 2009.

[39] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, volume 171, 2005.

[40] Shih-Hung Wu, Yong-Zhi Chen, Ping-che Yang, Tsun Ku, and Chao-Lin Liu. Reducing the false alarm rate of chinese character error detection and

correction. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 54–61, 2010.

[41] Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[42] Nianwen Xue. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48, 2003.

[43] Jui-Feng Yeh, Sheng-Feng Li, Mei-Rong Wu, Wen-Yi Chen, and Mao-Chuan Su. Chinese word spelling correction based on n-gram ranked inverted index list. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 43–48, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[44] Lei Zhang, Changning Huang, Ming Zhou, and Haihua Pan. Automatic detecting/correcting errors in chinese text by an approximate word-matching algorithm. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL00)*, pages 248–254. Association for Computational Linguistics, 2000.

[45] Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. Exploring representations from unlabeled data with co-training for Chinese word segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 311–321, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[46] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. Chinese parsing exploiting characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 125–134, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[47] Hai Zhao. Character-level dependencies in chinese: usefulness and learning. In *Proceedings of the 12th Conference of the European Chapter of the*

Association for Computational Linguistics, pages 879–887. Association for Computational Linguistics, 2009.

[48] Hai Zhao and Chunyu Kit. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Sixth SIGHAN Workshop on Chinese Language Processing*, page 106.

List of Publications

Fei Cheng

Journal Papers

- Fei Cheng, Kevin Duh and Yuji Matsumoto, Towards a Consistent Segmentation Level Across Multiple Chinese Word Segmentation Corpora, *Journal of Natural Language Processing.* Vol. 24. No. 5. Dec 2017.
- Xiaodong Liu*, Fei Cheng*, Kevin Duh and Yuji Matsumoto, A Hybrid Ranking Approach to Chinese Spelling Check, *ACM Transactions on Asian and Low-Resource Language Information Processing.* 14.4:16. 2015. (* denotes equal contribution)

International Conferences

- Fei Cheng, and Yusuke Miyao. Classifying Temporal Relations by Bidirectional LSTM over Dependency Paths. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* Vol. 2. pp. 1-6. 2017.
- Fei Cheng, Kevin Duh, and Yuji Matsumoto. Synthetic Word Parsing Improves Chinese Word Segmentation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.* Vol. 2. pp. 262-267. 2015.
- Fei Cheng, Kevin Duh, and Yuji Matsumoto. Parsing Chinese Synthetic Words with a Character-based Dependency Model. *Proceedings of the Ninth International Conference on Language Resources and Evaluation.* pp. 67-72. 2014.
- Xiaodong Liu, Fei Cheng, Yanyan Luo, Kevin Duh and Yuji Matsumoto, A Hybrid Chinese Spelling Correction System Using Language Model and Statistical Machine Translation with Reranking, *Proceedings of Seventh SIGHAN Workshop on Chinese Language Processing.* pp. 54-58. 2013.