Mitigating Impacts of Word Segmentation Errors on Collocation Extraction in Chinese

廖永賦 Yongfu Liao
國立臺灣大學語言學研究所
Graduate Institute of Linguistics
National Taiwan University
liao961120@gmail.com

謝舒凱 Shu-Kai Hsieh
國立臺灣大學語言學研究所
Graduate Institute of Linguistics
National Taiwan University
shukaihsieh@ntu.edu.tw

摘要

隨著網路的盛行，自動斷詞與標記的大規模語料庫逐漸普及。自動化不可避免地引入一些斷詞與標記的錯誤，並可能對下游任務產生負面影響。搭配詞的自動抽取是一項受斷詞品質影響的任務。本文探討一些方法試圖減輕斷詞錯誤對漢語搭配詞抽取之影響。我們嘗試了一個結合多個共現訊息的簡單線性模型，試圖減少抽取出之搭配詞含有的斷詞錯誤。實驗結果顯示，此模型無法區分搭配詞是否為斷詞錯誤所導致。因此，我們使用了 FastText 詞向量的訊息進行了另一個案例研究。結果顯示，由斷詞錯誤所產生的假搭配詞與真正的搭配詞，其之間的語義相似性具有不同的特徵。未來研究可嘗試在搭配詞抽取中加入詞向量的訊息。

Abstract

The prevalence of the web has brought about the construction of many large-scale, automatically segmented and tagged corpora, which inevitably introduces errors due to automation and are likely to have negative impacts on downstream tasks. Collocation extraction from Chinese corpora is one such task that is profoundly influenced by the quality of word segmentation. This paper explores methods to mitigate the negative impacts of word segmentation errors on collocation extraction in Chinese. In particular, we experimented with a simple model that aims to combine several association measures linearly to avoid retrieving false collocations resulting from word segmentation errors. The results of the experiment show that this simple model could not differentiate between true collocations and false collocations

resulting from word segmentation errors. An ad hoc case study incorporating information from FastText word vectors is also conducted. The results show that collocates resulting from correct and erroneous word segmentation have different profiles in terms of the semantic similarities between the collocates. The incorporation of word vector information to differentiate between true and false collocations is suggested for future work.
關鍵詞：搭配詞抽取、中文斷詞、詞向量

Keywords: Collocation Extraction, Chinese Word Segmentation, Word Vector

1 Introduction

A collocation, in Firthian sense, is a combination of words that tend to occur near each other in natural language [1]. To measure the tendency for words to co-occur, various statistical measures are proposed to quantify the association strengths of word pairs. These association measures are often used to rank and extract collocations from corpora. As the concept of collocation was developed in the western world, which has a writing system that clearly delimits word boundaries, the adoption of the concept of collocation in languages where the notion of *wordhood* is not clear necessitates a preprocessing step that segments the text into sequences of "words". Computing association measures based on the segmented text to extract collocations thus requires an additional assumption—the word segmentation must return correct results. Otherwise, the collocations extracted might be nonsense—instead of being recurrent "word" combinations, the "collocations" may in fact be "character" combinations that have a tendency to co-occur.

With the prevalence of the internet, large corpora constructed from texts collected from the web has become common. At the same time, manual checking of the automatic segmentation and tagging of the corpora to ensure the quality has become nearly impossible, as the amount of data collected is enormous. In addition, out-of-vocabulary words such as named entities, new terms, and special usage of particular subcultures frequently appear in web texts [2], further casting doubt on the performance of automatic segmentation of the constructed corpora.

Since manual checking and corrections are not practical solutions to counter automatic preprocessing errors in large corpora, it is crucial to be aware of the negative impacts that such errors could have on downstream tasks. For instance, collocations extracted from Chinese social media texts may contain several instances of false collocations that resulted from word

segmentation errors. Table 1 lists the top 16 collocates of the node word 三 'three' retrieved from the social media PTT (see section 2.1 for the data). Strikingly, the top 10 ranked collocations in Table 1 all resulted from word segmentation errors. In other words, there are only 37.5% of the word pairs in this list that count as "true" collocations. The others are not even "word" pairs.

In this paper, we explore the potential of leveraging existing association measures, originally designed to quantify the association strengths between *words*, to detect or filter out false collocations resulting from word segmentation errors in collocation extraction. The assumption is that false collocations may behave differently from true collocates in the patterns of association measures. In particular, we explore the possibility of constructing a new association measure from existing ones that is robust against retrieving false collocations resulting from word segmentation errors.

Table 1. Top 16 collocates that have the highest tendency to co-occur (as measured by MI) with the node word 三 'three' in PTT corpus.

Word 1	Word 2	Frequency	MI	Rank	Word 1	Word 2	Frequency	MI	Rank
三	池崇史	17	10.062	1	三	丁	219	9.154	9
	浦友	5	10.062	2		日月	18	9.022	10
	浦春馬	7	10.062	3		次元	24	7.559	11
	角頭	20	9.683	4		餐	261	7.457	12
	人房	8	9.603	5		小	2367	6.585	13
	人行	52	9.496	6		秒	82	5.655	14
	倍速	8	9.477	7		年前	95	5.563	15
	班制	12	9.325	8		年	716	5.377	16

2 Combining Association Measures

The purpose of this research is to explore the possibility of constructing a robust association measure by combining several association measures, with the aim of mitigating the impact of retrieving false collocations resulting from word segmentation errors in Chinese. Below, we describe the data, the model for combining several association measures, and the training of the model.

2.1 Data

As a preliminary study, we focus here only on association strengths of word pairs occurring in a running window of two (i.e., bigrams). The corpus used to calculate various association measures was constructed from 36,000 texts from PTT forum[1], which is one of the largest online forums in Taiwan. The texts were collected from 12 categories (*BabyMother, Boy-Girl, gay, Gossiping, Hate, HatePolitics, Horror, JapanMovie, joke, LGBT_SEX, NTU, sex*)[2], with 3000 texts sampled from each category. The corpus was segmented with Jseg[3]. Word pair frequencies were then calculated from the corpus.

Eight association measures—MI, MI^3, MI.log-f, t, Dice, logDice, $\mathbf{\Delta}P_{1|2}$, $\mathbf{\Delta}P_{2|1}$—were calculated from the corpus. The first six measures follow the statistics used in the Sketch Engine [3], and the last two measures, $\mathbf{\Delta}P_{1|2}$ and $\mathbf{\Delta}P_{2|1}$, are directional association measures proposed in [4]. The MI measure measures the ratio between the observed frequency ($O = f_{AB}$) and the expected frequency ($E = f_A \cdot f_B / N$) of a word pair (w_A, w_B) on a logarithmic scale. Since MI tends to assign low-frequency word pairs (having low value of E) high scores, varients of the MI measure are proposed to counter this effect. MI^3 achieve this by taking the cube of the observed frequency to strengthen its influence relative to the expected frequency, and MI.log-f counters MI's low-frequency bias by multiplying the MI score with $\ln(O + 1)$. T measures the discrepancy between the observed and expected frequency against the square root of the observed frequency. The Dice coefficient compares the cooccurrence frequency of the word pair against the summed frequencies of the words in the pair. As proposed in [4], $\mathbf{\Delta}P_{1|2}$ and $\mathbf{\Delta}P_{2|1}$ are different from the other measures in that they are "directional" while others are "symmetric". That is, instead of assigning a single score that indicates the strength of "mutual" attraction between a word pair (w_1, w_2), $\mathbf{\Delta}P_{1|2}$ and $\mathbf{\Delta}P_{2|1}$ assign two separate scores to a single word pair—$\mathbf{\Delta}P_{1|2}$ indicates how predictable w_1 is given w_2, and $\mathbf{\Delta}P_{2|1}$ indicates how predictable w_2 is given w_1.

$$\text{MI} = \log_2(O/E) \qquad \text{MI}^3 = \log_2(O^3/E) \qquad \text{MI.log-f} = \text{MI} \cdot \ln(O + 1)$$

$$\text{t} = (O - E)/O^{0.5} \qquad \text{Dice} = 2 \cdot f_{AB}/(f_A + f_B) \qquad \text{logDice} = 14 + \log_2(\text{Dice})$$

$$\mathbf{\Delta}P_{2|1} = p(w_2 \mid w_1) - p(w_2 \mid \text{not } w_1) \qquad \mathbf{\Delta}P_{1|2} = p(w_1 \mid w_2) - p(w_1 \mid \text{not } w_2)$$

Figure 1. Formula of the association measures used in this study. f_A is the frequency of a word

[1] https://www.ptt.cc/bbs

[2] https://www.ptt.cc/bbs/{category}

[3] A modified version of Jieba trained with Sinica Corpus. https://github.com/amigcamel/Jseg

w_A in the corpus; O (or f_{AB}) is the observed frequency of a word pair (w_A, w_B); E (equals $f_A \cdot f_B/N$, where N is the corpus size) is the expected frequency of a word pair (w_A, w_B); p(w_A | w_B) is the probability that w_A occurs before w_B, and p(w_A | not w_B) is the probability that w_A occurs before words other than w_B.

Due to the limitation of computing power, only word pairs with one of the words being a single-character word occurring in the Chinese Lexical Database [5] were calculated for the association measures. In addition, word pairs with frequencies below or equal to 3 were excluded from the calculation. This resulted in a dataset of 334,686 word pairs with their corresponding eight association measures.

2.2 Model

As a preliminary investigation, the model used in this study was intended to be simple and transparent. The model M_{comb} is a simple linear combination of several association measures, as shown in equation (1).

$$M_{comb} = \alpha_1 \cdot \mathbf{M}_1 + \alpha_2 \cdot \mathbf{M}_2 + \alpha_3 \cdot \mathbf{M}_3 + ... + \alpha_n \cdot \mathbf{M}_n \quad (1)$$

In equation (1), M_i is the percentile rank of one of the eight association measures mentioned in the previous section, and α_i is the weight of M_i on the model M_{comb}. The weights α_i are determined by a grid search [9] that finds the best configuration of (α_1, α_2, ..., α_n).

The goal of the model is to retrieve a list of collocations that has a low portion of false collocations resulting from word segmentation errors. To achieve this, we score the model during the grid search as *the portion of "correct" collocations in a list of top n collocations ranked according to* M_{comb}. "Correct" collocations are defined as collocations that (1) do not result from word segmentation errors, and (2) have ranks below 100 in at least m association measures (the parameter "low rank num" in Table 2). Word segmentation errors are defined using a dictionary constructed from tokens in ASBC [6], lexical entries in the Chinese dictionary compiled by the Ministry of Education[4], and Chinese Wikipedia page titles[5]. A word pair is defined as a false collocation if it results from a single lexical entry in the dictionary that is split apart due to a word segmentation error. Note that this definition is limited in that only a certain kind of word segmentation errors (e.g. "蔡 | 英文") is captured.

[4] https://github.com/g0v/moedict-data

[5] https://dumps.wikimedia.org/zhwiki/20200620

Other kinds of segmentation errors such as “軍官軍 | 銜”, which segments a string of two words (“軍官” and “軍銜”) into two words in a wrong way (“軍官軍” and “銜”), would not be captured by this dictionary checking approach. This restricted definition is a compromise since a more precise definition would require costly human annotation of the word pairs.

2.3 Training

The dataset described in section 2.1 was split into 80% for training and 20% for testing. The training set was used to perform the grid search to find the optimal weight configurations for the component association measures. For each iteration (a set of weights α_i), an M_{comb} score can be calculated for each word pair in the training set. The top n collocations were then retrieved according to the M_{comb} scores, from which a score (the proportion of “correct” collocations) could then be assigned to this weight configuration. After the grid search, weight configurations with the highest score were then used to retrieve the top n collocations from the testing set, from which the model was evaluated.

3 Evaluation

To see whether combining several association measures in a linear fashion could improve the results of collocation extraction, we evaluated the top n collocations retrieved by the model against the top n collocations retrieved by each association measure constituting the model. Ninety percent of the testing data were sampled and used for the retrieval of the top n collocations. For each set of top n collocations retrieved by the model and its component association measures, the proportion of “correct” word segmentation was calculated. This process was repeated 100 times, and the distribution of the proportion of the “correct” collocations for the model and its component association measures were compared.

Several configurations of the model were tested, most of which show qualitatively similar results. In the following sections, we describe two versions of the model—one consisting of three component association measures and the other consisting of eight. Table 2 summarizes the models and their performance.

3.1 Model 1: Linear Combination of Three Measures

The first model is a linear combination of three association measures—MI, logDice, and $\mathbf{\Delta}P_{1|2}$:

$$M_{comb} = \alpha_1 \cdot \textit{Percentile}(\text{MI}) + \alpha_2 \cdot \textit{Percentile}(\text{logDice}) + \alpha_3 \cdot \textit{Percentile}(\mathbf{\Delta}P_{1|2}) \quad (2)$$

3.1.1 Model Parameters

During training, the weight configurations (α_1, α_2, α_3) were searched over the space:

$$\{(i, j, k) \mid \forall i, j, k \in S\}, \text{ where } S = \{-1, -0.95, -0.9, \ldots, 0.9, 0.95, 1\}$$

For each weight configuration, 20 collocations with the highest M_{comb} scores were retrieved. The score of a weight configuration is the proportion of "correct" word segmentation in this list of top 20 collocations.

3.1.2 Comparing with Single Association Measures

Training with these parameters resulted in eight weight configurations that reached an optimal score of 0.7 in the training set. For each of them, the distribution of the proportion of the "correct" collocations in the testing set is shown in Figure 2. The optimal M_{comb} model on the testing set has a mean of 46.7% "correct" collocations.

Retrieving the top 20 collocations with the component measures of the M_{comb} model, on the other hand, yields better results—two of the three measures performed better than 46.7%, and even the least performant measure (MI) has an average score of 45.9% (Figure 3).

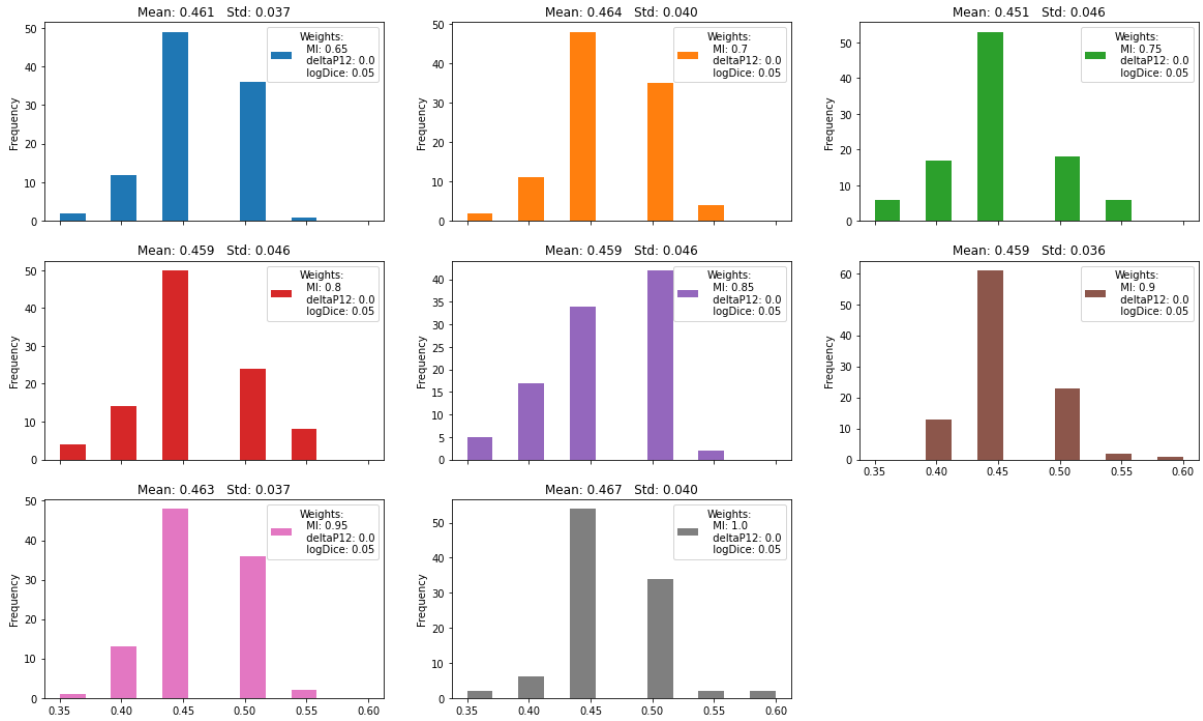

Figure 2. The distribution of the proportion of the "correct" collocations retrieved by each of the eight optimal M_{comb} scores in the testing set. Among these eight optimal weight configurations (on training set), the configuration **1.0** · *Percentile*(MI) +

0.05 · *Percentile*(logDice) + **0.0** · *Percentile*($\mathbf{\Delta}P_{1|2}$) (the subplot in the 3rd row and the 2nd column) achieved the best performance (46.7% "correct") on the testing set.

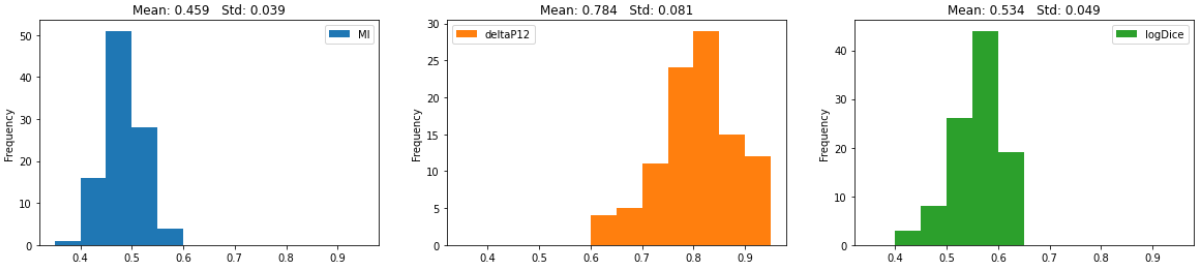

Figure 3. The distribution of the proportion of the "correct" collocations retrieved by each of the component association measures of the M_{comb} model—MI, $\mathbf{\Delta}P_{1|2}$, and logDice. The component measures, at least for $\boldsymbol{\Delta}P_{1|2}$ (the center subplot, 78.4% "correct") and logDice (the rightmost subplot, 53.4% "correct"), performed better individually than combining together into M_{comb} on the testing set.

3.2 Model 2: Linear Combination of Eight Measures

The setup of Model 2 is identical to Model 1 except that there are now 8 components in the model:

$$M_{comb} = \alpha_1 \cdot Percentile(\text{MI}) + \alpha_2 \cdot Percentile(\text{logDice}) + \alpha_3 \cdot Percentile(\mathbf{\Delta}P_{1|2}) + \alpha_4 \cdot Percentile(\mathbf{\Delta}P_{2|1}) + \alpha_5 \cdot Percentile(\text{MI}^3) + \alpha_6 \cdot Percentile(\text{MI.log-f}) + \alpha_7 \cdot Percentile(\text{t}) + \alpha_8 \cdot Percentile(\text{Dice}) \quad (3)$$

Due to the huge search space resulting from the eight weight configurations, instead of a full grid search, 1/10,000 of the search space was sampled and searched on. In addition, the space of the possible values for α_i was set smaller to S = {-1, -0.875, -0.75, …, 0.75, 0.875, 1}.

Evaluated using the procedure identical to Model 1, Model 2 showed no qualitatively different results. Several configurations of the parameters of Model 2 all resulted in models that do not perform better than their component association measures, again, showing that combining several association measures in a linear fashion cannot protect the model from retrieving false collocations resulting from word segmentation errors. Table 2 summarises several parameter settings of Model 1 and Model 2 and the results of the evaluation.

Table 2. Parameter settings and performance of the models in the experiment. For most

models (except 1-3), the performance is worse than at least one of their component measures.

	Parameters				Training	Testing (max correct %)	
Model ID	Component Measures	Top n	Low rank number	Search space	Number of optimal weight configs	Model	Component measures
1-1	MI, $\Delta P_{2\|1}$, logDice	20	2	{1, -0.95, ..., 0.95, 1}	101	0.557	**0.568**
1-2	MI, $\Delta P_{1\|2}$, logDice	20	2	{1, -0.95, ..., 0.95, 1}	8	0.462	**0.794**
1-3	MI, t, logDice	20	2	{1, -0.95, ..., 0.95, 1}	261	**0.632**	0.555
1-4	MI, MI3, t	20	2	{1, -0.95, ..., 0.95, 1}	436	0.504	**0.554**
2-1	All	10	1	{-1, -0.875, ..., 0.875, 1}	10	0.613	**0.796**
2-2	All	20	3	{-1, -0.875, ..., 0.875, 1}	3	0.503	**0.793**
2-3	All	20	2	{-1, -0.875, ..., 0.875, 1}	1	0.552	**0.787**
2-4	All	10	2	{-1, -0.875, ..., 0.875, 1}	13	0.632	**0.797**

4 Discussion

As seen in Table 2, generally, the model performs worse than its component measures. The only exception is Model 1-3, which by combining MI, t, and logDice, attained better performance than its component measures in 3 of 261 weight configurations. Hence, at least in the case of MI, t, and logDice, the combination of association measures may lead to better collocation extraction.

The general failure of the model suggests that if combining association measures could indeed capture patterns of word segmentation errors in collocation extraction, combining the measures in a linear fashion is too simple to capture these patterns. One direction for future research then is to use more complicated models, such as adding interaction terms to the model and see whether these more complicated models could capture word segmentation errors in the collocations. This approach, however, suffers from the exponential growth of the search space, making it computationally expensive or even impossible to find the optimal configurations.

Another direction of future work is to incorporate information additional to association measures into the model. Word vectors are promising candidates for this direction of work, as word segmentation errors might result in nonsense “words”, and these nonsense words might reveal themselves from the pattern of semantic similarities between normal and nonsense words, and between the words in each of these categories. To confirm our intuition, we carried out a pilot case study to inspect the semantic similarities among the words in two lists of collocations. The word 林 ‘Lin (family name)’ and 三 ‘three’ were used in the two lists

respectively as the node word, and their right collocates were extracted. For each of the two lists, collocations were extracted using seven measures (the eight measures except Dice mentioned in section 2.1)—20 collocations ranked as highest were retrieved for each measure, resulting in a list of 140 collocations (with duplications). Then, collocations that appeared less than 3 times were removed from the list (i.e., a collocation needs a rank of at least 20 in at least 3 measures to retain in the list). We then calculated the semantic similarities (cosine similarity of word vectors) between all words with FastText pre-trained word vectors [7]. The results are represented as network plots shown in Figure 4 and 5. The node in the network represents a word (either a node word or its collocates) in a list of collocations. The thickness of the edge between a pair of words indicates the degree of similarity between them, with higher similarity represented by a thicker edge.

One feature that instantly pops out from the figures is that correctly segmented collocates (blue nodes) form clusters. That is, these collocates are similar to each other in terms of semantic similarities as measured by the cosine similarity of their word vectors. On the other hand, collocates resulting from word segmentation errors are much more spread out throughout the network. This contrast between correctly and erroneously segmented collocates makes sense since word vectors are known to capture the extent to which words are replaceable (i.e., second-order, or paradigmatic, similarity) [8]. Thus, the collocates appearing within the same paradigm, such as 林{同學/老師/醫生} or 三{次/位/名/天/年/秒/小時}, are expected to have high pairwise similarities. Word segmentation errors, on the other hand, distort the well-formedness of the words, which may result in noisy patterns in similarities between these anomalous words, and the patterns are likely to vary case to case for collocations retrieved with different node words.

This simple ad hoc study, which shows that erroneously word segmented collocates may have a different profile to correctly segmented collocates, thus hints at a potential direction for future research by incorporating word vector information to improve the quality of collocation extraction. In addition, this pattern of similarities between collocates, which is observed in collocations that are defined with word pairs occurring in a window size of two (bigrams), is expected to generalize to collocations defined with word pairs occurring in larger window sizes. This is because the pattern observed seems to result from the well-formedness of the collocates. As long as word segmentation errors produce nonsense collocates, this approach is likely to capture the pattern of semantic similarities between true and false collocates.

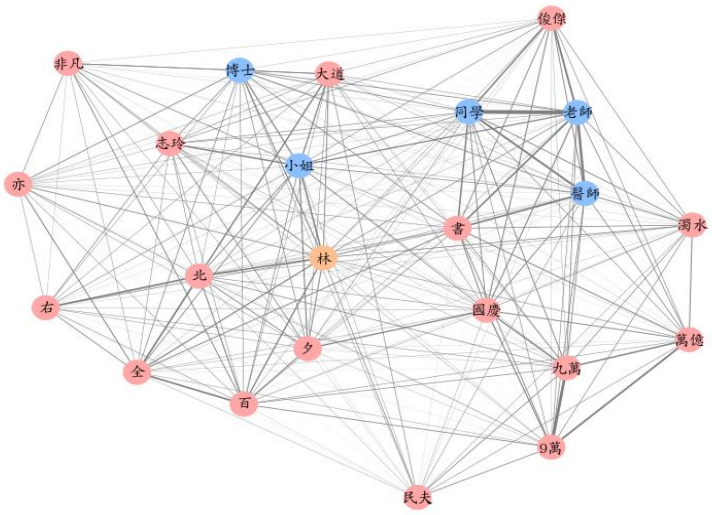

Figure 4. Semantic similarities between right collocates of 林 'Lin (family name)'. The thickness of the edge indicates the degree of similarity between a pair of nodes (edges with higher values of similarity are thicker). The orange node indicates the node word. Blue nodes indicate words of collocations that are correctly segmented, and red nodes are words resulting from word segmentation errors.

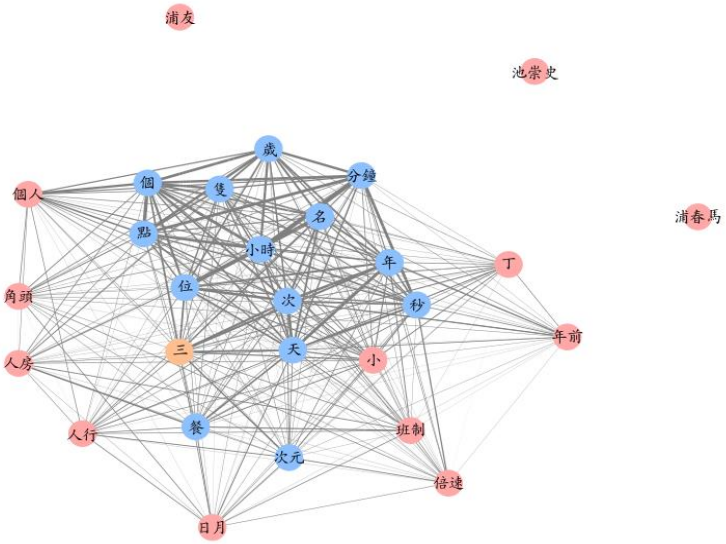

Figure 5. Semantic similarities between right collocates of 三 'three'. See the caption of Figure 4 for the meaning of the edges and the node colors. The nodes 浦友, 池崇史, 浦春馬 are isolated (i.e., similarities with other words cannot be measured) because word vectors of these words cannot be constructed due to the absence of necessary subword information in FastText pre-trained model.

5 Conclusion

Word segmentation is an important step in the NLP pipeline for Chinese, as the result of word segmentation largely influences the downstream tasks in the pipeline, such as PoS tagging, NER, and collocation extraction. In addition, with the prevalence of the internet, large corpora are constructed from texts collected from the web. With automatic word segmentation and PoS tagging performed on such large corpora, it is nearly impossible for manual checking on the correctness of such results. Thus, it is crucial to explore ways to mitigate the impacts of erroneous results stemming from such automatic tasks on downstream tasks.

In this paper, we investigated methods for improving the results of collocation extraction in automatically word segmented Chinese corpora, which suffers from retrieving false collocations resulting from word segmentation errors. A simple model, which combines several association measures in a linear fashion, are explored. Experiments with the simple linear model show that this model could not capture the necessary patterns to distinguish correctly word segmented collocations from erroneously segmented ones, as in most cases, the model performed worse than the association measures constituting the model. Facing this null result, we explore the potential for word vectors to capture the patterns of word segmentation errors in a list of collocations. An ad hoc case study of two lists of collocations shows that, in a list of collocations retrieved with a node word, correctly word segmented collocates are much more similar to each other in terms of semantic similarities computed from word vectors compared to erroneously segmented collocates. In future work, a study that investigates formal methods of incorporating word vector information to mitigate impacts of word segmentation errors on collocation extraction is suggested.

References

[1] S. Evert, “Corpora and collocations,” in *Corpus linguistics. An international handbook*, vol. 2, A. Lüdeling and M. Kytö, Eds. Berlin: Walter de Gruyter, 2008, pp. 1212–1248.

[2] S.-K. Hsieh, “Why chinese web-as-corpus is wacky? Or: How big data is killing chinese corpus linguistics,” in *Proceedings of the ninth international conference on language resources and evaluation (LREC’14)*, 2014, pp. 2386–2389.

[3] A. Kilgarriff *et al.*, “The Sketch Engine: ten years on,” *Lexicography*, vol. 1, no. 1, pp. 7–36, Jul. 2014.

[4] S. Th. Gries, “50-something years of work on collocations: What is or should be next …,” *International Journal of Corpus Linguistics*, vol. 18, no. 1. John Benjamins, pp.

137–166, 2013.

[5] C. C. Sun, P. Hendrix, J. Ma, and R. H. Baayen, “Chinese lexical database (CLD),” *Behavior Research Methods*, vol. 50, no. 6, pp. 2606–2629, Dec. 2018.

[6] C. Huang and K.-J. Chen, *Academia Sinica Balanced Corpus*. 1998.

[7] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” 2018.

[8] O. Levy, Y. Goldberg, and I. Dagan, “Improving distributional similarity with lessons learned from word embeddings,” *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[9] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, Feb. 2012.