Detecting Word Usage Errors in Chinese Sentences for Learning Chinese as a Foreign Language

Yow-Ting Shiue and Hsin-Hsi Chen

Department of Computer Science and Information Engineering, National Taiwan University

No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan

E-mail: orina1123@gmail.com; hhchen@ntu.edu.tw

Abstract

Automated grammatical error detection, which helps users improve their writing, is an important application in NLP. Recently more and more people are learning Chinese, and an automated error detection system can be helpful for the learners. This paper proposes n-gram features, dependency count features, dependency bigram features, and single-character features to determine if a Chinese sentence contains word usage errors, in which a word is written as a wrong form or the word selection is inappropriate. With marking potential errors on the level of sentence segments, typically delimited by punctuation marks, the learner can try to correct the problems without the assistant of a language teacher. Experiments on the HSK corpus show that the classifier combining all sets of features achieves an accuracy of 0.8423. By utilizing certain combination of the sets of features, we can construct a system that favours precision or recall. The best precision we achieve is 0.9536, indicating that our system is reliable and seldom produces misleading results.

Keywords: Grammatical error detection; HSK corpus; Second language learning.

1. Introduction

Recently, more and more people select Chinese as their second language. Developing grammatical error detection and correction tools for Chinese language learners is indispensable. The flexibility of the Chinese language makes error detection more challenging than other languages. According to the analysis on the HSK dynamic composition corpus created by Beijing Language and Culture University, word usage error (WUE) with error tag CC, is the most frequent type of error at the lexical level.[1] In the HSK corpus, the CC type errors are further divided into four major subtypes. The descriptions of the subtypes are shown as follows, each in terms of a pair (misused form, correct form).[2]

(1) Character disorder in a word, e.g., (先首, 首先) (first of all) and (眾所知周, 眾所周知) (as we all know).

(2) Incorrect selection of a word, e.g., 雖然現在還沒有(實踐, 實現), … (while it is not yet implemented, …).

(3) Non-existent word, e.g., (農作品, 農產品) (agricultural product).

(4) Word collocation error, e.g., 最好的辦法是兩個都(走去, 保持)平衡 (The best way is to keep both balance).

In Chinese, segmentation is a fundamental problem. When characters in a word are disordered, e.g., “首” and “先” are exchanged in the word “首先”, the resulting form may not be a word. Thus, they may be segmented into a sequence of characters by a dictionary-based segmentation system. In word collocation error, both the misused form and the correct form are real words, but the latter collocates with other words in the given sentence and the former does not.

CC (1) and (3) are similar in that the misused forms are not in a dictionary. Likewise, CC (2) and (4) are similar. The misused forms are in a dictionary. In this paper, CC (1) along with (3), and CC (2) along with (4) are merged into morphological errors (W) and usage errors (U), respectively. This paper deals with the detection of WUE in Chinese sentences. Given a Chinese sentence, we tell if it contains any WUE.

This paper is organized as follows. Section 2 surveys the related work. Section 3 describes the dataset used in this study. Section 4 proposes the classifiers and features for MUE detection. Section 5 shows and discusses the experimental results. Section 6 concludes the work.

2. Related Work

Leacock et al. (2014) give a comprehensive study of grammatical error correction (GEC). They pointed out the errors made by non-native language learners are quite different from those by native language learners. Training data should come from non-native language learners to capture the phenomena of grammatical errors.

To measure the performance of GEC systems, several shared tasks have been organized in recent years for English, including HOO 2011 (Dale and Kilgarriff, 2011), HOO 2012 (Dale et al., 2012), CoNLL 2013 (Ng et al., 2013) and CoNLL 2014 (Ng et al., 2014). Different types of grammatical errors were investigated. Language models, machine learning-based classifiers, rule-based classifiers, and machine translation models have been explored.

In Chinese, spelling check evaluations were held at SIGHAN 2013 Bake-off (Wu et al., 2013) and SIGHAN 2014 Bake-off (Yu et al., 2014). Yu, Lee and Chang (2014) extended the evaluation to Chinese grammatical error diagnosis. Four kinds of grammatical errors, i.e., redundant word, missing word, word disorder, and word

[1] http://202.112.195.192:8060/hsk/tongji2.asp

[2] http://202.112.195.192:8060/hsk/help2.asp

selection, were defined.

Yu and Chen (2012) adopted the HSK corpus to study word ordering errors (WOEs) in Chinese, and proposed syntactic features, web corpus features and perturbation features for WOE detection. Cheng, Yu and Chen (2014) identified sentence segments containing WOEs, and further recommended the candidates with correct word orderings by using ranking SVM.

Different from the above researches, this paper focuses on Chinese word usage error detection. WUE appears at lexical level rather than character level in spelling checking. Moreover, this task is also different from Chinese diagnosis task defined in Yu, Lee and Chang (2014). To the best of our knowledge, it is the first attempt to detect WUEs in Chinese sentences.

3. Data Preparation

Both wrong and correct sentences are selected from the HSK corpus. Sentences are determined by punctuation marks "？", "！", and "。". The sentences which do not contain any error tags are regarded as correct ones. To simplify the problem, we convert a sentence with *n* errors into *n* sentences, each of which with only one error. That is, the following sentence, which contains three errors,

○ ○ E1 ○ ○ ○ ○ ○ ○ E2 ○ ○ ○ ○ E3 ○

will be converted to three sentences like:

○ ○ E1 ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
○ ○ ○ ○ ○ ○ ○ ○ ○ E2 ○ ○ ○ ○ ○ ○
○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ E3 ○

In Chinese, a sentence is usually composed of several segments separated by comma "，". For example, the following sentence is composed of three segments:

如果我當推銷員的話，為了早點兒習慣，打算盡可能努力。

The longer a sentence is, the more easily a learner makes grammatical errors. If we mark the whole sentence as "wrong" only because one of the segments contains WUE, the benefit to the learner will be limited. Therefore, we consider a segment as a unit of WUE detection.

We adopt the ICTCLAS Chinese Word Segmentation System[3] to perform word segmentation, and define the length of a sentence to be the number of words in the segmentation result. After excluding short segments of length less than 5, we get 63,612 correct segments and 17,324 segments with WUEs. Table 1 shows that learners make usage errors more often than writing a word as a wrong form.

Finally, we randomly select 15,000 correct and WUE segments respectively, and combine them into a dataset with 30,000 segments in total. This dataset is called "15000s".

	W Error	U Error
HSK WUE	(1) & (3)	(2) & (4)
#segments	4,010	13,314

Table 1: Distribution of WUEs.

4. WUE Detection

3.1. Classification Models

Several properties of the Chinese WUE detection problem are worth noticing. W-type errors can be identified almost at first sight, but for U-type errors, even native speakers may have to "think twice". For example, to determine if "體會" (realize) is a misuse of the word "體驗" (experience) in the sentence "親身體會了一場永遠難忘的電單車意外" (personally realize an accident which was never forgotten), we have to consider its collocation with "意外" (accident). On the other hand, any sentences using a non-existent word such as "農作品" can be detected solely by its extremely low frequency in a Chinese corpus.

In this paper, WUE detection is formulated as a binary classification problem. Given a Chinese segment, we tell if there is a WUE in the segment. Decision tree, random forest, and support vector machine with RBF kernel are explored.

3.2. Google n-gram features

We adopt the Chinese version of Google Web 5-gram (Liu et al., 2010) to generate n-gram features. For every word sequence of length n (n=2, 3, 4, 5) in a segment, we calculate the n-gram probability by Maximum Likelihood Estimation (MLE). Taking trigram for example, the probability is defined as follows.

$$\mathrm{p}(w_i|w_{i-2},w_{i-1}) = \frac{c(w_{i-2},w_{i-1},w_i)}{c(w_{i-2},w_{i-1})} \quad (1)$$

where c(•) is the frequency of the word sequence in the Google Web 5-gram corpus. We combine the sum of n-gram probabilities with segment length (s_len). All n-gram features are concatenated into a feature vector G = (g_2, g_3, g_4, g_5), where

$$\mathrm{g_n} = \sum_{\mathrm{i=n}}^{\mathrm{L}} \mathrm{p}(\mathrm{w_i}|\mathrm{w_{i-n+1}},\dots,\mathrm{w_{i-1}}) \quad (2)$$

3.3. Dependency count feature

Errors in a sentence affect the result of segmentation and parsing. We postulate that there is a certain distribution of dependency counts in normal sentences, and the counts of error sentences deviate from the distribution. Therefore, we take the count of each type of dependency of Stanford Parser (Chang et al., 2009) output as a set of features. For each dependency, there are two types of "count": (1) internal count, which counts the occurrence if the two words are both in the target segment, and (2) external count, which counts as long as one of the words is in the target segment.

There are 45 types of dependency in our dataset, and we also include total internal and external counts. The result feature vector **D** has 92 dimensions. We also combine them with segment length (s_len).

3.4. Dependency bigram feature

Long distance dependency is common in Chinese sentences. In the example, "親身/體會/了/一場/永遠/難忘/的/電單車/意外", "意外" is the object of "體會", but

[3] http://ictclas.nlpir.org/

	Model: **support vector machine**				Model: **decision tree**			
Features	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
G	0.7706	0.7650	**0.7813**	0.7731	**0.8333**	0.9532	0.7011	0.8079
D	0.6586	0.6771	0.6068	0.6400	0.6242	0.6248	0.6228	0.6238
B	0.6102	0.6226	0.5595	0.5894	0.6148	0.6094	0.6447	0.6266
S	0.6217	0.6435	0.5456	0.5905	0.6196	0.6453	0.5314	0.5828
DB	0.6534	0.6702	0.6041	0.6354	0.6231	0.6272	0.6114	0.6192
GD	0.7638	0.7710	0.7507	0.7607	**0.8325**	0.9513	0.7009	0.8071
GB	0.7550	0.7453	0.7749	0.7598	**0.8316**	**0.9536**	0.6972	0.8055
GS	**0.7858**	**0.7885**	0.7810	**0.7874**	**0.8341**	0.9503	**0.7050**	**0.8095**
GDBS	0.7716	0.7765	0.7628	0.7696	**0.8332**	0.9486	0.7046	0.8086

Table 2: Performance of support vector machine and decision tree on 15000s dataset.

there are 6 words in-between, falling outside the range of n-gram features. To cope with the problem, we generate dependency bigrams. The above sentence contains dependencies such as nsubj(體會-2, 親身-1) and dobj(體會-2, 意外-9). We compose the two words in each dependency, i.e., (親身, 體會) and (體會, 意外), query the Google n-gram corpus, and calculate the bigram probabilities. Since the collocating behavior may vary with dependency type, we sum the bigram probabilities of each type respectively. Similar to Section 3.3, we calculate both internal sum and external sum. This set of features, denoted as feature vector B, has 92 dimensions, and segment length is also considered.

3.5. Single character feature

A non-existent Chinese word (W-type error) is usually separated into several single-character words after segmentation, so the occurrence of single-character words is an important feature for the segments with WUEs. We define the following features:

(1) Number of contiguous single-character blocks (seg_cnt)

(2) Number of contiguous single-character blocks with length no less than 2 (len2above_seg_cnt)

(3) Length of the maximum contiguous single-character block (max_seg_len)

(4) Sum of the lengths of all contiguous single-character blocks (sum_seg_len)

Consider the following segment as an example:

而且 我 認為 貴 公司 是 我國 最 大 的
(…, and I thought that your company is the biggest in our country.)

The feature values are 4, 1, 3, and 6, respectively. The proposed features are concatenated into a vector **S** and segment length is also considered.

5. Experimental Results and Analysis

4.1. Results on the 15000s dataset

The performance of our classifiers on 15000s dataset is shown in Table 2. Three classification models are adopted: decision tree, support vector machine, and random forest. We use the implementation in the scikit-learn[4] Python library. For support vector machines, we scale the feature values to unit variance. Since we use a balanced dataset, the baseline accuracy is simply 50%. We report the best accuracy among various parameter settings. All accuracy values are the average of 10-fold cross validation.

For every set of features, decision trees outperformed support vector machines, showing that decision tree is a better model for WUE detection on the features we proposed. Google n-gram (G) is the most effective feature in decision tree, while accuracies of the other three individual features are only about 0.60. GS, the best feature combination in decision tree, has F1 of 0.8095.

The feature combinations with accuracy higher than 0.83 are further used in the experiments of random forest, as shown in Table 3. The best accuracy among various parameter settings is 0.8423 for the combination of all 4 sets of features. We compare the best model with the other models resulting from decision tree and random forest. All p values are less than 0.05 with the paired t-test, so the improvement is significant.

	Model: **random forest**			
Feature	Accuracy	Precision	Recall	F1
G	0.8324	**0.9496**	0.7021	0.8073
GD	0.8371	0.9023	0.7560	0.8227
GB	0.8386	0.9251	0.7369	0.8203
GS	0.8391	0.9443	0.7206	0.8174
GDBS	**0.8423**	0.8998	**0.7705**	**0.8301**

Table 3: Performance of random forest on 15000s dataset.

Note also that our methods can achieve high precision, which means that the segments marked by our system are very likely to contain true error, so the learners are seldom misled. The decision tree model with GB features provides the best precision, 0.9536. By adopting the random forest model, the precision slightly drops, but more errors are detected, which results in the increase of recall and the overall accuracy. The Google n-gram features (G) in general tend to facilitate more accurate detection. Other set of features help discover more errors, but might have a cost of lower precision. By utilizing

[4] http://scikit-learn.org/stable/

suitable model and certain combination of the sets of features, we can construct a system that favors precision or recall, according to specific application purposes.

4.2. Results on different subtypes of WUEs

To test the performance of our system on different error subtypes, we take 4,000 segments from each subtype and combine them with 4,000 correct segments respectively. The generated dataset, called 4000s_W and 4000s_U, contains 8,000 segments respectively. The experimental results of the two datasets are shown in Tables 4 and 5. Detecting U-type errors are generally harder than detecting W-type errors. Feature combinations with higher accuracy are also used in the experiments of random forest. The best accuracy of detecting W-type and U-type errors is 0.8421 and 0.7083 respectively.

Model: **decision tree**				
Feature	Accuracy	Precision	Recall	F1
G	**0.8143**	0.8776	0.7322	0.7983
D	0.6438	0.6657	0.5810	0.6205
B	0.6418	0.6706	0.5638	0.6126
S	0.6199	0.6362	0.5678	0.6001
DB	0.6643	0.6642	0.6648	0.6645
GD	**0.8163**	**0.9035**	0.7085	0.7942
GB	0.8091	0.8790	0.7188	0.7909
GS	**0.8205**	0.8882	**0.7345**	**0.8041**
GDBS	**0.8111**	0.8699	**0.7345**	0.7965
Model: **random forest**				
Feature	Accuracy	Precision	Recall	F1
G	0.8156	**0.9117**	0.6993	0.7915
GD	0.8286	0.8602	0.7853	0.8211
GS	0.8271	0.8801	0.7578	0.8144
GDBS	**0.8421**	0.8610	**0.8165**	**0.8382**

Table 4: Performance on 4000s_W dataset.

Model: **decision tree**				
Feature	Accuracy	Precision	Recall	F1
G	0.6299	0.6128	**0.7143**	**0.6597**
D	0.6234	0.6283	0.6078	0.6179
B	0.6225	0.6481	0.5588	0.6001
S	0.6081	0.6212	0.5573	0.5875
DB	0.6236	0.6478	0.5485	0.5940
GD	**0.6558**	**0.6671**	0.6273	0.6466
GB	**0.6414**	0.6484	0.6350	0.6416
GS	0.6331	0.6345	0.6408	0.6376
GDBS	**0.6556**	0.6668	0.6278	0.6467
Model: **random forest**				
Feature	Accuracy	Precision	Recall	F1
GD	0.7024	0.6975	0.7153	0.7063
GB	0.6989	0.6970	0.7040	0.7005
GDBS	**0.7083**	**0.7039**	**0.7195**	**0.7116**

Table 5: Performance on 4000s_U dataset.

By evaluating on the error subtypes separately, we can also observe the function of different sets of features. The single character features (S) is designed for W-type errors and is less helpful on the U-type dataset in the experiments. For the U-type dataset, the useful features except G are those derived from dependency relations, which have the potential to reveal long distance collocations.

Figure 1 further shows the relationship between the best accuracy and the dataset size in the experiments of random forest. With the largest dataset, the accuracy for U-type errors reaches 0.8521. Due to the amount of available data for W-type errors, only two datasets are generated. We can observe that accuracy of the two sub-types both increases with the amount of training data. To reach the same level of accuracy, more training data are needed for U-type errors.

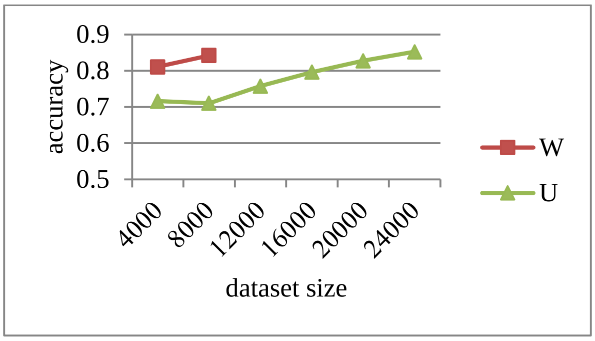

Figure 1: Accuracy vs. dataset size.

6. Conclusion

We address the Chinese word usage error detection problem with n-gram features, dependency count features, dependency bigram features, and single-character features. The best model achieves accuracy of 0.8423, precision of 0.8998, recall of 0.7705, and F1 of 0.8301 with random forest in the 15000s dataset. By utilizing suitable model and combination of features, we can also construct a word usage error system that favors precision, up to 0.9536. The single character features in combination with n-gram features are effective for morphological errors (W), while dependency-derived features better capture usage errors (U). The detection of usage error is harder and need more training data. In the future, we will narrow down the detection scope from segment level to word level, and propose candidates to correct WUEs.

7. Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-103-2815-C-002-089-E, MOST-102-2221-E-002-103-MY3, and MOST-104-2221-E-002-061-MY3.

8. References

Chang, P.C., Tseng, H., Jurafsky, D. and Manning, C.D. (2009). Discriminative Re-ordering with Chinese Grammatical Relations Features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pp. 51–59.

Cheng, S.M., Yu, C.H., and Chen, H.H. (2014). Chinese Word Ordering Errors Detection and Correction for Non-Native Chinese Language Learners. In *Proceedings of the 25th International Conference on Computational Linguistics*, pp. 279–289.

Dale, R., Anisimoff, Il. and Narroway, G. (2012). HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pp. 54–62.

Dale, R., and Kilgarriff, A. (2011). Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pp. 242-249.

Liu, F., Yang, M. and Lin, D. (2010). Chinese Web 5-gram Version 1. Linguistic Data Consortium, Philadelphia.

Leacock, C., Chodorow, M., Gamon, M. and Tetreault, J. (2014). Automated Grammatical Error Detection for Language Learners. 2nd Edition. Morgan and Claypool Publishers.

Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., and Bryant. C. (2014). The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14.

Wu, S. H., Liu, C. L., and Lee, L. H. (2013). Chinese spelling check evaluation at SIGHAN Bake-off 2013. In *Proceedings of the 7th SIGHAN Workshop on Chinese Language Processing*, pp. 35-42.

Yu, C. H., and Chen, H. H. (2012). Detecting Word Ordering Errors in Chinese Sentences for Learning Chinese as a Foreign Language. In *Proceedings of COLING 2012: Technical Papers*, pp. 3003–3018.

Yu, L.C., Lee, L.H. and Chang, L.P. (2014). Overview of Grammatical Error Diagnosis for Learning Chinese as a Foreign Language. In *Proceedings ICCE 2014 Workshop of Natural Language Processing Techniques for Educational Applications*, pp. 42–47.

Yu, L. C., Lee, L. H., Tseng, Y. H., and Chen, H. H. (2014). Overview of SIGHAN 2014 Bake-off for Chinese spelling check. In *Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 126–132.