KUNLP Grammatical Error Correction System For CoNLL-2013 Shared Task

Bong-Jun Yi, Ho-Chang Lee, and Hae-Chang Rim
Department of Computer and Radio Communications Engineering
Korea University
Anam-dong 5-ga, Seongbuk-gu, Seoul, South Korea
{bjyi,hclee,rim}@nlp.korea.ac.kr

Abstract

This paper describes an English grammatical error correction system for CoNLL-2013 shared task. Error types covered by our system are article/determiner, preposition, and noun number agreement. This work is our first attempt on grammatical error correction research. In this work, we only focus on reimplementing the techniques presented before and optimizing the performance. As a result of the implementation, our system's final F1-score by m2 scorer is 0.1282 in our internal test set.

1 Introduction

As the number of English learners is increasing world widely, the research topic of automated grammar error correction is lively discussed. However, automated grammar error correction is a very difficult field and the result is not satisfactory. Therefore, the shared task about English error correction has been annually held and many researchers are trying to solve this problem.

Helping Our Own (HOO) 2011 is a pilot shared task for automated correction of errors in non-native English speakers' papers. The shared task evaluates the performance of detection, recognition, correction on thirteen types of English grammatical errors by using F1-score. Because each error type has different characteristics, they have to use different approaches to correct appropriate error types.

In *HOO 2012*, only two types of errors, preposition and determiner were handled. This shared task also evaluated the performance of detection, recognition, correction by using F1-score. The best result of the preposition error correction is 0.2371 in F1-score and the determiner error correction is 0.3460 in F1-score. These are remarkable achievement.

This year *CoNLL 2013* shared task covers five types of errors based on the result of *HOO 2012*. These error types are determiner, preposition, noun number, verb form, and subject-verb agreement. Because of the limited amount of time and manpower, we only focus on preposition, determiner and noun number.

2 Previous Works

Most methods for grammar error correction have tried to correct one type of errors. Researchers have never attempted to correct different types of errors at the same time.

In this work, we try to solve the error correction problem based on the previous research presenting good performance.

First, the preposition error correction is based on (Han et al., 2010). They tried to correct the most commonly used 10 preposition errors based on the classification approach. 10 prepositions are *about, at, by, for, from, in, of, on, to, with.* They have implemented 11-way classifier to output 11 types of proper word(10 prepositions + 'NULL') for 11 types of source words. This work assumes that three kinds of corrections exist. If 'NULL' is taken as input and some preposition is produced, it is omission. If some preposition is taken as input and another preposition is produced, it is replacement. If some preposition is taken as input and 'NULL' is produced, it is commission. In the case of replacement, correction precision is 0.817 and recall is 0.132. Furthermore, they reported that the performance is much better when they train the model with well edited error tagged corpus.

(Felice and Pulman, 2008) also used a method based on classification. It is, nevertheless, unusual that they did not use error tagged learner's corpus but error free British National Corpus. Without using an error tagged corpus, they have achieved 51.5% accuracy for error correction.

To improve low recall of Han's method, to con-

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 123–127,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

struct large training data is the best way. However, it is very costly and hard work to obtain well edited error tagged corpus. By the way, error free corpus like news articles is relatively easy to acquire. We plan to utilize large error free corpus as the training data to overcome the problem of low recall. That plan motivated by Felice's work has not been tried on the proposed system. We will attempt to reimplement the system by utilizing the error free corpus in the near future.

3 System Description

Our system is composed of three components, preposition error corrector, article error corrector, and noun number error corrector. In this work, we do not consider complex cases of grammar errors, thus we assume that the order of correction does not influence the result of correction. And all components are based on the machine learning method.

3.1 Preposition Error Correction

In the training corpus, there are more article errors than preposition errors in number. However, the preposition error correction is much more difficult and the performance of correction is worse than the article error correction.

We select preposition error candidates for replacement or commission or omission as follows.

- Replacement or Commission
  - Preposition : tagged as 'IN' or 'TO' and dependency relation with its parent(DPREL) is identified as a 'prep'
- Omission
  - In front of a noun phrase : the preceding word of the noun phrase is not preposition
  - In front of a verb phrase : the preceding word of the verb phrase including 'VBG'(verb, gerund/present participle) is not preposition

As described above, we use all words(preposition and 'NULL' when omission) in that place as source word for preposition correction.

We have implemented only one classifier that takes a source word as input and produces corrected preposition or 'NULL' as output. We use the part of feature set(Table 1) proposed by (Han et al., 2010) for learning. They are presented in the experiment part.

No.	Feature Name	Description
1	s	the source word
2	wd-1	the word preceding s
3	wd+1	the word following s
4	wd-1,2_s	the two words preceding s and s
5	s_wd+1,2	s and the two words following s
6	3GRAM	the trigram, wd-1, s, wd+1
7	5GRAM	the five-gram, wd-2, wd-1, s, wd+1, wd+2
8	MOD	the lexical head which the preposition modifies
9	ARG	the lexical head of the preposition argument
10	MOD_ARG	MOD and ARG
11	MOD_s	MOD and s
12	s_ARG	s and ARG
13	MOD_s_ARG	MOD, s, ARG
14	MODt_ARGt	POS tags of MOD and ARG
15	MODt_s	the POS tag of MOD and s
16	s_ARGt	s and the POS tag of ARG
17	MODt_s_ARGt	MODt and s and ARGt
18	TRIGRAMt	POS trigram
19	wd_L	3 words preceding s
20	wd_R	3 words following s

Table 1: Set of features proposed by (Han et al., 2010)

Each feature represents the word itself in the Han's work. However, the same word can be extracted again as a different kind of features. In order to distinguish the same word used for the different features, we attach the feature name to the word as postfix. This naming convention can make the feature sparse, but increase the discrimination power and improve the performance of the classifier. In our experiment, we have tested the system with two different sets of features(i.e. raw word and with feature names).

3.2 Article Error Correction

We have implemented the article error corrector just like the preposition error corrector. When we experiment the pilot article correction system just like the preposition correction system, it shows a good performance unexpectedly. There is a little difference in presenting set of features. In preposition error corrector, we add postfix to the set of feature to keep sort of features(e.g. word 'in' as source word feature, postfix is 'S', final feature is 'in_S'). This method gives more discrimination ability to the classifier. But in case of article, using raw word lead to a better result.

3.3 Noun Number Error Correction

Noun number error indicates improper use of singular or plural form of nouns. For example, the singular form 'problem' should be corrected to the plural form 'problems' in the following sentence.

"They are educational and resource problem."

As far as we know, there have been few attempts to correct noun number agreement errors. In this shared task, we propose a novel noun number agreement correction system based on a machine learning method trained with basic features.

In order to extract nouns from the input sentences, we parse the sentence and extract the last noun in every noun phrase for the error correction candidates. If there is a coordinating conjunction in the noun phrase, we split the noun phrase into two parts and extract two candidates.

[S [NP Relevant **information**] [VP are [ADJP readily ROOT available [PP on [NP [NP the **internet**] and [NP **article**] [PP in [NP **magazines** and **newspapers**] .]]]

Figure 1: Extracting candidates for error correction of noun number(candidates are indicated by bold)

Figure 1 shows the example of selecting candidates for the error correction of noun number. We classify a noun into four classes using features of Table 2 based on the machine learning method. Four classes are NN(plural noun), NNP(plural proper noun), NNS(singular noun), and NNPS(singular proper noun). It is based on the observation that the common noun and the proper noun have many different characteristic. The set of features used for learning is shown in Table 2.

No.	Feature Name	Description
1	p	POS tag for the source word
2	s	the source word
3	DET	the determiner of the noun argument
4	CD	boolean value whether the cardinal number exists
5	UCNT	boolean value whether the noun is uncountable
6	MOD	the lexical head which the noun modifies
7	MMOD	the lexical head which MOD modifies
8	ARG	the lexical head of the noun argument
9	AARG	the lexical head of the ARG argument
10	MOD_s_ARG	MOD, s, and ARG
11	MODt	POS tag for MOD
12	MMODt	POS tag for MMOD
13	ARGt	POS tag for ARG
14	AARGt	POS tag for AARG
15	MODt_p_ARGt	MODt, p, and ARGt
16	MMOD_MOD_s	MMOD, MOD, and s
17	s_ARG_AARG	s, ARG, and AARG
18	MOD_s_ARG	MOD, s, and ARG
19	MM_M_s_A_AA	MMOD, MOD, s, ARG, and AARG
20	MMODt_MODt_p	MMODt, MODt, and p
21	p_ARGt_AARGt	p, ARGt, and AARGt
22	MODt_p_ARGt	MODt, p, and ARGt
23	MMt_Mt_s_At_AAt	MMODt, MODt, s, ARGt, and AARGt

Table 2: Set of features for learning noun number error correction

4 Experiments

4.1 Corpus

We use only NUS Corpus of Leaner English(NUCLE)((Dahlmeier, 2013)) provided from CoNLL 2013 shared task. We construct the development set with first sentences for every 10 sentence and the test set with second sentences and the training set with the rest of sentences. The system is trained to learn error correction with the training set and optimized with the development set and finally evaluated with the test set.

4.2 Preposition Correction Experiment

Table 1 shows 20 types of features used by (Han et al., 2010). We have found that the features consist of various types and the learning world be disturbed by too many features. In our experiment,

Number of feature		20	18	9
Raw Word	Precision	0.0571	0.1194	0.0196
	Recall	0.0402	0.0402	0.1256
	F1-score	0.0472	0.0602	0.0339
With Feature Name	Precision	0.1034	0.1750	0.0208
	Recall	0.0302	0.0352	0.1307
	F1-score	0.0467	0.0586	0.0359

Table 3: The result of preposition error correction

we exclude wd_L(19), wd_R(20) and employ 18 kinds of features.

We will try to train the correction model by using large amount of error free corpus in order to overcome the problem of low recall. To parse large corpus is very time consuming task. So, in this experiment, we select 9 features which can be extracted without parsing, and test the possibility of using 9 features by training and testing the correction model.

We have performed two different experiments. In the first experiment, we have used the word itself as a feature. In the tables 3˜5, "Raw Word" represents the case when we use just the word itself. In the second experiment, we have used the feature name as the postfix of the feature. In the tables 3˜5, "With Feature Name" represents the case when we attach the feature name to the feature and use it as a feature. For all experiments, we have tried to differentiate the number of features. 20 features are same as Han's work. 18 features are the case when we exclude 2 features(i.e. wd_L(19), wd_R(20)). 9 features are the case when we use only features which do not require parsing.

We have experimented with Maximum Entropy learning method, and fixed the iteration number to 200. Table 3 shows that the precision has highly increased although the recall has decreased when we add the feature name to the set of features used for learning.

When we use 18 features except wd_L(3 words preceding s) and wd_R(3 words following s), the error correction system achieves the best performance. According to the experimental result, we can achieve the better result when we use 18 features and the raw word. But we select final option using 18 features and the word with feature name because of optimization strategies that improve the precision.

Number of feature		20	18	9
Raw Word	Precision	0.1827	0.3176	0.0914
	Recall	0.1123	0.1264	0.1139
	F1-score	0.1391	0.1808	0.1014
With Feature Name	Precision	0.1942	0.3174	0.1059
	Recall	0.1154	0.1139	0.1061
	F1-score	0.1448	0.1676	0.1060

Table 4: The result of article error correction

Kinds of feature		Basic	Basic& Indep- endent	Basic& Com- plex
Raw Word	Precision	0.2462	0.1435	0.2469
	Recall	0.0379	0.0811	0.0540
	F1-score	0.0662	0.1036	0.0887
With Feature Name	Precision	0.2413	0.1676	0.2875
	Recall	0.0378	0.0838	0.0621
	F1-score	0.0654	0.1117	0.1022

Table 5: The result of noun number error correction

4.3 Article Correction Experiment

Table 4 shows that the feature name addition does not improve the precision in the case of article correction, and the set of 18 features achieves the best performance for article correction. Therefore, we just use raw words for features and select 18 features for article correction.

4.4 Noun Number Correction Experiment

In Table 2, features of number 1˜5 belong to the basic feature set and features of number 6˜15 belong to the independent feature set and features of number 16˜23 belong to the complex feature set. The experimental result with various combinations of feature sets shows that the set of basic and complex features achieves the best precision in spite of low recall as shown in Table 5. We use this option and experimentally select the iteration number 700.

5 Conclusions

We develop a grammatical error correction system which can recognize and correct preposition, article, and noun number errors. In this experiment, we have found out the set of good features for preposition and article error correction, and proposed a novel noun number error correction technique based on the machine learning method. For the future work, we will try to utilize large amount of external resources such as well written error free corpus.

References

Daniel Dahlmeier, Hwee Tou Ng, Siew Mei Wu 2013. *Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English*, *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)*, Atlanta, Georgia, USA.

Rachele De Felice and Stephen G. Pulman 2008. *A classifier-based approach to preposition and determiner error correction in L2 English*, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 169–176, Manchester, UK

Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, Jin-Young Ha 2010. *Using an Error-Annotated Learner Corpus to Develop an ESL/EFL Error Correction System*, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, 763–770, Malta