A Hybrid Model for Grammatical Error Correction

Yang Xiang, Bo Yuan, Yaoyun Zhang[*], Xiaolong Wang[†],
Wen Zheng, Chongqiang Wei
Intelligent Computing Research Center, Key Laboratory of Network Oriented Intelligent
Computation, Computer Science and technology Department,
Harbin Institute of Technology Shenzhen Graduate School,
Shenzhen, Guangdong, 518055, P.R. China
{windseedxy, yuanbo.hitsz, xiaoni5122, zhengwen379, weichongqiang}@gmail.com
wangxl@insun.hit.edu.cn[†]

Abstract

This paper presents a hybrid model for the CoNLL-2013 shared task which focuses on the problem of grammatical error correction. This year's task includes determiner, preposition, noun number, verb form, and subject-verb agreement errors which is more comprehensive than previous error correction tasks. We correct these five types of errors in different modules where either machine learning based or rule-based methods are applied. Preprocessing and post-processing procedures are employed to keep idiomatic phrases from being corrected. We achieved precision of 35.65%, recall of 16.56%, F_1 of 22.61% in the official evaluation and precision of 41.75%, recall of 20.29%, F_1 of 27.3% in the revised version. Some further comparisons employing different strategies are made in our experiments.

1 Introduction

Automatic Grammatical Error Correction (GEC) for non-native English language learners has attracted more and more attention with the development of natural language processing, machine learning and big-data techniques. The CoNLL-2013 shared task focuses on the problem of GEC in five different error types including determiner, preposition, noun number, verb form, and subject-verb agreement which is more complicated and challenging than previous correction tasks. Other than most previous works which concentrate most on determiner and preposition errors, more error types introduces the possibility of correcting multiple interacting errors such as determiner vs. noun number and preposition vs. verb form.

Generally, for GEC on annotated data such as the *NUCLE* corpus (Dahlmeier et al., 2013) in this year's shared task which contains both original errors and human annotations, there are two main types of approaches. One of them is the employment of external language materials. Although there are minor differences on strategies, the main idea of this approach is to use frequencies as a filter, such as n-gram counts, and take those phrases that have relatively high frequencies as the correct ones. Typical works are shown in (Yi et al., 2008) and (Bergsma et al., 2009). Similar methods also exist in HOO shared tasks[1] such as the web 1TB n-gram features used by Dahlmeier and Ng (2012a) and the large-scale n-gram model described by Heilman et al. (2012). The other type is machine learning based approach which considers most on local context including syntactic and semantic features. Han et al. (2006) take maximum entropy as their classifier and apply some simple parameter tuning methods. Felice and Pulman (2008) present their classifier-based models together with a few representative features. Seo et al. (2012) invite a meta-learning approach and show its effectiveness. Dahlmeier and Ng (2011) introduce an alternating structure optimization based approach.

Most of the works mentioned above focus on determiner and preposition errors. Besides, Lee and Seneff (2008) propose a method to correct verb form errors through combining the features of parse trees and n-gram counts. To our knowledge, no one focused on noun form errors in specific researches.

In this paper, we propose a hybrid model to solve the problem of GEC for five error types.

[*] Corresponding author

[1] http://clt.mq.edu.au/research/projects/hoo/hoo2012

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 115–122,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

Machine learning based methods are applied to solve determiner (ArtOrDet), preposition (Prep) and noun form (Nn) problems while rule-based methods are proposed for subject-verb agreement (SVA) and verb form (Vform) problems. We treat corrections of errors in each type as individual sub problems the results of which are combined through a result combination module. Solutions on interacting error corrections were considered originally but dropped at last because of the bad effects brought about by them such as the accumulation of errors which lead to a very low performance. We perform feature selection and confidence tuning in machine learning based modules which contribute a lot to our performance. Also, pre-processing and post-processing procedures are employed to keep idiomatic phrases from being corrected.

Through experiments, we found that the result of the system was affected by many factors such as the selection of training samples and features, and the settings of confidence parameters in classifiers. Some of the factors make the whole system too sensitive that it can easily be trapped into a local optimum. Some comparisons are shown in our experiments section.

No other external language materials are included in our model except for several NLP tools which will be introduced in §**5.2**. We achieved precision of 35.65%, recall of 16.56% and F_1 of 22.61% in the official score of our submitted result. However, it was far from satisfactory mainly due to the ill settings of confidence parameters. Trying to find out a set of optimal confidence parameters, our model is able to reach an upper bound of precision of 34.23%, recall of 25.56% and F_1 of 29.27% on the official test set. For the revised version, we achieved precision of 41.75%, recall of 20.29%, and F_1 of 27.3%.

The remainder of this paper is arranged as follows. The next section introduces our system architecture. Section 3 describes machine learning based modules. Section 4 shows rule based modules. Experiments and analysis are arranged in Section 5. Finally, we give our discussion and conclusion in Section 6 and 7.

2 System Architecture

Initially, we treat errors of each type as individual sub problems. Machine learning based methods are applied to solve ArtOrDet, Prep and Nn problems where similar problem solving steps are shared: sample generation, feature extraction, training, confidence tuning in development data, and testing. We apply some hand-crafted heuristic rules in solving subject-verb agreement (SVA) and verb form (Vform) problems. Finally, results from different modules are combined together. The whole architecture of this GEC system is described in Figure 1.

A pre-processing and a post-processing filter are utilized which include filters for some idiomatic phrases extracted from the training dataset. The Frequent Pattern Growth Algorithm (FP-Growth) is widely used for frequent pattern mining in machine learning. In pre-processing, we firstly apply FP-Growth to gather the frequent items in the training set. Through some manual refinements, a few idiomatic phrases are removed from the candidate set to be corrected. In post-processing, the idiomatic phrase list is used to check whether a certain collocation is still grammatical after several corrections are performed. There are 996 idiomatic phrases in our list which is composed by mainly patterns from the training set and a series of hand-crafted ones. Typical phrases we extracted are *in general*, *have/need to be done*, *on the other hand*, *a large/big number/amount of*, *at the same time*, *in public*, etc.

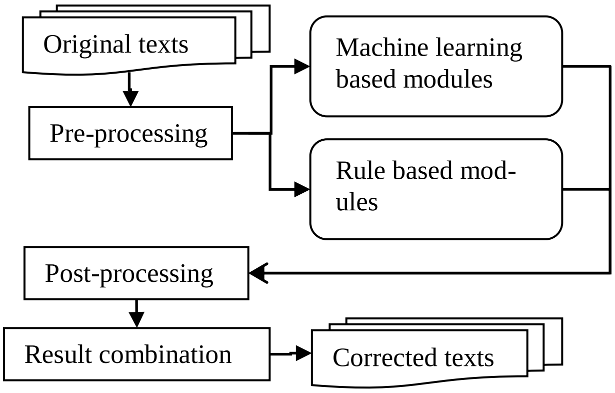


Figure 1. Architecture of our GEC system.

3 Machine Learning Based Modules

For the error types ArtOrDet, Prep and Nn, we choose machine learning based methods because we consider there is not enough evidence to directly determine which word or form to be used. Moreover, it is impossible to transfer all the cases we encounter into rules. In this section, we describe our processing ideas for each error type respectively and then specifically introduce our feature selection and confidence tuning approach.

3.1 Determiners

Determiners in the error type "ArtOrDet" contain articles *a*/*an*, *the* and other determiners such as *this*, *those*, etc. This type of error accounts for a large proportion which is of great impact on the final result. We consider only articles since the other determiners are rarely used and the usages of them are sometimes ambiguous. Like approaches described in some previous works (Dahlmeier and Ng, 2012a; Felice and Pulman, 2008), we assign three types *a*/*an*, *the* and *empty* for each article position and build a multi-class classifier.

For training, developing and testing, all noun phrases (NPs) are chosen as candidate samples to be corrected. For NPs whose articles have been annotated in the corpus, the correct ones are their target categories, and for those haven't been annotated, the target categories are their observed article types. Samples we make use of can be divided into two basic types in each category: *with* and *without* a wrong article. Two examples are shown below:

with: *~~a~~/empty big apples* ~ *empty* category
without: *the United States* ~ *the* category

For each category in *a*, *the*, and *empty*, we use the whole *with* data and take samples of *without* ones from the set of correct NPs to make up training instances of one category. The reason why we make samples of the *without* ones is for the consideration that the classifier would always predicts the observed article and never proposes any corrections if given too many *without* samples, the case of which is mentioned in (Dahlmeier and Ng, 2012a). However, we found that the ratio of *with*-*without* shows little effect in our model. The article *a* is regulated to *a* or *an* according to pronunciation.

Syntactic and semantic features are considered in feature extraction with the help of WordNet and the ".conll" file provided. We adopt syntactic features such as the surface word, phrase, part-of-speech, n-grams, constituent parse tree, dependency parse tree and headword of an NP; semantic features like noun category and hypernym. Some expand operations are also done based on them (reference to Dahlmeier and Ng, 2012a; Felice and Pulman, 2008). After feature extraction, we apply a genetic algorithm to do feature subset selection in order to reduce dimensionality and filter out noisy features which is to be described in §**3.4**.

Maximum Entropy (ME) has been proven to behave well for heterogeneous features in natural language processing tasks and we adopt it to train our model. We have also tried several other classifiers including SVM, decision tree, Naïve Bayes, and RankSVM but finally find ME performs well and stably. It provides confidence scores for each category which we will make use of downstream.

3.2 Prepositions

Preposition error correction task is similar to the previous one except the different categories and corresponding features. Since there are 36 common prepositions listed by the shared task, originally, we assign 37 types including 36 prepositions and *empty* for each preposition position and build a multi-class classifier. For training, developing and testing, each preposition as well as the empty position directly after a verb is considered as a candidate. Syntactic and semantic features extracted are similar to those in article error correction except for some specific cases for prepositions such as the verbs related to prepositions and the dependency relations. Similarly, we treat those preposition phrases *with* and *without* a certain preposition as the two types of samples in training (as described in §**3.1**). Two examples are listed below:

with: *~~on~~/in the 1860s*~ *in* category
without: *have to be done* ~ *to* category

Through statistics on the training data, we found that most prepositions have very few samples which may not contribute to the performance at all and even bring about noise when assigned to wrong categories. After several rounds of experiments, we finally adopt a classifier with seven prepositions which are frequently used in the whole corpus. They are *on*, *of*, *in*, *at*, *to*, *with* and *for*. As to the classifier, ME also outperforms the others.

3.3 Noun Form

Noun form may be interacting with determiners and verbs which may also have errors in the original text. So errors may occur in the context features extracted from the original text. However, if we use the context features that have been corrected, more errors would be employed due to the low performance of the previous steps. Through statistics, we found that co-occurrence between two types of errors such as SVA and ArtOrDet only accounts for a small proportion. After a few experiments, we decided to give up interacting errors so as to avoid accumulated errors.

This is a binary classification problem. All head nouns in NPs are considered as candidates. Each category contains *with* and *without* samples similar to the cases in §**3.1** and §**3.2**. Features are highly related to the deterministic factors for the head noun form such as the countability, WordNet type, name entity and whether there some specific dependency relations including *det*, *amod* etc.

ME also outperforms other classifiers.

3.4 Feature Selection Using Genetic Algorithm

Features we extracted are excessive and sparse after binarization. They bring noise in quality as well as complexity in computation and need to be selected a priori. In our work, it is a wrapper feature selection task. That is, we have to select a combination of features that perform well together rather than make sure each of them behaves well. This GEC task is interesting in feature selection because word surface features that are observed only once are also effective while we think that they overfit. Genetic algorithm (GA) has been proven to be useful in selecting wrapper features in classification (ElAlami, 2009; Anba-rasi et al, 2010). We used GA to select features as well as reduce feature dimensionality.

We convert the features into a binary sequence in which each character represents one dimension. Let “1” indicates that we keep this dimension while “0” means that we drop it, we use a binary sequence such as “0111000…100” to denote a combination of feature dimensions. GA functions on the feature sequences and finally decides which features should be kept. The fitness function we used is the evaluation measure F_1 described in §**5.3**.

3.5 Confidence Tuning

The Maximum Entropy classifier returns a confidence score for each category given a testing sample. However, for different samples, the distribution of predicted scores varies a lot. For some samples, the classifier may have a very high predicted score for a certain category which means the classifier is confident enough to perform this prediction. But for some other samples, two or more categories may share close scores, the case of which means the classifier hesitates when telling them apart.

We introduce a confidence tuning approach on the predicted results through a comparison between the observed category and the predicted category which is similar to the “thresholding” approach described in Tetreault and Chodorow (2008). The main idea of the confidence tuning algorithm is: the choice between *keep* and *drop* is based on the difference between the confidence scores of the predicted category and the observed category. If this difference goes beyond a threshold *t*, the prediction is kept while if it is under *t*, we won’t do any corrections. We believe this tuning strategy is especially appropriate in this task since to distinguish whether the observed category is correct or not affects a lot to the predicted result.

The confidence threshold for each category is generated through a hill climbing algorithm in the development data aimed at maximizing F_1-meaure of the result.

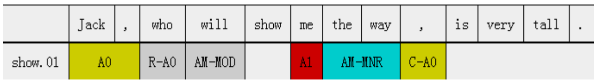
4 Rule-based Modules

A few hand-crafted rules are applied to solve the verb related corrections including SVA and Vform. In these cases, the verb form is only related to some specific features as described by Lee and Seneff (2008).

4.1 SVA

SVA (Subject-verb-agreement) is particularly related to the noun subject that a verb determines. In the dependency tree, the number of the noun which has a relation *nsubj* with the verb determines the form of this verb. Through observation, we find that the verbs to be considered in SVA contain only *be*s (including *am, is, are, was, were*) and the verbs in simple present tense whose POSs are labeled with *VBZ* (singular) or *VBP*(plural).

To pick out the noun subject is easy except for the verb that contained in a subordinate clause. We use semantic role labeling (SRL) to help solve this problem in which the coordinated can be extracted through a trace with the label “R-Argument”. The following Figure is an example generated by the SRL toolkit *mate-tools* (Bernd Bohnet, 2010)[2].



	Jack	,	who	will	show	me	the	way	,	is	very	tall	.
show.01	A0		R-A0	AM-MOD		A1	AM-MNR		C-A0				

Figure 2. SRL for the demo sentence “Jack, who will show me the way, is very tall.” The subject of the verb *show* can be traced through R-A0 -> A0.

However, the performance of this part is partly correlated with the noun form that may have errors in the original text and the wrong SRL result brought about because of wrong sentence grammars.

[2] http://code.google.com/p/mate-tools/

4.2 Verb Form

The cases are more complicated in the verb form error correction task. Modal, aspect and voice are all forms that should be considered for a verb. And sometimes, two or more forms are combined together to perform its role in a sentence. For example, in the sentence:

*He **has been working** in this position for a long time.*

The bolded verb ***has been working*** is a combination of the active voice *work*, the progressive aspect *be+VBG* and the perfect aspect *has+VBN*. It is a bit difficult for us to take all cases into consideration, so we just apply several simple rules and solve a subset of problems for this type. Some typical rules are listed below:

1. The verb that has a dependency relation *aux* to preposition *to* is modified to its base form.
2. The verb that has a dependency relation *pcomp* to preposition *by* is modified to its past form.
3. The verb related to other prepositions (except *to* and *by*) is modified to *~ing* form.
4. The verb depends on auxiliary *do* and modal verb (including its inflections and negative form) is modified to its base form.

We have also tried to use SRL and transitivity of a verb to determine the active and passive voice but it didn't work well.

5 Experiments and Analysis

5.1 Data Description

The *NUCLE* corpus introduced by NUS (National University of Singapore) contains 1414 essays written by L2 students with relatively high proficiency of English in which grammatical errors have been well annotated by native tutors. It has a small proportion of annotated errors which is much lower than other similar corpora (Dahlmeier et al., 2013). In our experiments, we divide the whole corpus into 80%, 10% and 10% for training, developing and testing. And we use 90% and 10% for training and developing for the final test.

5.2 External tools and corpora

External tools we used include WordNet (Fellbaum, 1998) for word base form and noun category generation, Morphg (Minnen et al., 2000)[3] to generate inflections of nouns and verbs, mate-tools (Bohnet, 2010) for SRL, Stanford-ner (Finkel et al., 2005)[4] for name entity extraction and Longman online dictionary[5] for generation of noun countability and verb transitivity.

We didn't employ any external corpora in our system.

5.3 Experiments

The performance of each machine learning module is affected by the selection of training samples, features and confidence tuning for the maximum entropy classifier. All these factors contribute more or less to the final performance and need to be carefully developed. In our experiments, we focus on machine learning based modules and make comparisons on sample selection, confidence tuning and feature selection and list a series of results before and after applying our strategies.

In our experiment, the performance is measured with precision, recall and F_1-measure where

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}$$

Precision is the amount of predicted corrections that are also corrected by the manual annotators divided by the whole amount of predicted corrections. Recall has the same numerator as precision while its denominator is the amount of manually corrected errors. They are in accordance with those measurements generated by the official m2scorer (Dahlmeier and Ng, 2012c) to a great extent and easily to be integrated in our program.

As we have mentioned in Section 3, we don't employ all samples but make use of all *with* (with errors and annotations) instances and sample the *without* ones (without errors) for training. And the sampling for *without* type is totally random without loss of generality. We apply the same strategy in all of these three error types (ArtOrDet, Prep and Nn) and try several ratios of *with-without* to find out whether this ratio has great impact on the final result and which ratio performs best. We use the 80%-10%-10% data (mentioned in §**5.1**) for our experiments and make comparisons of different ratios on developing data. The experimental results are described in detail in Figure 3.

Confidence tuning is applied in all these three error types which contributes most to the final performance in our model. We compare the results before and after tuning in all sample ratios

[3] http://www.informatics.sussex.ac.uk/research/groups/nlp/carroll/morph.html

[4] http://www-nlp.stanford.edu/software/CRF-NER.shtml

[5] http://www.ldoceonline.com/

that we designed and they are also depicted in Figure 3.

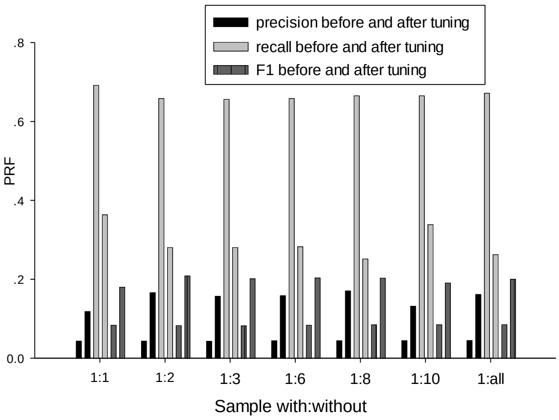


Figure 3-1. Comparisons before and after tuning in ArtOrDet. *1:all* means to use the whole *without* samples.

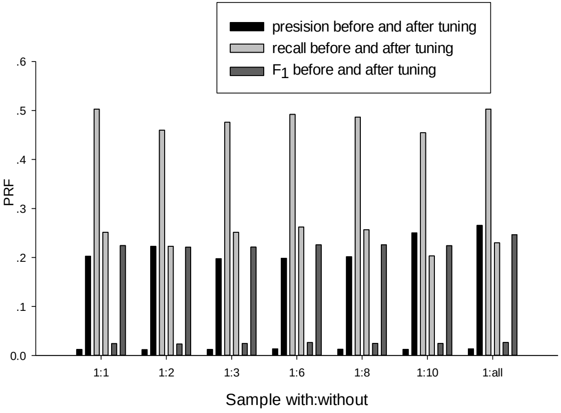


Figure 3-2. Comparisons before and after tuning in Prep.

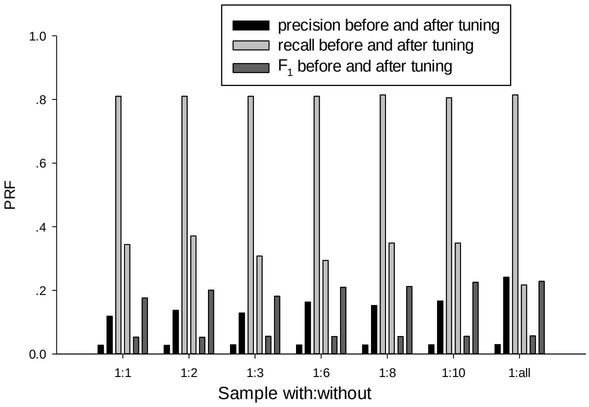


Figure 3-3. Comparisons before and after tuning in Nn.

From the three groups of data in Figure 3, we notice that the ratio of samples has little impact on F_1. This phenomenon shows that our conclusion goes against the previous work by Dahlmeier and Ng (2012a). We believe it is mainly due to our confidence tuning which makes the parameters vary much under different sample ratios, that is, if given the same parameters, the effect of sample ratio selection may become obvious. Unfortunately, we didn’t do such a systematic comparison in our work. The improvement under confidence tuning can be seen clearly in all ratios of *with-without* samples. The confidence tuning algorithm employed in our work is better than the traditional tuning methods that assign a fixed threshold for each category or for all categories (about 1%~2% better measured by F_1).

However, although we are able to pick out the training data with a high F_1 through confidence tuning for the developing data, it is difficult for us to choose a set of confidence parameters that also fits the test data well. Given several close F_1s, the numerical values of denominators and numerators which determine the precision and recall can vary a lot. For example, one set that has a high precision and low recall may share the similar F_1 with another set that has a low precision and high recall. Our work lacked of the development on how to control the number of proposed errors to make leverage on the performance between developing set and testing set. It resulted in that the developing set and the testing set were not balanced at all, and our model was not able to keep the sample distribution as the training set. This is the main factor that leads to a low performance in our submitted result which can be clearly seen in Table 1. The upper bound performance of our system achieves precision of 34.23%, recall of 25.56% and F_1 of 29.27%, in which the F_1 goes 7% beyond our submitted system. We notice that results of all metrics of the three error types where machine learning algorithms are applied improve with the simultaneous increase of numerators and denominators. This is especially noticeable in Prep.

For the other two types SVA and Vform, we just apply several heuristic rules to solve a subset of problems and the case of Vform has not been solved well such as tense and voice.

Genetic Algorithm (GA) is applied to process feature reduction and subset selection. This is done in ArtOrDet type in which we extract as many as 350,000 binary features. For error type Prep and Nn, the feature dimensionalities we constructed were not as high as that in ArtOrDet, and the improvements under GA were not obvious which we would not discuss in this work. Through experiments on a few sample ratios, we notice that feature selection using genetic algorithm is able to reduce the feature dimensionality to about 170,000 which greatly lowers down the downstream computational complexity. However, the improvement contributed by GA after confidence tuning is not obvious as that before confidence tuning. We think it is partly because of the bad initialization of GA which is to be improved in our future work. The unfixed parameters may also lead to such a result which we didn't discuss enough in our work. The comparison before and after GA is described in Figure 4.

	Our submission%	Upper bound%
P(Det)	41.38(168/406)	36.44(254/697)
R(Det)	24.35(168/690)	36.81(254/690)
F_1(Det)	30.66	36.63
P(Prep)	13.79(4/29)	26.12(35/134)
R(Prep)	1.29(4/311)	11.25(35/311)
F_1(Prep)	2.35	15.73
P(Nn)	24.81(65/262)	27.27(102/374)
R(Nn)	16.41(65/396)	25.76(102/396)
F_1(Nn)	19.76	26.49
P(SVA)	24.42(21/86)	24.42(21/86)
R(SVA)	16.94(21/124)	16.94(21/124)
F_1(SVA)	20.00	20.00
P(Vform)	19.35(6/31)	19.35(6/31)
R(Vform)	4.92(6/122)	4.92(6/122)
F_1(Vform)	7.84	7.84
P(all)	35.65(272/763)	34.23(420/1227)
R(all)	16.56(272/1643)	25.56(420/1643)
F_1(all)	22.61	29.27

Table 1. Different performances according to different confidence parameters. *Det* stands for ArtOrDet.

Pre-processing and post-processing we propose also contribute to some extent which we could see from Table 2. Some idiomatic phrases are excluded from being corrected in pre-processing which enhances precision while some are being modified in post-processing to improve recall.

	Without pre-processing and post-processing%	Final%
P	33.72(265/768)	35.65(272/763)
R	16.13(265/1643)	16.56(272/1643)
F_1	21.82	22.61

Table 2. Comparison with and without pre-processing and post-processing.

We didn't do much on the interacting errors problem since we didn't work out perfect plans to solve it. So, in the result combination module, we just simply combine the result of each part together.

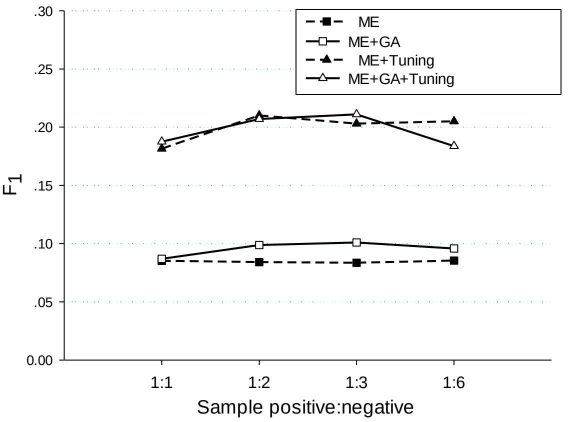


Figure 4. Comparisons before and after Genetic Algorithm on ArtOrDet error type. ME, GA, and Tuning stand for Maximum Entropy, Genetic Algorithm and confidence tuning.

In the revised version, under further corrections for the gold annotations, our model achieves precision of 41.75%, recall of 20.19% and F_1 of 27.3%.

6 Discussion

Which factor contributes most to the final result in the problem of grammatical error correction? Since we didn't include any external corpora, we discuss it here only according to the local classifiers and context features.

Based on our experiments, we find that, in our machine learning based modules, a tiny modification of confidence parameter setting for each category, no matter which type of error, can have great impact on the final result. It results in that our model is much too sensitive to parameters which may easily lead to a poor behavior. Perhaps a sufficient consideration of how to keep the distribution of samples, such as cross-validation, may be helpful. In addition, the selection of classifiers, features and training samples all have effect on the result more or less, but not as obvious as that of the confidence threshold setting.

7 Conclusion

In this paper, we propose a hybrid model combining machine learning based modules and rule-based modules to solve the grammatical error correction task. We are able to solve a subset of the correction problems in which ArtOrDet and Nn perform better. However, our result in the testing data shows that our model is sensitive to parameters. How to keep the distribution of training samples needs to be further developed.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (No. 61272383 and 61173075).

References

Bernd Bohnet. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of COLING,* 2010.

C. Fellbaum. WordNet: An Electronic Lexical Data-base. *MIT Press*. 1998.

Daniel Dahlmeier, and Hwee Tou Ng. Grammatical error correction with alternating structure optimization. In *Proceedings of ACL*. Association for Computational Linguistics, 2011.

Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. NUS at the HOO 2012 Shared Task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012a.

Daniel Dahlmeier and Hwee Tou Ng. A beam-search decoder for grammatical error correction. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 2012b.

Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In Proceedings of NAACL, Association for Computational Linguistics, 2012c.

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

De Felice, Rachele and Stephen G. Pulman. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of COLING*. Association for Computational Linguistics, 2008.

G. Minnen, J. Carroll and D. Pearce. Robust, applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference*, 2000.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of ACL* , 2005.

Joel R. Tetreault and Martin Chodorow. The ups and downs of preposition error detection in ESL writing. In *Proceedings of COLING*, Association for Computational Linguistics, 2008.

John Lee and Stephanie Seneff. Correcting misuse of verb forms. In *Proceedings of ACL: HLT*, 2008.

M Anbarasi, E Anupriya, and NC Iyengar. Enhanced prediction of heart disease with feature subset selection using genetic algorithm. *International Journal of Engineering Science and Technology*,Vol.2(10),2010: 5370-5376.

ME ElAlami. A filter model for feature subset selection based on genetic algorithm. *Knowledge-Based Systems*,Vol.22(5), 2009: 356-362.

Michael Heilman, Aoife Cahill, and Joel Tetreault. Precision isn't everything: a hybrid approach to grammatical error detection. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012.

Hongsuck Seo et al. A meta learning approach to grammatical error correction. In *Proceedings of ACL*. Association for Computational Linguistics, 2012.

N.R. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, Vol.12(02):115-129.

S. Bergsma, D. Lin, and R. Goebel. 2009. Web-scale ngram models for lexical disambiguation. In *Proceedings of IJCAI*.2009.

X. Yi, J. Gao, and W.B. Dolan. 2008. A web-based English proofing system for English as a second language users. In *Proceedings of IJCNLP*.2008.