Grammatical Error Correction Using Feature Selection and Confidence Tuning

Yang Xiang[1†], Yaoyun Zhang[1], Xiaolong Wang[1‡], Chongqiang Wei[1], Wen Zheng[1], Xiaoqiang Zhou[1], Yuxiu Hu[2], Yang Qin[1]
[1]Key Laboratory of Network Oriented Intelligent Computation,
Harbin Institute of Technology Shenzhen Graduate School, China
[2]South University of Science and Technology of China
[†]windseed@gmail.com [‡]wangxl@insun.hit.edu.cn

Abstract

This paper proposes a novel approach to resolve the English article error correction problem, which accounts for a large proportion in grammatical errors. Most previous machine learning based researches empirically collected features which may bring about noises and increase the computational complexity. Meanwhile, the predicted result is largely affected by the threshold setting of a classifier which can easily lead to low performance but hasn't been well developed yet. To address these problems, we employ genetic algorithm for feature selection and confidence tuning to reinforce the motivation of correction. Comparative experiments on the NUCLE corpus show that our approach could efficiently reduce feature dimensionality and enhance the final F_1 value for the article error correction problem.

1 Introduction

Grammatical errors in English are common in written issues especially for learners of English as a second language (L2 leaners). As a result, automatic grammatical error correction (GEC) sprung up and has attracted more and more research attention recently. Among various error types, article errors account for a large proportion (over 12% in NUCLE) and are very difficult to be corrected.

Articles in the English language include indefinite article *a* and *an*, definite article *the* and zero article *empty* which means no article is used in this position. Articles are determiners of noun phrases which are indispensable in English grammar. Article errors are common in written English including wrong use, missing, and unnecessary use of articles. For example, in the following sentence *"Over these years, it had helped humans to improve the accessibility in the forms of cards to gain access to certain places."* there are two *the*s in which the first one is required but the second one is unnecessary. It is difficult for L2 learners to judge whether an article is necessary or not, or which article is needed. These errors are highly correlated with the context features around noun phrases. Errors occur frequently in various written issues which motivates researchers to exploit automatic error correction.

There are two main approaches for English article error correction. One of them is the external language materials based approach. Although there are minor differences on strategies, the main idea of this approach is to use frequencies such as n-gram counts as a filter and keep those phrases that have relatively high frequencies. Typical researches are shown by (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 in (Heilman et al., 2012). The other is machine learning based approach in which syntactic and semantic context features are utilized to train classifiers. 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.

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

International Joint Conference on Natural Language Processing, pages 1067–1071,
Nagoya, Japan, 14-18 October 2013.

As far as we know, most machine learning based approaches collect their features empirically and mainly depend on the feature selection of the classifiers which may bring about noises and increase the computational complexity when the feature dimensionality goes excessive. Moreover, discussions about the setting of threshold in classifiers are insufficient. Some work made simple adjustments on predicted thresholds after training their classification models like (Han et al., 2006; Dahlmeier and Ng, 2012a). Tetreault and Chodorow (2008) proceed from the different confidence of predicted categories which is similar to the approach employed in our work. We consider it is crucial to measure the differences between predicted scores of each category especially for GEC task on those documents with relatively high quality because in many cases, to keep the original form are actually the best choice.

In this paper, we focus on the machine learning based approach on error annotated corpus and propose a novel strategy to solve article error correction problem. Primarily, we extract a large number of related syntactic and semantic features from the context. With the help of genetic algorithm, a best feature subset is selected out which could greatly reduce the feature dimensionality. For each testing instance, according to the predicted confidence scores generated by the classifier, our tuning approach measures the trade-off between scores in order to enhance the confidence to a certain category. We didn't include any external corpora as references in our work which is to be further exploited. Experiments on NUCLE corpus show that our approach could efficiently reduce feature dimensionality and take full advantage of predicted scores generated by the classifier. The evaluation result shows our approach outperforms the state-of-the-art work (Dahlmeier and Ng, 2011) by 2.2% in F_1 on this corpus.

There are two main contributions in our work: one is that we add feature selection before training and testing which reduces feature dimensionality automatically. The other is that we make use of the differences of confidence scores between categories and discuss about various tuning approaches which may affect the final performance.

The remainder of this paper is arranged as follows. The next section introduces feature extraction and selection. Section 3 describes model training and confidence tuning. Experiments and analysis are arranged in Section 4. Finally, we give our conclusion in Section 5.

2 Feature Extraction and Selection

We take article correction as a multi classification task. Three categories including *a/an*, *the* and *empty* are assigned to specify the correct article forms in corresponding positions (*a* and *an* are distinguished according to pronunciation of the following word). For training, developing and testing, all noun phrases (NPs) are chosen as candidates to be corrected. We extract related features based on the context of an NP and do feature selection afterwards.

2.1 Feature Extraction

A series of syntactic and semantic features are extracted with the help of NLP tools like Stanford parser (Klein and Manning, 2003), Stanfordner (Finkel et al., 2005) and WordNet (Fellbaum, 1999). We adopt syntactic features such as the surface word, word n-gram, part-of-speech (POS), POS n-gram, constituent parse tree, dependency parse tree, name entity type and headword; semantic features like noun category and noun hypernym. Some extended features are extracted based on them and some previous work (Dahlmeier and Ng, 2012b; Felice and Pulman, 2008).

Through feature extraction, we get over 90 groups of different features. After binarization, the dimensionality exceeds to about 350 thousand in which many features occur only once. We tried to prune all sparse features but found the performance fell off greatly while a manual deletion of several of them could instead improve the result. We infer that the sparse features may become useful when serving as an element of some feature subset which motivates us to carry out feature selection.

2.2 Feature Selection

Feature subset selection is conducted in this module to select out wrapped features. Genetic algorithm (GA) has been proven to be useful in selecting wrapped features in previous work (ElAlami, 2009; Anbarasi et al, 2010) and is applied in our work.

The features are encoded into a binary sequence in which each character represents one dimension. We use the number "1" to denote that this dimension should be kept while the number "0" means that dimension should be dropped in classification. A binary sequence such as "0111000…100*"* is able to denote a combination of feature dimensions. GA functions on the feature sequences and finally decides

which feature subsets should be kept. Following the steps of traditional GA, our approach includes generation of initial individuals, crossovers, mutations and selection of descendants for each generation.

The fitness function is the evaluation metric F_1 described in §**4.1**. After feature selection, we reduced our feature dimensionality from 350 thousand to about 170 thousand which greatly reduced complexity in training. As expected, there are still a great number of sparse features left.

3 Training and Tuning

3.1 Training Using Maximum Entropy

All noun phrases (NPs) are chosen as candidate instances to be corrected. For NPs whose articles are erroneous with annotations, the correct ones are their target categories, and for those haven't been annotated (error-free), their target categories are the observed articles. These NPs contain two basic types: *with* and *without* wrong articles. Two examples are shown below:

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

For each category in *a*, *the*, and *empty*, we use the whole *with* instances and randomly take samples of *without* ones, making up the training instances for each category. We consider all the *with* samples useful because each of them has an observed wrong article which indicates that the correct article is easily misused as the wrong one. Different ratios of *with : without* are experimented in our work to see how much the number of *without* samples, which is mentioned in previous work (Dahlmeier and Ng, 2011), affects the result in our model.

Maximum entropy (ME) is employed for classification which has been proven to have good performance for heterogeneous features in natural language processing tasks. We have also tried several other classifiers including SVM, decision tree, and Naïve Bayes but finally found ME performs better.

3.2 Confidence Tuning

ME returns with confidence of each category for a given testing instance. However, for different instances, the distributions of predicted scores vary a lot. In some instances, the classifier may have a very high predicted score to a certain category which means the classifier is confident enough to perform this prediction while for some other instances, two or more categories may share close scores, the case of which means the classifier hesitates when telling them apart.

Our confidence tuning strategy (Tuning) on the predicted results is based on a comparison between the observed category and the predicted category. It is similar to the "thresholding" approach described in (Tetreault and Chodorow, 2008). The main idea of this confidence tuning strategy is: the selection between *keep* and *drop* is based on the difference between confidence of the predicted category and the observed category. If this difference goes beyond a threshold *t*, the prediction is proposed while if it is under *t*, we won't do any corrections. The confidence threshold is generated through hill-climbing in development data aiming at maximizing F_1 of the result.

4 Experiments

4.1 Data Set and Evaluation Metrics

The NUCLE corpus (Dahlmeier and Ng, 2011) introduced by 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 corpora. Only about 1.8% of articles contain errors in this corpus. The corpus provides the original texts as well as annotations which we make use of to generate training and developing samples. We divide the whole corpus into 80%, 10% and 10% for training, developing and testing to make our approach comparable with the previous work.

The performance is measured with precision, recall and F_1-measure where 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 with precision while its denominator is the amount of manually corrected errors.

4.2 Experiment and Analysis

In our experiments, we firstly compare the results of the baseline system (without GA and tuning, labeled as ME) and GA to see how much GA contributes to the performance. And also, we list the results of our initial strategy that all sparse features were deleted from the feature space (-SF). The comparisons are shown in Table 1 (all the *without* instances are used without sampling). The results show the effectiveness of GA and the usefulness of the sparse features.

Secondly, we tried several *with*: *without* ratios in the composition of training instances to see how much the selection of instances affects the result in our model. Figure 1 describes the comparative results under different ratios and the results with or without confidence tunings (discussed next).

Model	Prec.	Rec.	F_1
ME(-SF)	4.29	66.67	8.05
ME	4.46	65.80	8.35
ME+GA	5.42	68.68	**10.06**
ME+Tuning(-SF)	13.33	26.17	17.66
ME+Tuning	15.85	28.20	20.03
ME+GA+Tuning	20.19	23.04	**21.53**

Table 1. Experiments on feature selection.

This experiment is conducted without the intervention of GA. Different from the conclusion in the previous work, we find that, there are not obvious differences between results under different ratios in our model. Before tuning, the differences are tiny, and after tuning, we believe it is mainly due to the advantage of the tuning strategy that eliminates the effects of randomness to a great extent. It is also interesting to see the improvement of F_1 always follows the increase of precision and decrease of recall which is good for the trend of correction without human intervention. We use all the *without* instances in our following experiments to avoid other randomness.

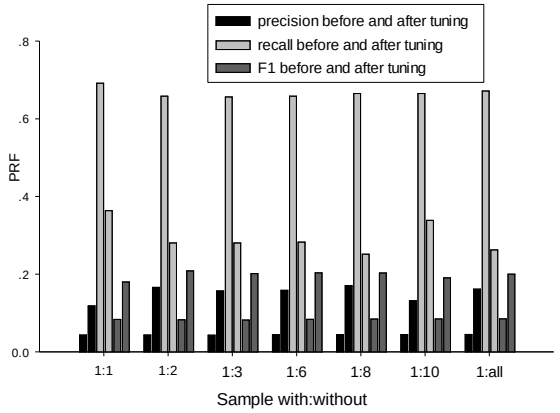

Figure 1. Comparisons before and after tuning. (*1:all* means to use the whole negative samples which is about *1:13*).

The best result of our model is achieved with GA and confidence tuning (ME+GA+Tuning in Table 1). Through experiments, we notice that the contribution of confidence tuning accounts for the largest proportion which directly enables our model outperform the previous state-of-the-art work (precision of 26.44%, recall of 15.18%, and F_1 of 19.29% by (Dahlmeier and Ng, 2011)) by about 2.2% which is a big improvement in this task. Besides, the performance on the test set keeps that on the developing set which achieves precision of 20.97%, recall of 21.25%, and F_1 of 21.11%.

At last, we make comparisons on four threshold tuning strategies to verify the appropriateness of the thresholding approach applied in this paper. The five approaches labeled as *no-tuning*, *self-tuning*, *all-tuning*, *self-diff*, and *all-diff* in Table 2 correspond to the following four strategies. **(1)** Choose the category with the maximum predicted score; **(2)** Assign each category a fixed threshold beyond which a score goes most, that category is predicted; **(3)** Assign a fixed threshold for all categories beyond which a score goes most, that category is predicted; **(4)** Similar to (2) except that the threshold is the difference between scores of the predicted maximum and the observed category; **(5)** Similar to (3) except that the threshold is the difference between scores of the predicted maximum and the observed category.

Tuning method	Prec.	Rec.	F_1
no-tuning	5.42	68.68	10.06
self-tuning	20.38	21.04	20.90
all-tuning	22.04	15.88	18.47
self-diff(our)	20.19	23.04	**21.53**
all-diff	22.82	17.00	19.49

Table 2.Different tuning strategies

It is noticeable that to assign a threshold for each category always performs better than to use a single threshold. We infer that the tuning strategies based on difference perform better mainly because they consider that the observed category should have a relatively high confidence if it is error-free even it is not the maximum.

5 Conclusion

In this paper, we introduce feature selection and confidence tuning for the article error correction problem. Comparative experiments show that our approach could efficiently reduce feature dimensionality and enhance the final F_1 value. However, for automatic grammatical error correction, there is still a long way to go. More resources and methods need to be exploited in the next stage for further performance improvement.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 612-72383 and 61173075).

References

Anbarasi, M, 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.

Bergsma, S., D. Lin, and R. Goebel. 2009. Web-Scale Ngram Models for Lexical Disambiguation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (*IJCAI*), 2009.

Dahlmeier, Daniel and Hwee Tou Ng. Grammatical Error Correction with Alternating Structure Optimization. In *Proceedings of the 49th Annual Meeting on Association for Computational Linguistics* (*ACL*). Association for Computational Linguistics, 2011.

Dahlmeier, Daniel, Hwee Tou Ng, and Eric Jun Feng Ng. NUS at the HOO 2012 Shared Task. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012a.

Dahlmeier, Daniel and Hwee Tou Ng. A Beam-Search Decoder for Grammatical Error Correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP*). Association for Computational Linguistics, 2012b.

ElAlami, ME. A Filter Model for Feature Subset Selection Based on Genetic Algorithm. *Knowledge-Based Systems*, Vol.22(5), 2009: 356-362.

Felice, Rachele De and Stephen G. Pulman. A Classifier-based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics* (*COLING*). Association for Computational Linguistics, 2008.

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

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of *the 43rd Annual Meeting on Association for Computational Linguistics* (*ACL*). Association for Computational Linguistics, 2005.

Han, N.R., M. Chodorow, and C. Leacock. Detecting Errors in English Article Usage by Non-native Speakers. *Natural Language Engineering*, Vol.12(02):115-129. 2006.

Heilman, Michael, Aoife Cahill, and Joel Tetreault. Precision Isn't Everything: A Hybrid Approach to Grammatical Error Detection. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2012.

Klein, Dan and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (*ACL*). Association for Computational Linguistics, 2003.

Seo, Hongsuck et al. A Meta Learning Approach to Grammatical Error Correction. In *Proceedings of the 50th Annual Meeting on Association for Computational Linguistics* (*ACL*). Association for Computational Linguistics, 2012.

Tetreault, Joel R. and Martin Chodorow. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics* (*COLING*). Association for Computational Linguistics, 2008.

Yi, X., J. Gao, and W.B. Dolan. 2008. A Web-Based English Proofing System for English as a Second Language Users. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing* (*IJCNLP*), 2008.