Convolutional Neural Networks for Correcting English Article Errors

Chengjie Sun[1(✉)], Xiaoqiang Jin[1], Lei Lin[1], Yuming Zhao[2], and Xiaolong Wang[1]

[1] Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
cjsun@insun.hit.edu.cn
[2] Northeast Forestry University, Harbin 150040, Heilongjiang, China

Abstract. In this paper, convolutional neural networks are employed for English article error correction. Instead of employing features relying on human ingenuity and prior natural language processing knowledge, the words surrounding the context of the article are taken as features. Our approach could be trained both on an error annotated corpus and an error non-annotated corpus. Experiments are conducted on CoNLL-2013 data set. Our approach achieves 38.10 % in F1, and outperforms the best system (33.40 %) that participates in the task. Experimental results demonstrate the effectiveness of our proposed approach.

Keywords: Convolutional neural networks · Article error correction · Deep learning

1 Introduction

Grammatical Error Correction (GEC) for non-native English language learners has gained more and more attention with the developing of the Natural Language Processing (NLP), machine learning and big-data techniques [1]. Given an English essay written by a learner of English as a Second Language (L2) or English as a Foreign Language (EFL), the task of GEC is to detect and correct the grammatical errors present in the essay, and return the corrected essay.

The most representative approaches for GEC use machine learning-based classifier. However, features employed in these approaches either rely on human ingenuity or prior NLP knowledge. It takes a lot of manpower and lead to the propagation of errors in the existing tools of NLP. And different errors usually employ different features, that is to say, to correct a new type of error, features need to be extracted again. Besides, lacking of training data often prohibits a robust statistical model to be trained.

In this paper, we focus on the article error since it is one of the most difficult challenges faced by non-native speakers. A simple convolutional neural networks with one layer of convolution and pooling is employed to solve article error. The model takes words surrounding the context of article as input and outputs the

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 102–110, 2015.
DOI: 10.1007/978-3-319-25207-0_9

label of the {a/an, the, ϵ } representing the correct article which should be used in the context (ϵ stands for no article).

The contributions of this paper can be summarized as follows:

- Instead of employing features relying on human ingenuity and prior NLP knowledge, our approach simply takes contexts words surrounding the articles as features.
- Our approach could be trained on both an error annotated corpus and an error non-annotated corpus making it possible to learn a robust statistical model on sufficient examples of an error type.

2 Related Work

According to the approaches used, researches on GEC could be divided into five categories [2]: machine learning-based classifier approach [1,3], machine translation approach [4–6], hybrid classifier approach [7], language modeling-based approach [8] and rule-based [9] approach.

As deep learning approaches have achieved remarkable results in computer vision [10] and speech recognition [11], lots of researches have been done to explore how deep learning could be used to tasks of NLP. Word embeddings [12–14] have been one of the most successful research achievements of deep learning on NLP. In most work of deep learning on NLP, word embeddings features have been fed to a Convolutional Neural Network (CNN) to solve different tasks including semantic parsing [15], search query retrieval [16], sentence classification [17], and other traditional NLP tasks [18].

3 Model

Same as most researches on GEC, we treat the article error as a problem of multi-class classification with three labels: *a/an*, *the*, and ϵ. A convolutional neural networks (CNN) based method was proposed for this multi-class classification task. The idea behind the proposed method is that given the context of an article, the proper article may be chosen, just as a human would.

Instead of using much complicated syntactic or semantic features which may lead to the propagation of errors in the existing tools of NLP, the proposed method only takes words surrounding the article (not including the article) as features. In this way, the CNN model can be trained on both an error annotated corpus and an error non-annotated corpus. Through looking up the table of word embeddings, words are transformed into vectors. The deep feature representation of the contexts of articles are learned through CNN. Finally, the features learned are fed into a softmax classifier to compute the confidence of the each label that may occur in the given context.

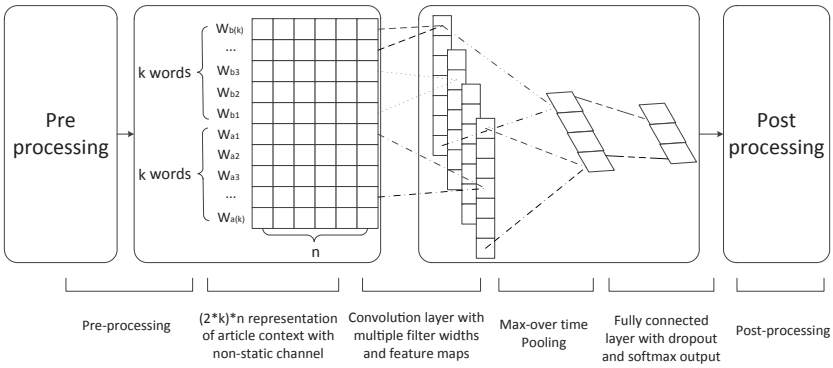

Fig. 1. Model Architecture for Article Error

3.1 Model Architecture

Figure 1 describes the architecture of the proposed article error correction method. The architecture includes the following three parts: preprocessing module, CNN module with one layer of convolution and pooling, and postprocessing module.

The preprocessing module produces the inputs for CNN module. It extracts the words surrounding the articles or the spaces at the beginning of a noun phrase (if there are no articles). The CNN module is the same as [17] except that only non-static channel is used. The postprocessing module is to discriminate when to use the article *a* or *an* by rules and to produce the output text.

3.2 Preprocessing Module

This module aims at extracting surrounding context of an article including ϵ representing not using an article. Firstly, we extract the surrounding context of *a/an* and *the* from the data. For ϵ, we get the word before the beginning of a noun phrase. If the word is not an article, we treat the surrounding context of space at the beginning of a noun phrase as ϵ's surrounding context.

A sentence can be denoted as the following format:

$\ldots w_{b4}\ w_{b3}\ w_{b2}\ w_{b1}\ Art\ w_{a1}\ w_{a2}\ w_{a3}\ w_{a4} \ldots$ where $Art \in \{a/an, the, \epsilon\}$, w_{bi} *and* w_{ai}, $i \in \{1, 2, 3, 4, \ldots\}$ represent the ith word before and after Art. Let w_b^k refer to $w_{b(k)}, w_{b(k-1)}, \ldots, w_{b1}$ and w_a^k refer to $w_{a1}, w_{a1}, \ldots, w_{a(k)}$. The inputs of CNN module are M words $(w_b^{M/2}, w_a^{M/2})$ surrounding the article Art.

For input whose article have been annotated in the corpus, the correct one is its label, otherwise, the label is the Art.

3.3 CNN Module

This module first extracts contexts words surrounding the article. Then these words are mapped to vectors by looking up word-embeddings tables. After that,

CNN module takes the vectors of contexts words as input. New features are produced by applying many convolution operations to a window of h words in CNN module. A max over time pooling operation is applied to each feature map [18]. The idea behind this is to capture the most important feature by the maximum value. Finally, a fully connected layer with dropout is used to compute the confidence of each possible output class.

The CNN module uses non-static channel and word vectors are also taken as parameters [17]. In this way, article specific vectors could be learned through training. The optimal parameters are achieved through Stochastic Gradient Descent (SGD). And the implement of CNN is most based on the code[1] provided by [17].

3.4 Postprocessing Module

In English, there are rules to determine when to use *a* or *an* by considering the phonetic properties of word immediately after the a/an. Those rules implemented through CMU pronouncing dictionary[2], are employed to revise the output of our CNN module.

4 Dataset and Evaluation Metrics

To evaluate the performance of our approach, we use the data provided by CoNLL-2013. The training data use the NUCLE Corpus [19] and has been annotated with error-tag and correction-labels. In the shared task, 25 non-native speakers of English from NUS were recruited to write new essays to be used as blind test data. The statistics of the NUCLE corpus are shown in Table 1.

Table 1. Statistics of training and test data

	Train(NUCLE)	Test
#Essays	1,397	50
#Sentences	57,151	1,381
#Word Tokens	1,161,567	29,207
#ArtOrDet* Error	6,642	690

* ArtOrDet is short for articles and determiners.

The performance of a grammatical error correction system is evaluated by how well its proposed corrections or edits match the gold-standard edits. Precision, recall and F-score are often chosen as the evaluation criteria. The test data and official scorer (M^2 scorer [20]) provided by CoNLL-2013 are freely available[3].

[1] https://github.com/yoonkim/CNN_sentence
[2] In this paper, we use the interface provided by Natural Language Took Kit (NLTK).
[3] http://www.comp.nus.edu.sg/~nlp/conll13st.html

5 Experiments

In this section, two sets of experiment are conducted. One is to understand how the choice of hyperparameters affects the performance on development data set which was obtained by holding out 20% of the training data. The other evaluates the final performance of our approach on the test data of CoNLL-2013.

5.1 Pre-trained Word Embeddings

Researchers [18,21] have reported that initializing word embeddings with those learned from significant amounts of unlabeled data are far more satisfactory than the randomly initialized.

In this paper, we do not conduct a comparison of the available word embeddings, for it's beyond the scope of this paper. Embeddings provided by [22] are utilized in our experiments to initialize word embeddings table. Words not present in the set of pre-trained words are initialized randomly, which is same as [17] does.

5.2 Parameter Settings

We experimentally study the effects of the two parameters in our model: window size of error contexts and the number of feature map.

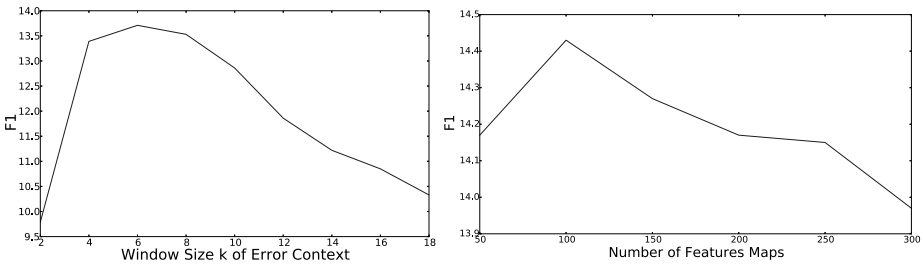

Fig. 2. Effects of Hyperparameters

In Figure 2, parameters about window size of error contexts and number of feature map are respectively varied. As shown in Figure 2 , the performance does not improve when the window size k of error contexts is larger than 6. Since the training data is limited, the model is prone to overfitting especially when the number of feature map (f) exceeds 100. The dimensionality of word embeddings (n) is the same as in [18]. The filter windows (h), dropout rate (p), l_2 constraint (s), and mini-batch size (m) are following [17]'s experiment settings. Table 2 lists the parameters used in the following experiments.

Table 2. Hyperparameter Settings

Parameter	k	n	h	p	s	m	f
Value	6	50	3,4,5	0.5	3	50	100

5.3 Experiment Results

Table 3 shows the results of article error correction of the top three systems in CoNLL-2013 [2] open evaluation and our approach. Besides, the linguistic features used in each systems are also given.

Table 3. The Article Result of Top3 in CoNLL-2013 and Our Approach

Team	Precision	Recall	F1	Linguistic Features
UIUC	47.84	25.65	33.40	lexical, POS, shallow parse
HIT	42.82	24.20	30.93	lexical, POS, constituency parse, dependency parse, semantic
NTHU	35.80	21.01	26.48	lexical, POS, constituency parse, dependency parse
Ours	30.15	51.74	38.10	lexical

Team UIUC employed a multi-class averaged perception for article error correction. Maximum entropy with confidence tuning was used for article error correction in team HIT. In team NTHU, N-gram-based and dependency-based language model was employed.

Our approach performs better than UIUC, HIT and NTHU in recall and F1, though only taking lexical as features. However, the precision of our approach is much lower. One possible reason is that source words are not used in our CNN model. Previous works have showed that the authors' word choices (source word) obey certain regularities [23, 24] and systems without employing the source word have a very poor precision [25]. When the source word is directly utilized as feature, model tends to have a low recall due to the error sparsity. [25] proposed an error inflation method to avoid this problem by adding artificial errors in train data based on the error distribution in the train set. We try to explain the low precision by adopting error inflation method to our model.

Figure 3 shows the result of effects of source word on development data set. The results show that source word could balance the recall the precision. When the inflation constant (C) is set to 0.7, F1 get the best result on the development set. The result on test is shown in Table 4. We fail to find the best trade-off between recall and precision on test set due to different distribution of error between train data and test data, but shed some light on why CNN model have a low precision.

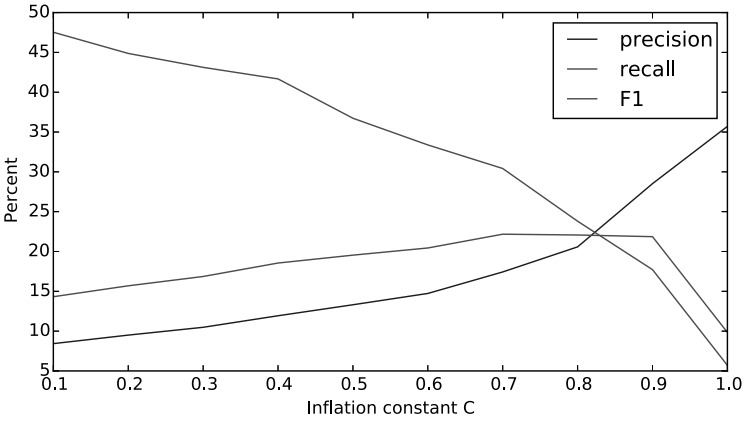

Fig. 3. Effects of Source Word

Table 4. CNN with Source Word

Precision	Recall	F1	C
45.10	22.03	29.60	0.7

6 Conclusion and Feature Work

In this paper, we exploit a convolutional neural network for English article error correction. Instead of employing features relying on human ingenuity and prior NLP knowledge, our approach simply takes words surrounding the contexts of articles as features. Though simple features are employed, experimental results conducted on CoNLL-2013's data set demonstrate the effectiveness of our approach. Besides, our approach could be trained on both an error annotated corpus and an error non-annotated corpus.

In the future, an effective post processing module determining whether to accept the correction will be explored to solve the drawback of low precision. The effectiveness of our approach on other error types will also be explored.

Acknowledgments. We thank reviewers for their helpful comments on an earlier version of this work. This work is supported by National Natural Science Foundation of China (61300114 and 61272383) and Natural Science Foundation of Heilongjiang Province(F201132).

References

1. Xiang, Y., Yuan, B., Zhang, Y., Wang, X., Zheng, W., Wei, C.: A hybrid model for grammatical error correction. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 115–122 (2013)

2. Ng, H.T., Wu, S.M., Wu, Y., Hadiwinoto, C., Tetreault, J.: The conll-2013 shared task on grammatical error correction. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–12 (2013)
3. Rozovskaya, A., Chang, K.W., Sammons, M., Roth, D.: The university of illinois system in the conll-2013 shared task. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 13–19 (2013)
4. Yuan, Z., Felice, M.: Constrained grammatical error correction using statistical machine translation. CoNLL-2013, pp. 52–61 (2013)
5. Buys, J., van der Merwe, B.: A tree transducer model for grammatical error correction. CoNLL-2013, pp. 43–51 (2013)
6. Wilcox-OHearn, L.A.: A noisy channel model framework for grammatical correction. CoNLL-2013, pp. 109–114 (2013)
7. Xing, J., Wang, L., Wong, D.F., Chao, L.S., Zeng, X.: Um-checker: A hybrid system for english grammatical error correction. CoNLL-2013, 34 (2013)
8. Kao, T.H., Chang, Y.W., Chiu, H.W., Yen, T.H., Boisson, J., Wu, J.c., Chang, J.: Conll-2013 shared task: Grammatical error correction nthu system description. CoNLL-2013, 20 (2013)
9. Sidorov, G., Gupta, A., Tozer, M., Catala, D., Catena, A., Fuentes, S.: Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2). CoNLL-2013, pp. 96–101 (2013)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
11. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
12. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. The Journal of Machine Learning Research **3**, 1137–1155 (2003)
13. Yih, W.T., Toutanova, K., Platt, J.C., Meek, C.: Learning discriminative projections for text similarity measures. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, pp. 247–256 (2011)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
15. Yih, W.t., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: Proceedings of ACL (2014)
16. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, pp. 373–374 (2014)
17. Kim, Y.: Convolutional neural networks for sentence classification (2014). arXiv preprint, arXiv:1408.5882
18. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. The Journal of Machine Learning Research **12**, 2493–2537 (2011)
19. Dahlmeier, D., Ng, H.T., Wu, S.M.: Building a large annotated corpus of learner english: The nus corpus of learner english. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 22–31 (2013)

20. Dahlmeier, D., Ng, H.T.: Better evaluation for grammatical error correction. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 568–572 (2012)
21. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 151–161 (2011)
22. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 384–394 (2010)
23. Rozovskaya, A., Roth, D.: Annotating esl errors: Challenges and rewards. In: Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp. 28–36 (2010)
24. Lee, J., Seneff, S.: An analysis of grammatical errors in non-native speech in english. In: Spoken Language Technology Workshop, 2008. SLT 2008, pp. 89–92. IEEE (2008)
25. Rozovskaya, A., Sammons, M., Roth, D.: The UI system in the hoo 2012 shared task on error correction. In: Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, pp. 272–280 (2012)