7th ISCA Workshop on Speech and Language Technology in Education
25-26 August 2017, Stockholm, Sweden

Deep Context Model for Grammatical Error Correction

Chuan Wang, RuoBing Li, Hui Lin

Shanghai Liulishuo Information Technology Ltd, China
{chuan.wang,ruobing.li,h}@liulishuo.com

Abstract

In this paper, we propose a deep context model based on recurrent neural networks (RNN) for grammatical error correction. For a specific error type, we treat the error correction task as a classification problem where the grammatical context representation is learnt from native text data that are largely available. Compared with traditional classifier methods, our model does not require sophisticated feature engineering which usually requires linguistic knowledge and may not cover all context patterns. Experiments on CoNLL-2014 shared task show that our approach significantly outperforms the state-of-the-art classifier and machine translation approaches for grammatical error correction.

Index Terms: Grammar error correction, deep context model, recurrent neural network

1. Introduction

Automated grammatical error correction (GEC) is an essential and useful tool for millions of people who learn English as a second language. In recent years, much work has been done including several shared tasks: HOO [1, 2] and CoNLL [3, 4]. The methods used in HOO and CoNLL are generally based on three types of methods: pre-defined rules, classification and machine translation (MT). Rule-based methods cannot cover all grammar error patterns and are usually used in combination with other methods. In the classifier approach [5, 6], GEC is cast as a multi-class classification problem, where a confusion set is specified for a given error type, and features typically consist of surface forms of text as well as linguistic abstractions (e.g., part-of-speech tags, and parse information). In the classifier approach, error types shall be defined clearly before they can be corrected. For example, the article classifier [5] corrects errors using maximum entropy classifier, where features are combinations of words and part of speech tags. Other classifiers including averaged perceptron and naive Bayes algorithm are also used for GEC [6]. In these methods, features must be designed manually and it is difficult to cover all situations, and as a result, manually-designed features may not be sufficient for GEC due to the complexity of language.

Another mainstream method is based on statistical machine translation (MT)[7] and aims to translate incorrect text into correct text. One advantage of the machine translation approach is that it can take advantages of both large-scale linguistic resource (web-scale language models) and error-corrected texts. However, phrase-based MT methods suffer from limitations of discrete word representation, linear mapping and lack of global context. Recently, the neural machine translation (NMT) method has been applied to the GEC problem using encoder-decoder framework [8]. The NMT approaches can cope with redundancy and non-idiomatic phrasing errors which the classification method cannot handle. Other neural network models like bidirectional LSTMs [9] are also used in grammatical error detection tasks.

While the MT approaches cover a larger variety of error types and are better at dealing with complex mistakes such as those where multiple errors interact, classifier approaches enjoy at least two advantages. Firstly, it does not rely on annotated learner data which are expensive but required by most MT approaches. Secondly, classifier approaches are easy to incorporate higher level context information that goes beyond the surface form. Many grammatical errors may benefit from generalizations based on POS or parse information, and indeed it has been shown that classifiers perform better on errors that require linguistic abstractions [10].

In this paper, we propose a novel classifier approach for GEC based on a deep context model. Instead of using surface and shallow features (POS, parse information, etc), we use *deep* features directly. In particular, we use bidirectional Gated Recurrent Units (GRUs) to represent context. Compared with traditional classifier approach for GEC, our new method does not require elaborated feature engineering for each error type. Deep context representations are learnt from large plain text corpora in an end-to-end fashion.

Note that learning context representations with task-specific optimization using labelled data has been applied to various NLP tasks, including word sense disambiguation [11], coreference resolution [12] and paraphrase detection [13]. Generic word embeddings, such as word2vec [14] and Glove [15], learned from the large scale corpus, also capture the semantic and syntactic information about each individual word. In those methods, there are effective neural network architectures modelling the context [16]. Indeed, context is essential for the word choice and can help us correct the grammatical errors. On the other hand, unlike in those tasks, where large amount of supervised data is usually required but only available in limited size, our approach for GEC leverages the abundant native plain text corpora and learns context representation and classification jointly to correct grammatical errors effectively. Experiment results on CoNLL-2014 dataset show that our approach significantly outperforms state-of-the-art classifier approaches as well as MT approaches for GEC.

2. Model

2.1. Model overview

For a certain error type, the corresponding model learns an embedding function of variable-length contexts around the target word and then predicts the target word with the context embedding. If the predicted word is different from the original target word, the original word is flagged as a mistake and the prediction is then used as correction. Our deep context model uses a bidirectional Gated Recurrent Units and is based on the context2vec's [16] architecture. The context representation can be trained either from the beginning or the end of the sentence to the target word, or from the context words within a fixed-size

10.21437/SLaTE.2017-29

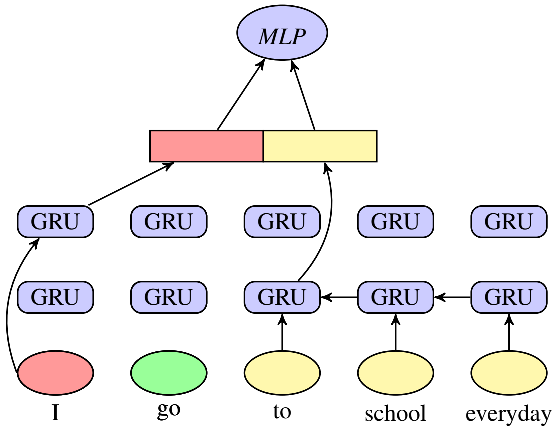

Figure 1: *Deep context model for grammatical error correction*

Error Types	Values of y
Article	0 = a/an 1 = the 2 = None
Preposition	label = preposition index
Verb form	0 = base form, 1 = gerund or present participle, 2 = past participle
Subject agreement	0 = non-3rd person singular present, 1 = 3rd person singular present
Noun number	0 = singular, 1 = plural

Table 1: *Values of* y *and their corresponding meanings for different error types.*

window.

Figure 1 illustrates how deep context vector represents the context and corrects text. We use two GRU recurrent neural networks. For the target word "go", we feed one GRU network with the context words (*"I"*) from left to right, and the other with context words (*"to school everyday"*) from right to left. Given a context $w_{1:n}$, our context vector for the target w_i is defined as the following equation:

$$biGRU(w_{1:n}, i) = lGRU(w_{1:i-1}) \oplus rGRU(w_{i+1:n}) \quad (1)$$

where the lGRU is a GRU reading the words from left to right in a given context and rGRU is a reverse one reading the words from right to left. l/r represent distinct left-to-right/right-to-left word embeddings of the context words. After that, we feed the concatenated vector to the multi-layer perceptron (MLP) to capture the inter-dependencies of the two sides, at the second layer of MLP, we use a softmax layer to predict the target word or the status (e.g., singular or plural) of the target word:

$$MLP(x) = softmax(ReLU(L(x))) \quad (2)$$

where MLP stands for Multi Layer Perceptron, ReLU is the Rectified Linear Unit activation function, $ReLU(x) = max(0, x)$, $L(x) = Wx + b$ is a fully connected linear operation. The final output of our model is

$$y = MLP(biGRU(w_{1:n}, i)) \quad (3)$$

where y could be either the predicted word or the predicted status of the target word. If the prediction is different from the original word or its status, a grammatical error is detected and the prediction is used as correction. For different error type, y is defined in different ways as shown in Table 1. In the article model, if y equals $0, 1$ or 2, it means the article should chosen "a/an", "the", or non-article respectively; In the preposition model, y represents the index of each preposition; In the verb form model, y denotes the form of the verb (0 is for the base form, 1 is for the gerund or present participle, and 2 is for the past participle); In the subject agreement model, 0 represent the noun-3rd person singular present, and 1 represents the 3rd person singular present. In the noun number model, 0 represents the singular noun, while 1 represents the plural noun.

We denote the labels of the classification as ŷ, and the objective function of training is then

$$loss = \frac{1}{n}\sum_{i=1}^{n} \hat{y_i} log(y_i) \quad (4)$$

where n is the number of training samples.

Note that the former deep learning method [8] considers all the errors unified and attempt to translate incorrect text into correct text, while our deep context approach learns a model for every specific grammatical error type.

3. Experiment

3.1. Dataset and metric

We evaluate the deep context model on CoNLL-2014 test dataset, which contains 1312 sentences. We use $F_{0.5}$ as the main evaluation measure for error correction. $F_{0.5}$ combines both precision (P) and recall (R), while assigning twice as much weight to precision, since accurate feedback is often more important than coverage in error detection.

The Precision, recall and $F_{0.5}$ are defined as follows:

$$P = \frac{e \cap g}{e} \qquad R = \frac{e \cap g}{g} \quad (5)$$

$$F_{0.5} = \frac{(1 + 0.5^2) \times R \times P}{R + 0.5^2 \times P} \quad (6)$$

where *g* is the gold standards of two human annotators for specific error type and *e* is the corresponding system edits. There are overlaps between many other error types and verb form error type, so *g* is based on the annotations of all error types when calculating verb form model performance.

We extract training samples from the wiki dump[1]. In the experiments, we use the Glove word embedding [15] to initialize the word embedding which are later updated during the training process. We set word embedding size to 300. The input text is lowercased and all tokens which are not in the vocabulary are represented as a single *unk* token. The vocabulary is made up of the most 40000 common used words in the wiki dump.

3.2. Error types

We build deep context models for five common types of grammatical errors: article, preposition, verb form, noun number, and subjective agreement. For each error type, classifiers are trained independently.

NLP tools like Stanford corenlp tools [17] are used to locate the target words that need to be checked. If the predication is different form the original label and the probability is larger than the predefined threshold, the grammatical error is deemed

[1] https://dumps.wikimedia.org/enwiki/

No.	Original	Proposed
1.	he might end up dishearten his family	he might end up disheartening his family
2.	it will just adding on their misery	it will just add on their misery
3.	... negative impacts to the family	... negative impacts on the family
4.	for the case of marriage, people should be honest	in the case of marriage, people should be honest.
5.	The popularity of social media sites have made ...	The popularity of social media sites has made ...
6.	Having support from relatives are vital	Having support from relatives is vital
7.	... be honest with his or her feeling	... be honest with his or her feelings
8.	... after realising his or her conditions	... after realising his or her condition .
9.	People get certain disease because of genetic changes.	People get a certain disease because of genetic changes.
10.	Especially for the young people without marriage	Especially for young people without marriage
11.	the government encourage people to give more birth	the government encourages people to give more birth

Table 2: *Examples of the deep context model corrections*

	CUUI (classifier)			Deep context model		
error type	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Article	31.5	46.7	33.7	45.7	31.9	**42.1**
Preposition	30.0	7.67	19.0	35.4	6.74	**19.1**
Verb form	53.6	5.39	19.2	57.7	3.88	15.3
Noun number	40.3	43.9	41.0	48.8	27.8	**42.4**
Subjective agreement	50.0	46.8	49.3	61.4	28.6	**49.9**

Table 3: *Compared result with the best classifier result CUUI in CoNLL-2014. The results are based on the combination of two annotators without alternative answers in CoNLL-2014.*

to be found. For example, in the subject agreement task, we use NLP tools to extract the non-3rd person singular present words and 3rd person singular present word map relationships in advance. During the test, the tool can locate the verbs which should be checked by our model. If an error is detected, we can then use the extracted word mapping to correct the sentence.

English learners often have problems with when to use a (or an), the, or no article at the beginning of a noun phrase. We treat article error correction as a three-category classification problem: a/an, the and zero article. The position where the article can appear should be in front of noun phrases and we regard noun phrase as a combination of noun words and adjective words in our grammatical error correction system.

Similar to article error correction, the subjective agreement task can be converted to a two-category problem: whether the verb should be third person singular present or not. We check every verb which is base form or the 3rd person singular present form.

We model the verb form error correction as a three-category problem: verb base form, gerund or present participle and past participle.

As for the preposition correction, we choose 11 most often used prepositions (*"about","at", "by", "for", "from", "in", "of", "on", "to", "until", "with", "against"*) as the classification labels.

Noun number correction can also be considered as a two-category problem: whether the noun should be plural or not. And we check all the noun words.

3.3. Window size

error type	window size	P	R	$F_{0.5}$
preposition	3	4.84	1.93	3.72
	5	16.7	3.86	10
	10	24.3	2.89	9.8
article	3	35.5	24.2	32.5
	5	45	30.9	41.3
	10	6.63	16.7	7.54
noun num.	10	50.0	32.8	45.3
	15	51.3	33.8	46.5
	20	52.1	30.8	45.8

Table 4: *Performance of models trained using context of different window sizes*

Correction of different types of grammatical errors might require dependencies from different distances. For instance, in subject agreement task, the status of verb can be affected by the subject which might be far away from the verb. In "*frequently, the intention of the carriers does not want to tell their families is to continue their own ...*" for example, the predicate "*is*" is far away from the subject "*the intention*". On the contrary, which preposition can be used is determined by the words near the target word. In "*to prevent the bigger problem from happen...*" , "*prevent from*" is a collocation which is usually close to each other. Therefore, we use different context window sizes for different grammatical error types. For subject agreement and verb form, we use the whole sentence as context since these two error types typically require dependencies from context words that are far away from the target words. As for article, preposition and noun number errors, we introduce a window and only context words that are within that window are considered. The window size is chosen based on its performances on the CoNLL-2013 testset, as shown in Table 4. In the article error type, the window size is set 5. The window size is set 3 and 15 for the preposition and noun number errors respectively.

3.4. Lemma

Lemma is the base form of a word. For instance, words *"walk", "walks", "walked", "walking"* all have the same lemma *"walk"*. In the noun number model, in addition to the existing context words around the target noun word, we also introduce the lemma form of the target noun word as extra "context" information, because whether the target should be singular or plural is closely related to itself. For example, for target word "*apples*" in sentence "*many apples are...*", the left context is now "*many apple*".

Table 5 shows that the noun number model fed with lemma can achieve better performance than the one without the lemma. So we choose to feed the lemma of the noun into the noun number model.

contain lemma	P	R	$F_{0.5}$
No	26.0	13.3	21.8
Yes	50.6	30.3	44.6

Table 5: *Performance of noun number model which is fed with and without the lemma*

3.5. Result

Some examples of deep context model corrections are shown in Table 2. The grammatical error patterns can be captured with their context representation. The first and second sentences show that the verb form (from base form to gerund or present participle form and the reverse) errors can be corrected. The preposition error is corrected in 3rd and 4th sentences. Even though the subject is not near the verb, the error is still corrected in the 5th and 6th sentence. Even though the surrounding words are similar in 7th and 8th sentences, the deep model still successfully corrects these noun number errors. Article errors are demonstrated in the 9th and 10th sentences.

Table 3 shows type-specific performances of deep context model, and the best classifier methods in CoNLL-2014. Compared with the best classifier approach in CoNLL-2014 (CUUI), deep context model performs better on article, preposition noun number and subjective agreement error types. For all five error types, the deep context models have higher precision than the CUUI method. Our system is a more effective grammar error correction system since precision is more important than recall in GEC tasks. In fact, the precision numbers shown in Table 3 could be much larger. Since some error types interact with each other, the correct corrections are sometimes under-counted. For example, in sentence 11 in Table 2, the word "encourage" is corrected into "encourages", but for this sentence "government" is annotated as "governments" in the gold-standard edits, and therefore the correct correction is then counted as a false correction. In other words, the errors which are left out in the gold-standard edits is one of the reasons for the under-estimated precision.

Lastly, we fix the mechanical errors (punctuation, spelling and capitalization errors) using existing resources and rule-based methods [10]. We measure the model performance and compare our system to several state-of-the-art systems on CoNLL-2014 shared task test dataset. The results are shown in Table 6. The top-1 system in CoNLL-2014 is a hybrid system combining rules and machine translation methods, while the top-2 system is a classifier based system (CUUI in Table 3). Our system outperforms these two systems significantly. We also compare our system with two more recent systems who have reported results on CoNLL-2014 testset. System in [8] considers all the errors in a unified way and attempts to translate incorrect text into correct text using an encoder-decoder recurrent neural network with an attention mechanism. Our system, although only addressing five common error types, achieves better results. In [10], the authors explore key strengths of both classifier approach and MT approach for GEC, and show that classifier approach actually does better in many aspects including the overall performance. We compare our system to the best classifier system in [10] without tailored training[2]. As can be seen, our system also outperforms this system significantly.

system	Performance		
	P	R	$F_{0.5}$
CoNLL-2014 top-2 system	41.8	24.9	36.8
CoNLL-2014 top-1 system	39.7	30.1	37.3
Xie et al.[8]	49.2	23.8	40.6
Rozovskaya et al.[10]	42.7	27.7	38.5
Deep Context Model	54.5	21.3	**41.6**

Table 6: *Overall performance of deep context model compared with state-of-the-art*

4. Discussion and Future Work

We propose a new neural network architecture to learn context representation and then use it to correct grammatical errors. It outperforms state-of-the-art classifier and MT methods for GEC. Compared with traditional classifier method, our approach does not need complex feature engineering since the context feature representation can be learnt jointly with classification in an end-to-end fashion, and the learning can be quite effective by utilizing enormous and easy-to-get native data. We find that different error types might require different amount of context information as shown in Section 3.3.

In the future we plan to introduce attention mechanism into our deep context model such that the model could focus only on those context words that affect grammatical usage. Also, as shown in [10], using a pipeline architecture where the MT is applied to the output of classifier can greatly improves the GEC performance. We believe that combining our deep context model with a state-of-the-art MT system will lead to further gain in the performance of GEC.

5. References

[1] D. Dahlmeier and H. T. Ng, "Grammatical error correction with alternating structure optimization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 915–923.

[2] ——, "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*. Association for Computational Linguistics, 2012, pp. 568–578.

[3] H. T. Ng, S. M. Wu, T. Briscoe, and C. Hadiwinoto, "The conll-2013 shared task on grammatical error correction." in *CoNLL Shared Task*, 2013, pp. 1–12.

[4] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The conll-2014 shared task on grammatical error correction." in *CoNLL Shared Task*, 2014, pp. 1–14.

[2] For some error types, only native data are used, while for some other error types, both native and learner data are used.

[5] N.-R. Han, M. Chodorow, and C. Leacock, “Detecting errors in english article usage by non-native speakers,” *Natural Language Engineering*, vol. 12, no. 02, pp. 115–129, 2006.

[6] A. Rozovskaya, K.-W. Chang, M. Sammons, D. Roth, and N. Habash, “The illinois-columbia system in the conll-2014 shared task.” in *CoNLL Shared Task*, 2014, pp. 34–42.

[7] M. J.-D. R. Grundkiewicz, “The amu system in the conll-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation,” *CoNLL-2014*, p. 25, 2014.

[8] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, and A. Y. Ng, “Neural language correction with character-based attention,” *arXiv preprint arXiv:1603.09727*, 2016.

[9] M. Rei and H. Yannakoudakis, “Compositional sequence labeling models for error detection in learner writing,” *arXiv preprint arXiv:1607.06153*, 2016.

[10] A. Rozovskaya and D. Roth, “Grammatical error correction: Machine translation and classifiers,” *Urbana*, vol. 51, p. 61820, 2016.

[11] A. Trask, P. Michalak, and J. Liu, “sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings,” *arXiv preprint arXiv:1511.06388*, 2015.

[12] K. Clark and C. D. Manning, “Deep reinforcement learning for mention-ranking coreference models,” *arXiv preprint arXiv:1609.08667*, 2016.

[13] Z. Wang, H. Mi, and A. Ittycheriah, “Sentence similarity learning by lexical decomposition and composition,” *arXiv preprint arXiv:1602.07019*, 2016.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[15] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.” in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[16] O. Melamud, J. Goldberger, and I. Dagan, “context2vec: Learning generic context embedding with bidirectional lstm,” in *Proceedings of CONLL*, 2016.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit.” in *ACL (System Demonstrations)*, 2014, pp. 55–60.