Toward a Task of Feedback Comment Generation for Writing Learning

Ryo Nagata
Konan University / 8-9-1 Okamoto, Kobe, 658-8501 Hyogo, Japan
Japan Science and Technology Agency, PRESTO / 4-1-8 Honcho, Kawaguchi,
Saitama 332-0012, Japan
nagata-emnlp2019 @ ml.hyogo-u.ac.jp.

Abstract

In this paper, we introduce a novel task called feedback comment generation — a task of automatically generating feedback comments such as a hint or an explanatory note for writing learning for non-native learners of English. There has been almost no work on this task nor corpus annotated with feedback comments. We have taken the first step by creating learner corpora consisting of approximately 1,900 essays where all preposition errors are manually annotated with feedback comments. We have tested three baseline methods on the dataset, showing that a simple neural retrieval-based method sets a baseline performance with an F-measure of 0.34 to 0.41. Finally, we have looked into the results to explore what modifications we need to make to achieve better performance. We also have explored problems unaddressed in this work.

1 Introduction

In this paper, we introduce a novel task called *Feedback Comment Generation*. Feedback comment generation is[1] the task of generating hints or explanatory notes (hereafter, feedback comments) for language learners[2]. Figure 1 shows an example of feedback comments on preposition use. As in this example, feedback comments are typically given to erroneous words in a given text so that the writer can understand why their writing is not good together with underlying grammatical rules and more importantly so that they can improve their writing skill.

Feedback comment generation complements grammatical error detection/correction in language learning. With the advent of Deep Neural Networks (DNN), their performance has dramatically improved in recent years. However, they are not capable of explaining why detected words are erroneous nor why they should be corrected as indicated. This limitation is particularly problematic for beginner to intermediate learners. They might not be able to understand and acquire the underlying rules; if so, they will likely make similar errors in the future. This is exactly where feedback comment generation comes in. Namely, it complements grammatical error detection/correction by generating feedback comments to help learners understand and acquire the underlying rules. This form of feedback should be useful for language learners as Bitchener et al. (2005)[3] and Sheen (2007) show.

Despite its usefulness, there has been almost no work on feedback comment generation as Sect. 2 will describe. One of the major reasons is that there exists no publicly available dataset for research on feedback comment generation. It is costly and time-consuming to annotate writings of non-native learners with feedback comments. It is not straightforward to decide what information one should give to learners of English as feedback comments and how; as far as we are aware of, the current work on feedback for writing learning (e.g., Bitchener et al. (2005); Ferris and Roberts (2001); Robb et al. (1986); Sheen (2007)) focuses on the comparison of the salience of feedback (only detection results, detection and correction results, or those with error types and so on). Besides, even if a dataset existed, it would still be a difficult task to generate human-like feedback comments.

To attack the above difficulties, we take the first step towards feedback comment generation targeting preposition use. The reason for the choice of preposition use is that (a) preposition er-

[1] The strict definition will be introduced in Subsect. 3.

[2] In this paper, *language learners* refer to learners of English as Foreign Language (EFL).

[3] Note that in their work, human teachers did error correction in written form and provided the learners with feedback comments orally.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3206–3215,
Hong Kong, China, November 3–7, 2019.

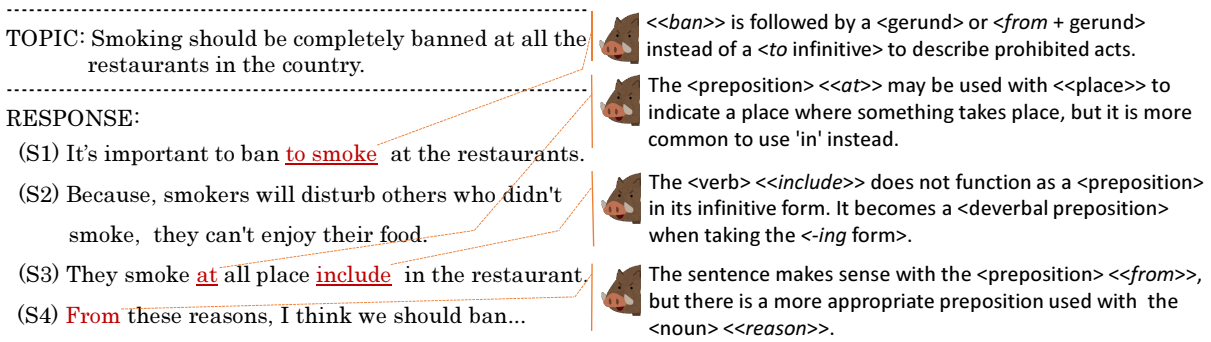

Figure 1: Example of Feedback Comments on Preposition Use.

rors are one of the most frequent error types in learner English (Tetreault and Chodorow, 2008); (b) they are relatively explainable (c.f., article errors); and (c) it would be a good starting point for feedback comment generation research considering the potential difficulty of the task. Specifically, we have annotated learner corpora with feedback comments, written in English and Japanese, on preposition use; we have released a part of the dataset to the public. Using the created dataset, we have trained and tested three baseline methods to estimate the feasibility of the task. In addition, we have looked into the results to reveal what problems we have left to achieve more complete feedback comment generation.

2 Related Work

The work most related to feedback comment generation is probably grammatical error detection/correction. The former is a task of detecting where errors exist in a given sentence whereas the latter is of retrieving its corrected form. In the beginning, researchers focused on specific error types – the most frequent ones such as errors in article (Han et al., 2006), preposition (Felice and Pulman, 2008), and number (Nagata et al., 2006). The state-of-the-art methods typically solve the problems as sequence labeling (e.g., Rei and Yannakoudakis (2016); Rei (2017); Kaneko et al. (2017)) or Machine Translation (MT) with DNNs (e.g., Junczys-Dowmunt et al. (2018); Napoles and Callison-Burch (2017)). As a result, performance has dramatically improved.

Unlike grammatical error correction, there has been almost no work on feedback comment generation. To the best of our knowledge, no work has even formally defined the task of feedback comment generation as we do in this paper. Some researchers (Kakegawa et al., 2000; McCoy et al., 1996; Nagata et al., 2014) made an attempt to develop rule-based methods for diagnosing errors in line with grammatical error detection/correction. Rule-based methods typically parse the input sentence and then apply rules to the resulting parse to diagnose errors. However, they encounter the tremendous difficulty of covering a wide variety of errors and of maintaining a large set of rules.

Comment generation for program source code is another task of comment generation. Its input and output are a piece of program source code such as a Java method and its comment or summary, respectively. Iyer et al. (2016) and Hu et al. (2018) solved the task as an MT problem (source code to comment) by using sequence-to-sequence models. This implies that feedback comment generation may be solved in the same manner (i.e., learner sentence to feedback comment).

3 Task Definition

Feedback comment generation in general is the task of generating feedback comments given a text (referred to as *essay*, hereafter). Thus, the input is an essay. The output is a set consisting of pairs of an offset and a feedback comment. An offset is a range of integers indicating where the paired feedback comment applies in the input essay. A feedback comment is a string that helps the writer (learner) improve their writing skill. It is typically a comment about a grammatical error, but can be about other things including discourse, organization, and content. It may be a side note to make the present writing better or praise to motivate the writer.

Note that because of the above, the range in feedback comment generation does not necessary match that in grammatical error detec-

tion/correction. Also note that there are cases where the input has no or more than one feedback comments; on the one hand, there exist sentences that receive no feedback comment at all as in (S2) in Fig. 1; on the other hands there exit sentences that receive more than one feedback comments as in (S3).

Our present target, feedback comment generation for preposition use, follows this task definition. Here, it is worthwhile to mention that we include *to*-infinitives in preposition use as in (S1) in Fig. 1. We also include errors that can be classified into multiple error types (preposition and other types). For example, in the sentence **Have a part-time job is easy.*, the underlined word *Have* can be interpreted as a missing *to*-infinitive (i.e., *To have*) or as a verb form error (i.e., *Having*). We include such cases in our targets to provide richer information.

4 Data Development

We created a dataset for feedback comment generation for preposition use. We used it in the evaluation as described in Sect. 6. We have released a part of it to the public on the web[4].

We selected the written essays in the International Corpus Network of Asian Learners of English (hereafter, ICNALE) as our base corpus. Their essay topics are controlled; all essays are written on two common topics: (a) *It is important for college students to have a part-time job.* and (b) *Smoking should be completely banned at all the restaurants in the country.*, which hereafter will be referred to as *PART-TIME JOB* and *SMOKING*, respectively. This simulates the condition common to the language learning settings that essays are often written on one topic in language learning as in writing exercises in class and writing tasks in language proficiency tests. The writers are college students (including graduate students) from 10 countries and regions in Asian (although we only used the Chinese, Indonesian, Japanese, and Korean portions for time and cost reasons). Their proficiency levels are estimated to be from A2 to B2+ in the CEFR metric. As preprocessing, we split the essays into sentences and in turn tokenized them by using the Stanford Statistical Natural Language Parser (ver.2.0.3) (de Marneffe et al., 2006).

[4]The corpus data are available at `https://www.gsk.or.jp/en/catalog//`.

Before annotation, we had to choose in which language we would create feedback comments. Basically, the choice would be either English or the writer's native language. We chose Japanese, one of the writers' native languages, for the following reasons: (1) Beginner to intermediate learners may have difficulty in understanding feedback comments in English when working on writing exercises; (2) it will likely be more technically challenging and interesting to generate feedback comments on English in a different language; (3) it would be too costly to create feedback comments in all the native languages and accordingly we chose the one we can understand. For accessibility, we manually translated them into English for future research. In this work, we only used the Japanese feedback comments.

In addition, we created two special symbols to augment feedback comments: grammatical term ($<$, $>$) and citation ($<<$, $>>$). Grammatical terms are tagged with $<$ and $>$ as in $<$*intransitive verb*$>$. With this, one can make links to corresponding grammatical items in a grammar book, for example, as an additional source of information for the user[5]. Citation is used to denote that the word(s) inside the symbols is cited from the commented sentence as in $<<$*agree*$>>$. This makes feedback comments more flexible. That is, the word(s) inside $<<$ and $>>$ in a generated feedback comment may be replaced with another word(s) in the commented sentence.

This is the big picture of how we created the data. More details are described in the guideline accompanied with the dataset[6].

We hired two professional annotators who had a good command of English. Both of them have had experience in English syntactic annotation for more than ten years; one of them had two years of professional English writing teaching experience where she provided high school and college students with feedback comments as in this work.

We sampled out 1,040 essays from ICNALE and assigned them to either annotator. The annotators read an entire essay first one at a time and then added feedback comments to all preposition errors they found, plus other parts on which they wanted to annotate (e.g., praise). They used the commenting function in MS-Word for this annotation.

[5]We have also created a grammar database with a list of grammar items and their explanations. However, it is not included in the dataset due to copyright issues.

[6]The guideline is included in the dataset.

Corpus	PART-TIME JOB				SMOKING			
Split	Total	Training	Dev.	Test	Total	Training	Dev.	Test
# essays	521	371	75	75	519	369	75	75
# sentences	7,989	5,615	1,215	1,159	7,979	5,665	1,206	1,180
# tokens	131,947	93,896	19,261	18,781	130,154	92,419	19,017	18,709
# comments	1,408	1,008	210	190	1,426	1,034	208	184
comments/sent.	0.18	0.18	0.17	0.16	0.18	0.17	0.17	0.17

Table 1: Statistics on the created dataset.

Finally, we split 1,040 essays into training, development, and test sets as shown in Table 1[7]. We transformed the MS-Word format into a TSV format (learner sentence, offset, feedback comment[8]) and used the resulting essays for evaluation.

5 Baseline Methods

To examine the feasibility of feedback comment generation for preposition use, we implemented the following three simple baseline methods: a neural retrieval-based method, a sequence-to-sequence model, and a rule-based method. The following subsections describe each method in detail.

5.1 Neural Retrieval-based Method

This method solves the task as a feedback comment retrieval problem. The procedure consists of:
(0) Preprocessing
(I) Feedback comment vector encoding
(II) Detection and context vector extraction
(III) Mapping between the two vectors
(IV) Feedback comment retrieval
(V) Output
The main steps (I)–(IV) are depicted in Fig. 2.

(0) Preprocessing is applied to the input essay in both training and prediction. First, it is split into sentences. Then, each sentence is tokenized and put into lowercase. Finally, tokens whose occurrence is lower than a certain threshold are replaced with the special symbol <unk>. The same procedures except sentence splitting are applied to feedback comments in the training data.

In **(I) Feedback comment encoding**, the preprocessed feedback comments are encoded as vectors (Fig. 2 (I)) by using LSTM-based Language Models (LSTMLMs). Tokens in a feedback comment are turned into their corresponding word embeddings in an embedding layer and then passed on to an LSTM, which is trained to predict the next token. After training, all feedback comments are once again passed through the trained network. Its final hidden state of the final layer, which is merely a vector, can be regarded as an abstract representation of a feedback comment. In addition, another LSTMLM of the same architecture, which takes as input each token in reverse order, is used to augment the representation power. The two final states are simply concatenated to produce a vector for the input feedback comment. In the end, all feedback comments are abstractly represented as vectors, which will be used later in (III).

(II) Detection and context vector extraction consist of two subtasks. The first is to detect where to comment in the input essay. Hereafter, each sentence in the input essay is processed (namely, given to the network) one at a time. Under this condition, the subtask can be regarded as a binary sequence labeling problem in which input and output are a sequence of tokens and a sequence of 0/1 (to comment or not), respectively as in the sequence labelling-based error detection (e.g., Rei and Yannakoudakis (2016); Rei (2017); Kaneko et al. (2017)).

Following the work (Kaneko et al., 2017), this paper uses a BiLSTM-based sequence labeling model as shown in Fig. 2 (II). To be precise, each token is transformed into its corresponding word embedding, then is given to a BiLSTM, and finally to a softmax layer that estimates the probabilities of 0 and 1. During training, the information about where to comment is given as offsets. For simplicity, we used word-based offsets. In addition, if a feedback comment refers to more than one words, we used the position of the central word as its offset. The loss function to minimize is simply the cross-entropy loss. During prediction, each token

[7]We are still conducting the annotation over ICNALE and another learner corpus (Konan-JIEM learner corpus (Nagata and Sakaguchi, 2016)). At the time of submission, the number of annotated essays has reached 1,900.

[8]The offset and feedback comment columns may be none or repeated an arbitrary number of times.

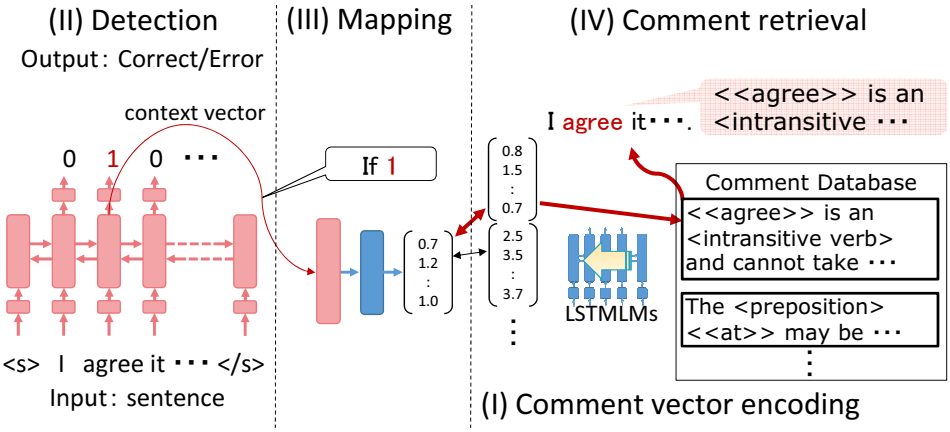

Figure 2: Neural Retrieval-based Method.

in the target sentence is given to the network to estimate the probabilities of 0 and 1.

The second subtask is to represent the surrounding context of the detected word abstractly. This is actually already done in the course of the first subtask. Namely, the corresponding BiLSTM hidden states can be used as context vectors because at each time step, the information about its surrounding words are recurrently accumulated in them.

In **(III) Mapping between the two vectors**, the detected context vectors are mapped onto the feedback comment vector space[9]. This is done by a Feed Forward Neural Network (FFNN) (Fig. 2 (III)). The purpose is to map context vectors near the corresponding comment vectors in the space. Accordingly, the mean square error can be naturally used as a loss function. Note that the whole networks (BiLSTM for sequence labeling in (II) and FFNN for vector mapping in (III)) are trained simultaneously (i.e., multi-task learning). The total loss l is defined by $l = (1-\beta)l_{\mathrm{s}} + \beta l_{\mathrm{m}}$ where l_{s}, l_{m}, and β are the losses for sequence labeling and vector mapping, and a hyperparameter that controls the weighting between the two, respectively.

(IV) Feedback comment retrieval and the following procedures are applied only during prediction. After (II) and (III), words to be commented on are detected and their context vectors are transformed into feedback comment vectors. This means that one can retrieve the most appropriate feedback comments by searching the space for feedback comment vectors similar to the mapped context vector. The similarity is measured by the cosine similarity between two vectors.

To improve performance, it is important to filter out inappropriate results. In some cases, a discrepancy occurs between the word detected in (II) (say, *at*) and the word to which the retrieved comment applies in the training data (say, *in*). In that case, the retrieved feedback comment will likely be inappropriate no matter how high its similarity is (in the above example, a feedback comment concerning *in* is applied to *at*). Consequently, these feedback comments are excluded from the results. In addition, the cosine similarity should be a good indicator to determine whether the retrieved feedback comments are appropriate or not, and thus retrieved feedback comments are discarded if their similarity is lower than a certain threshold. In other words, we do not generate feedback comments if we cannot find similar feedback comments in the training data.

In **(V) Output**, the most appropriate feedback comment (according to their similarities) for each commented word is displayed to the user.

5.2 Sequence-to-sequence Model-based Method

This method is an implementation of the sequence-to-sequence (encoder-decoder) model with an attention mechanism (Bahdanau et al., 2014)). It takes a learner sentence as input. It then puts the input sentence into a feedback comment.

Each word in the input learner sentence, which is preprocessed just as in the neural retrieval-based method, is encoded by a word embedding layer and a BiLSTM-based encoder. Then, a feedback

[9]Recall all comments in the training data have already been transformed into vectors in (I).

comment is generated by another word embedding layer and an LSTM-based decoder with an attention mechanism.

During the development phase, we observed that the network trained on the entire training data almost always resulted in generating no feedback comments. We observed a similar tendency when it was trained with a fixed feedback comment for the correct sentences (such as *There is no preposition error in this sentence.*). Considering this, we trained (and tested) the sequence-to-sequence model-based method on only sentences that had feedback comments to show the upper bound of this method.

5.3 Case frame-based Method

This method is an implementation of the case frame-based method (Nagata et al., 2014). The method automatically induces case frames by comparing learner and native corpora, to detect preposition errors with feedback comments.

We apply the exact same preprocessing as in the neural retrieval-based method to both learner and native corpora to induce case frames. We then manually created feedback comments for each induced case frame.

6 Evaluation

6.1 Conditions and Procedures

We implemented and trained the baseline methods with the created dataset. We first obtain word embeddings for learner sentences from the corpora as shown in Appendix A. We also used the word embeddings for English words in the LSTMLMs to encode feedback comments. For the rest (i.e., Japanese words), we initialized them using random-valued vectors. We used MeCab[10] to tokenize the feedback comments. With these word embeddings, we trained the networks on the training set of the respective subsets (PART-TIME JOB and SMOKING). We implemented the case frame-based method with the following corpora: British National Corpus (BNC) (Burnard, 1995), the EDR corpus (Japan electronic dictionary research institute Ltd, 1993) as a native corpus and the training and development set of the corresponding dataset[11] as a learner corpus. As a result, we obtained two versions of each method. We determined the hyperparameters by using the corresponding development set[12]. We tested the resulting models on the corresponding test set.

For evaluation, one of the professional annotators, who had been involved in the data creation, manually evaluated the results. She labeled each generated feedback comment as *appropriate*, *partially appropriate*, or *inappropriate*; *partially appropriate* refers to the case that the feedback comment is not completely appropriate but would become appropriate if part of it were edited.

We measured performance by recall, precision. We also used F-measure targeting *appropriate* feedbacks. We calculated their values considering only *appropriate* as correct generation. We also used their relaxed versions regarding *partially appropriate* as appropriate generation.

6.2 Results

Table 2 shows the results. It turns out that the simple application of the sequence-to-sequence model does not work well at all on this task; note that it is provided with the information about which sentence should have a feedback comment (i.e., tested on only the sentences having feedback comments). Nevertheless, its performance is very poor. This suggests that it requires modifications to achieve better performance with the sequence-to-sequence framework. *Case frame-based* successfully generates feedback comments in some cases. However, its recall is quite low. In contrast, the neural retrieval-based method achieves a far better performance in recall, achieving a precision comparable to that of *case frame-based*.

At the same time, Table 2 shows that there is still room for improvement. Subsect. 7.1 will investigate the generation results to reveal what has been solved by the methods.

[10] http://taku910.github.io/mecab/

[11] Because it does not require a development set for hyperparameter tuning, the development set was added to the training set.

[12] The hyperparameters are as follows: LSTMLMs: number of LSTM layers: 2, number of hidden states: 600; BiLSTM: number of LSTM layers: 1, number of hidden states: 600×2; FFNN: number of layers:1 (with dimension 1,200); BiLSTM encoder: number of LSTM layers:1, number of hidden states 600×2; LSTM decoder: number of LSTM layers:1, number of hidden states 600. Common hyperparameters: batch size: 64, number of epochs: 200 (with early stopping (patience: 10)), dropout rate: 0.5, optimization algorithm: Adam (step size: 0.01, the first and second moment: 0.9 and 0.999, respectively), threshold for <unk>: 5 in essays and 2 in feedback comments, threshold for the cosine similarity: 0.971 (PART-TIME JOB) and 0.976 (SMOKING), weight β for losses: 0.9.

Method	PART-TIME JOB			SMOKING		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Retrieval-based	0.23 (0.25)	0.61 (0.67)	0.34 (0.37)	0.28 (0.30)	0.72 (0.78)	0.41 (0.44)
Seq2seq	0.06 (0.07)	0.07 (0.08)	0.07 (0.08)	0.10 (0.13)	0.11 (0.13)	0.10 (0.13)
Case frame-based	0.10 (0.10)	0.62 (0.62)	0.16 (0.16)	0.05 (0.05)	0.75 (0.75)	0.09 (0.09)

Table 2: Generation Performance (numbers in brackets correspond to the relaxed measures).

7 Discussion

7.1 Analysis of Generated Comments

Looking into the generation results reveals that the neural retrieval-based method successfully generates feedback comments on errors typical to the topics (e.g., (S3) and (S4) in Fig. 3) and typical to argumentative essays (e.g., (S1) and (S7)), or more general ones (e.g., (S6)). While the superficial variations of these typical errors are large in general, they correspond to one or a few feedback comments. Actual examples are: *I/People (don't) agree the/this statement/opinion/that/thinking* and *(more) harmful (not only) for anyone/people/smokers/non-smokers*. These findings suggest that abstractly representing feedback comments and contexts and tying them in the feedback comment space, is effective in generating feedback comment on typical preposition errors.

It would probably be possible to create rule-based methods covering such typical errors. It would, however, take the expertise of at least a teacher of English and also a computer engineer to achieve it; the former would have to think of what is typical in the given topic and then to make a set of rules; the latter then would have to turn them into computer-readable forms. In contrast, the neural retrieval-based method only requires a teacher of English to annotate a given corpus with feedback comments, without examining what is typical, which is much more effective and efficient.

More importantly, there exist typical errors to which rules-based methods do not apply well. Examples include the word *Have* in (S2) and the word *include* in (S6) in Fig. 3. On the one hand, it would require a successful parsing to recognize the sources of the errors. On the other hand, it would require successful error detection to parse them correctly. Accordingly, rule-based methods, which normally require parsing, fail in generation in these cases. In contrast, the neural retrieval-based method gives a simple solution to this *parsing-first* or *error detection-first* dilemma because it does not depend on parsing at all.

To see the upper bound of the performance of the neural retrieval-based method, we evaluated its performance using the oracle offsets (where to comment). As a result, the strict recall and precision respectively improved to 0.27 and 0.73 in PART-TIME JOB and 0.30 and 0.79 in SMOKING. We also evaluated its detection performance as another upper bound of performance; if we cannot detect errors, we will never be able to generate feedback comments for them. We obtained a recall of 0.27 and a precision of 0.71 in PART-TIME JOB and a recall of 0.31 and a precision of 0.79 in SMOKING. The comparison between the two upper bounds suggests that one can successfully generate feedback comments for detected errors. This is partly because we used the same training data for error detection and feedback comment generation.

All these findings show neural retrieval-based methods are promising to generate feedback comments for typical errors. It is important from the viewpoint of language learning assistance to be able to generate feedback comments for typical errors considering that feedback is normally given to typical errors first in language learning (rather than to rare, irregular errors). More sophisticated retrieval-based methods such as the work by Hashimoto et al. (2018) and Qiu et al. (2017) will likely improve generation performance.

7.2 Model Analysis

One would probably think of using an attention mechanism to improve the neural retrieval-based method. We actually investigated its effectiveness during the development phase. It turned out that the neural retrieval-based method with and without an attention mechanism performed similarly; to be precise, the F-measure (of comment word detection) of the one with an attention mechanism was often slightly worse.

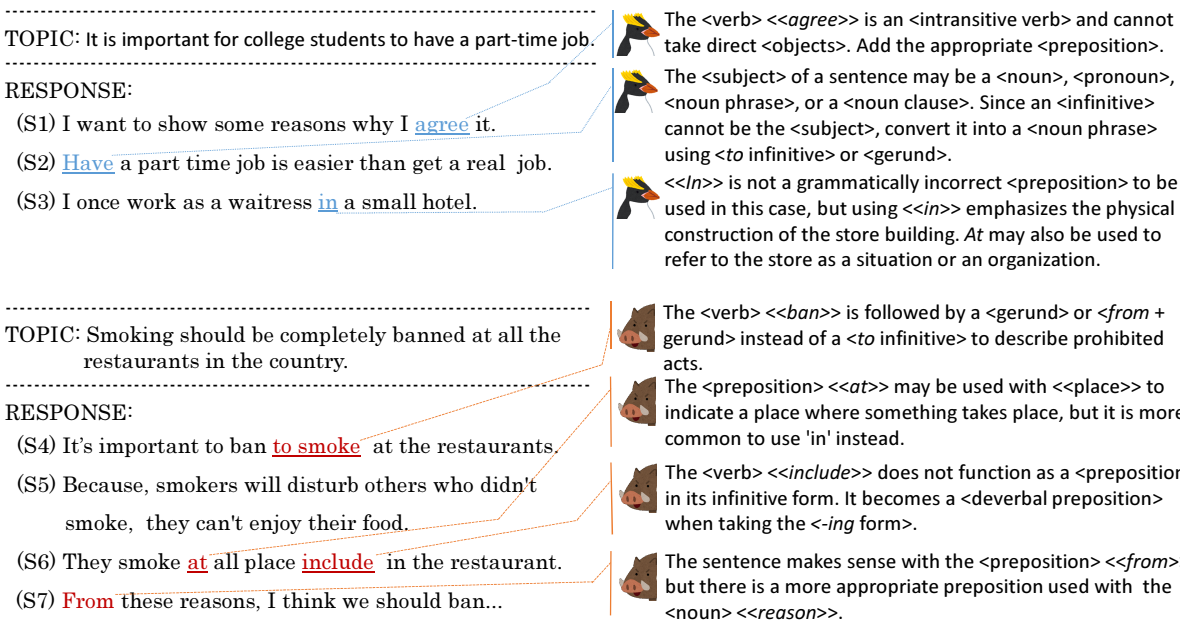

Figure 3: Example of Feedback Comments on Preposition Use Generated by Neural Retrieval-based Method.

This is partly because one can often tell from a narrow context why the usage of a given preposition is erroneous as in *agree it*. BiLSTM can handle well the information in such a narrow window without an attention mechanism. Having said that, we need further investigations to confirm that this argument is true; in particular, it might be necessary to generate feedback comments in general. Another reason would be that the amount of training data was not enough to train the attention layer, which also might be a reason why the sequence-to-sequence model (with an attention mechanism) does not perform well.

7.3 Unaddressed Problems

Sect. 6 has shown that feedback comments for preposition use can be automatically generated to some extent. The question now is whether the same argument can be made about feedback comment generation in general. Intuitively, it is a more difficult task because there are more variations in other grammatical errors. There are much more variations in feedback comments concerning other writing skills such as discourse and organization.

To answer the question, we have annotated and (are still annotating) learner corpora with feedback comments in general (together with those for preposition use). At the time of submission, the number of annotated essays has reached 2,300 (and will be 4,000 in the end) over two learner corpora (ICNALE and Konan-JIEM learner corpus (Nagata and Sakaguchi, 2016)). They range over a wide variety of writing skills including discourse, word choice, and organization. We will test the neural retrieval-based method on the dataset.

Another factor that makes the task hard is how to evaluate generated feedback comments. While, in this paper, we have applied manual evaluation, it is highly costly and time-consuming. For this reason, it would be difficult to make a thorough comparison between various methods or various model settings. One should have an efficient way of evaluating generated feedback comments to facilitate feedback comment generation research.

BLEU (Papineni et al., 2002) would be a choice for solving this problem as can be seen in machine translation research. As a pilot study, we estimated the validity of BLEU in this task. We randomly sampled out 702 feedback comments from the dataset and manually extracted pairs that had the same content. As a result, it turned out that the BLEU value ranged from as low as 0.15 to as high as 0.99; the average and the standard deviation were 0.61 and 0.25, respectively[13]. For comparison, we also calculated the BLEU value for pairs randomly chosen from the rest (that had not been paired previously); the average and the stan-

[13] We excluded the exact matches.

dard deviation were 0.13 and 0.09, respectively[14]. These numbers show that while BLEU can distinguish between appropriate and inappropriate feedback comments to some extent, it can be unreliable in some cases. All these observations suggest that we should have other evaluation methods together with BLEU.

8 Conclusions

In this paper, we introduced a novel task called feedback comment generation. We first described the task definition and created a dataset for the task of feedback comment generation for preposition use. We then tested a rule-based and two neural-based baselines on the dataset, showing that retrieval-based methods were promising. We further analyzed what the neural retrieval-based method can generate. We also discussed unaddressed problems. Especially, we pointed out that the evaluation was costly and time-consuming and that we needed a more efficient way of evaluation.

In future work, we will investigate how we can generate more flexible feedback comments. One way of achieving it is to apply a more sophisticated retrieval-based method such as (Hashimoto et al., 2018) to this task. We will also investigate how we can efficiently evaluate generation results.

Acknowledgments

We would like to thank the three anonymous reviewers for their useful comments on this paper. This work was supported by Japan Science and Technology Agency (JST), PRESTO Grant Number JPMJPR1758, Japan

A Corpora Used to Obtain Word Embeddings

Learner corpora: Corpus of English Essays Written by Japanese University Students (CEEEJUS)[15], ETS Corpus of non-native written English (Daniel Blanchard et al., 2014), The International Corpus of Learner English (ICLE) (Granger, 1993), Cambridge Learner Corpus (CLC) First Certificate in English (FCE) dataset (Yannakoudakis et al., 2011), and Nagoya Interlanguage Corpus of English (NICE) (Sugiura et al., 2007). Native corpus: English Web Treebank (EWT) (Bies, Ann, et al., 2012).

[14] We conducted the random sampling five times and took the average.

[15] http://language.sakura.ne.jp/s/doc/projects/CEEAUS.pdf

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv e-prints*.

Bies, Ann, et al. 2012. English web treebank LDC2012T13. Web Download.

John Bitchener, Stuart Young, and Denise Cameron. 2005. The effect of different types of corrective feedback on ESL student writing. *Journal of Second Language Writing*, 14(3):191–205.

Lou Burnard. 1995. *Users Reference Guide for the British National Corpus. version 1.0*. Oxford University Computing Services, Oxford.

Daniel Blanchard et al. 2014. ETS corpus of non-native written English, ldc2014t06. web download.

Rachele De Felice and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proc. of 22nd International Conference on Computational Linguistics*, pages 169–176.

Dana Ferris and Barrie Roberts. 2001. Error feedback in L2 writing classes: How explicit does it need to be? *Journal of Second Language Writing*, 10(3):161–184.

Sylviane Granger. 1993. The international corpus of learner English. In *English language corpora: Design, analysis and exploitation*, pages 57–69. Rodopi.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems 31*, pages 10052–10062.

Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. 2018. Deep code comment generation. In *Proc. of 26th Conference on Program Comprehension*, pages 200–210.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2073–2083.

Japan electronic dictionary research institute Ltd. 1993. *EDR electronic dictionary specifications guide*. Japan electronic dictionary research institute ltd.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proc. of 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606.

Jun'ichi Kakegawa, Hisayuki Kanda, Eitaro Fujioka, Makoto Itami, and Kohji Itoh. 2000. Diagnostic processing of Japanese for computer-assisted second language learning. In *Proc. of 38th Annual Meeting of the Association for Computational Linguistics*, pages 537–546.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error- and grammaticality-specific word embeddings. In *Proc. of 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 40–48.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of 5th International Conference on Language Resources and Evaluation*, pages 449–445.

Kathleen F. McCoy, Christopher A. Pennington, and Linda Z. Suri. 1996. English error correction: A syntactic user model based on principled "mal-rule" scoring. In *Proc. of 5th International Conference on User Modeling*, pages 69–66.

Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of English. In *Proc. of 44th Annual Meeting of the Association for Computational Linguistics*, pages 241–248.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner English. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics*, pages 1837–1847.

Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764.

Courtney Napoles and Chris Callison-Burch. 2017. Systematically adapting machine translation for grammatical error correction. In *Proc. of 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 345–356.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. AliMe Chat: A sequence to sequence and rerank based chatbot engine. In *Proc. of 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 498–503. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proc. of 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191.

Thomas Robb, Steven Ross, and Ian Shortreed. 1986. Salience of feedback on error and its effect on EFL writing quality. *TESOL QUARTERY*, 20(1):83–93.

Younghee Sheen. 2007. The effect of focused written corrective feedback and language aptitude on ESL learners' acquisition of articles. *TESOL Quarterly*, 41:255–283.

Masatoshi Sugiura, Masumi Narita, Tomomi Ishida, Tatsuya Sakaue, Remi Murao, and Kyoko Muraki. 2007. A discriminant analysis of non-native speakers and native speakers of English. In *Proc. of Corpus Linguistics Conference CL2007*, pages 84–89.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proc. of 22nd International Conference on Computational Linguistics*, pages 865–872.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proc. of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.