Verb Replacer: An English Verb Error Correction System

Yu-Hsuan Wu[1], Jhih-Jie Chen[2], Jason S. Chang[1]
[1]Institute of Information Systems and Applications
National Tsing Hua University
[2] Department of Computer Science
National Tsing Hua University
{shanny, jjc, jason}@nlplab.cc

Abstract

According to the analysis of Cambridge Learner Corpus, using a wrong verb is the most common type of grammatical errors. This paper describes *Verb Replacer*, a system for detecting and correcting potential verb errors in a given sentence. In our approach, alternative verbs are considered to replace the verb based on an error-annotated corpus and verb-object collocations. The method involves applying regression on channel models, parsing the sentence, identifying the verbs, retrieving a small set of alternative verbs, and evaluating each alternative. Our method combines and improves channel and language models, resulting in high recall of detecting and correcting verb misuse.

1 Introduction

It is estimated that over 1 billion people are learning English around the world, 600 to 700 million of which are English as a second language (ESL). Lacking lexical and collocation knowledge, ESL learners often have difficulties in choosing an appropriate word to fit the context.

Consider a learner's sentence "All Japanese children *accept* a solid education.". For most non-native English writers, this sentence may seem like an acceptable sentence. However, the verb *accept* is not appropriate and *receive* would be a better choice. Many learners misuse *accept* when they should use *receive* because these two verbs are semantically similar and have the same translation in learners native language. Therefore, it is difficult for learners to choose from the two to fit the context (i.e., the object *education*), leading to an awkward sentence.

According to the analysis of a sample of the Cambridge Learner Corpus (CLC) with 1,244 exam scripts for First Certificate English (FCE), verb selection errors (Replace-Verb errors, RV) is the most common error type, not counting spelling errors. In content word (e.g., verb and noun) errors correction, previous systems relied on mostly manually constructed resources (e,g., (Shei and Pain, 2000; Lee and Seneff, 2008; Liu et al., 2009)). It is not clear whether these manual resources can be easily scaled up and extended to other types of writing error and domains. Classifiers have been used for correcting verb errors. (Wu et al., 2010) describe an approach based on a classifier to predict the verb in the context of a given sentence. The main difference from our current work is that in(Wu et al., 2010), the context alone determine the outcome, the channel model information related to the potentially wrong verb is not used. Similarly, (Rozovskaya et al., 2014) use classifiers with the notion of verb finiteness to identify certain types of verb errors. (Rozovskaya et al., 2014) only address the agreement, tense, and form verb errors related to a small candidate set, while we deal with the verb selection problem with an open candidate set. In a noisy-channel approach closer to our work, (Sawai et al., 2013) use large learner corpus to construct candidate sets. They show that an GEC system that uses learner corpus outperforms systems that use WordNet and roundtrip translations, improving the performance of verb error detection and suggestion.

In this paper, we present a system, *Verb Replacer*, that uses both learner and web-scale corpora to extract errors to estimate the parameters in a channel model. Our system exploits the regularity of learner errors and a web-scale data set with a goal of maximizing the probability of an GEC system in returning alternatives for correcting misused verbs. An example *Verb Replacer* feedback

The Companion Volume of the IJCNLP 2017 Proceedings: System Demonstrations, pages 49–52,
Taipei, Taiwan, November 27 – December 1, 2017.

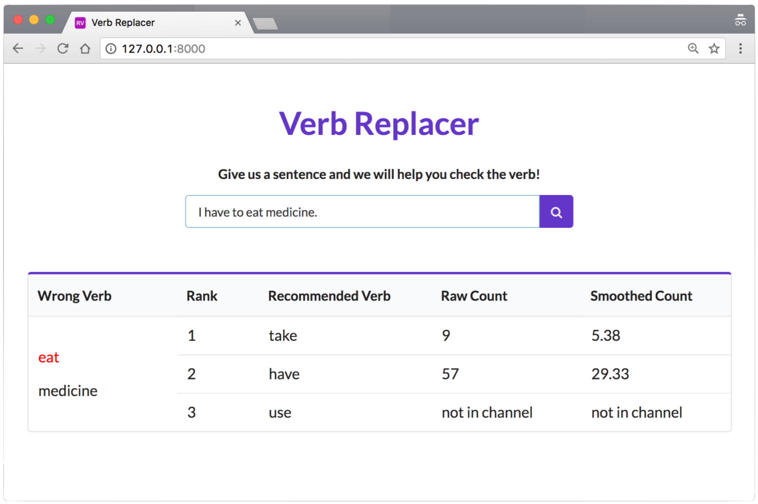

Figure 1: An example *Verb Replacer* search for input “I have to *eat* medicine.”

for the sentence “I have to *eat* medicine.” is shown in Figure 1.

The rest of this paper is organized as follows. We present our method for obtaining the verb alternatives, re-rank the alternatives and giving the correct suggestions in the next section. We introduce the data and discuss the experimental results in Section 3, and conclude with a summary and future work in Section 4.

2 Methodology

To correct verb misuse in a given sentence, a promising approach is to estimate quantitatively how words are typically misused based on a probabilistic channel model. In this section, we present our method for detecting and correcting RV errors.

2.1 Applying Regression Model

We use the regression model to deal with the data sparseness problem and to smooth the low counts of the channel model. To estimate the parameters of a channel model, we use an correction-annotated corpus to extract instances of Replace-Verb wrong-right pairs. However, some of these verb pairs have low counts, forcing the system to remove candidates with the same count and rank. Thus, we apply a regression model to smooth the low counts. We use Support Vector Regression (SVR) to train the regression model. The features used in the model are shown in Table 1. These features are based on the relationships between the wrong verb and each candidate verb.

There are five types of feature, including thesaurus-based similarity between the wrong verb and a candidate verb is calculated using a bilingual version of WordNet. We also use conjunction relation refer to the relationship related on the conjunctions *and* and *or* that link the wrong verb and the candidate verb. For a wrong verb X and a candidate verb Y, we first extract ngram with the patterns X *and* Y, X *or* Y, Y *and* X and Y *or* X from Google Web 1T Ngrams Corpus with the counts. Then we check whether both *and* and *or* link X and Y. Additionally, we use the proportion of these two types of patterns as features. Another information source for feature we use is translation. We use bilingual (English to Mandarin) data to find the translation of the wrong verb and a candidate verb, and count the number of translations they shared.

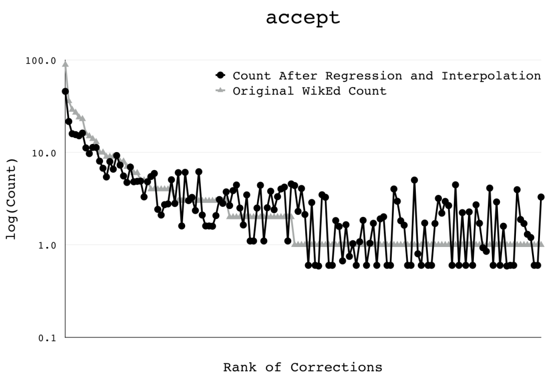

Figure 2: The log(count) before and after regression of a wrong verb ”*accept*” to the rank of its corrections

Once we have a regression model, we interpolate the new count with the original count, and a new estimate of count is given to each correction.

Feature	Description
Similarity	WordNet similarity between the wrong verb and a candidate verb
Conjunction relation	AND/OR relation between the wrong verb and a candidate verb
AND Proportion	Proportion of X and Y in all the patterns extracted through AND/OR relation
OR Proportion	Proportion of X or Y in all the patterns extracted through AND/OR relation
Translation	Number of the common words the wrong verb and a candidate verb share in Mandarin translation

Table 1: Features for regression model

Figure 2 shows an example of a wrong verb *accept* from an annotated reference corpus with the count and the rank of its corrections, before and after regression and interpolation.

2.2 Detecting and Correcting RV Errors

We attempt to build a candidate list based on a channel model and collocation list, which is then used to correct RV errors by reranking.

For the error-annotations in each sentence, if the error tag is RV, we keep the misuse verb. Otherwise, we remove the misuses and keep the correction in the sentence. After the sentences are replaced, we assign each token a POS tag. Tokens tagged as VERB are considered as potential errors in the next stage. For simplicity, we do not include auxiliary verbs such as *can*, *will* or *should*.

2.2.1 Building Candidate List

In this stage, we build a candidate list from a channel model and a collocation list. We rank the corrections in the channel model according to their new estimated counts. In order to improve the coverage, we use Google Web1T n-gram data to generate collocations for additional candidate verbs. If a Verb-Obj relation exists in a given learner sentence, the object will be extracted and used to find all of the verbs that are collocated with it. We then reorder the list based on sum of two reciprocal ranks.

2.2.2 Detecting RV Errors

In this stage, we evaluate each verb in the candidate list and rerank them based on a language model. First, the potentially wrong verb in the given sentence is replaced in turn by each verb in the candidate list. Then, the replaced sentences are evaluated based on a language model. We use two trigram language models, trained on a corrected learner corpus and a reference native corpus using SRILM ((Stolcke et al., 2002)), separately. We order the verb in the candidate list according to the log probability provided by the language model. We set a threshold t, and if the original verb ranks lower than t, the sentence will be returned in the next stage.

2.2.3 Reranking Alternatives

In the final stage, we rerank the alternatives, and suggest appropriate verbs to be returned to learner. For each verb in candidate list, we sum up the score from candidate list itself and the score from language model. Then we rerank the alternatives to suggest top 3 verbs to the user.

3 Experiments and Results

In this section, we describe the training data, development data, and test data we use for the experiments, and introduce the evaluation metrics we use for evaluating the performance of our system. We also show the experimental results.

3.1 Dataset

Wiked Error Corpus (WEC): WEC is a corpus of corrective Wikipedia revision logs. We used these revision edits for estimating the channel model. In total, 480,243 RV wrong-right pairs are extracted from WEC.

The EF-Cambridge Open Language Database (EFCAMDAT): The EFCAMDAT is an English L2 database, we used it for estimating the channel model. These essays were written by English learners, while WEC is composed by native and nonnative domain experts. We obtained around 113,000 RV errors from the dataset.

CLC-FCE Dataset: CLC-FCE Dataset is a collection of essays written by English language learners from around the world. Potential errors have been tagged with the CLC error coding scheme with corrections. We use CLC-FCE for developing and testing. We extracted 3,580 Replace-Verb (RV) errors for test.

3.2 Results for RV Error Detection

Figure 3 shows the results of RV error detection produced by the EFCAM-REG system at threshold t, varying t. If a given sentence with the original verb rank lower than t, the sentence will be handled in the correction step. As we can see in Figure 3, the higher the threshold is set, the higher precision the system can achieve. At threshold 5, the system has the highest F1 score.

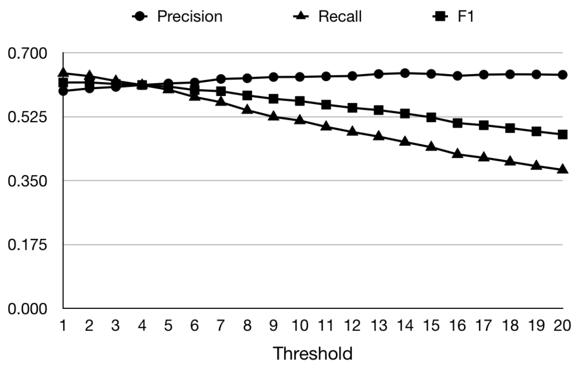

Figure 3: Precision, Recall and F1 score for RV error detection by the EFCAM-REG system at threshold t

3.3 Results for Verb Suggestion

To evaluate performance of suggestion for all erroneous verbs, we use Mean Reciprocal Rank (MRR). In our case, the measure is used to evaluate the Top 3 returned verbs for a given sentence. The MRR is the average of the reciprocal ranks of results for a set of sentences S:

$$MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{rank_i} \quad (1)$$

where $rank_i$ refers to the rank position of the gold standard for the $i - th$ sentence.

The results are shown in Table 2. We compare the systems that using WEC and EFCAMDAT channel model estimating. The results show that the systems with a regression-based channel model (**WEC-REG** and **EFCAM-REG**) perform better than those without regression (**WEC** and **EFCAM**). It is interesting to note that for the top 3 suggestions, using a learner corpus for the channel model estimation plus channel model regression (**EFCAM-REG**) performs the best. Also note that **EFCAM-REG** with the language model trained on the corrected part of EFCAMDAT (**EFCAM-REG-EFLM**) performs the best in terms of offering good suggestions.

4 Conclusion

In summary, we have introduced a new method for detecting and correcting Replace-Verb errors in a given learner sentence based on wrong-right verb pairs in annotated corpora. The analysis shows that our method, combining channel and language models, perform better than without using channel models. The results also show that using a learner corpus for RV error correction achieve better performance than using native reference corpus.

Many avenues exist for future research and improvement of the proposed method. For example, an interesting direction to explore is to use error-annotated sentences to train a sequence to sequence neural network to predict an replace RV errors as well as other types of errors.

Table 2: MRR for verb suggestion over 1,300 sentences

Systems	MRR_3	MRR_{found}
WEC	0.181	0.336
WEC-REG	0.191	0.342
WEC-REG-EFLM	0.190	0.346
EFCAM	0.260	0.428
EFCAM-REG	**0.273**	0.432
EFCAM-REG-EFLM	0.271	**0.446**

References

John Lee and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182.

Anne Li-E Liu, David Wible, and Nai-Lung Tsao. 2009. Automated suggestions for miscollocations. In *Proceedings of the 4th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 47–50.

Alla Rozovskaya, Dan Roth, and Vivek Srikumar. 2014. Correcting grammatical verb errors. In *Proceedings of the 14th EACL*, pages 358–367.

Yu Sawai, Mamoru Komachi, and Yuji Matsumoto. 2013. A learner corpus-based approach to verb suggestion for ESL. In *Proceedings of the 51st ACL*, pages 708–713.

C-C Shei and Helen Pain. 2000. An ESL writer's collocational aid. *Computer Assisted Language Learning*, 13(2):167–182.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002.

Jian-Cheng Wu, Yu-Chia Chang, Teruko Mitamura, and Jason S Chang. 2010. Automatic collocation suggestion in academic writing. In *Proceedings of the 48th ACL*, pages 115–119.