Model Combination for Correcting Preposition Selection Errors*

Nitin Madnani **Michael Heilman** **Aoife Cahill**

660 Rosedale Road
Princeton, NJ 08541, USA
{nmadnani,mheilman,acahill}@ets.org

Abstract

Many grammatical error correction approaches use classifiers with specially-engineered features to predict corrections. A simpler alternative is to use n-gram language model scores. Rozovskaya and Roth (2011) reported that classifiers outperformed a language modeling approach. Here, we report a more nuanced result: a classifier approach yielded results with higher precision while a language modeling approach provided better recall. Most importantly, we found that a combined approach using a logistic regression ensemble outperformed both a classifier and a language modeling approach.

1 Introduction

In this paper, we compare methods for correcting grammatical errors. Much of the previous work on grammatical error detection and correction has studied methods based on statistical classifiers (Tetreault and Chodorow, 2008; De Felice and Pulman, 2009; Tetreault et al., 2010; Rozovskaya and Roth, 2010; Dahlmeier and Ng, 2011; Seo et al., 2012; Cahill et al., 2013). In particular, Tetreault and Chodorow (2008) and Tetreault et al. (2010) found that classifiers based on a variety of contextual linguistic features performed well, especially in terms of precision (i.e., avoiding false positives). Gamon (2010), however, reported that a language modeling approach substantially outperformed a classifier using contextual features. Finally, Rozovskaya and Roth (2011) found that a classifier outperformed a language modeling approach on different data, making it unclear which approach is best.

Much of the previous work has used well-formed text when training contextual classifiers due to the lack of large error-annotated corpora. Han et al. (2010) conducted experiments with a relatively small error-annotated corpus and showed that it outperformed a contextual classifier trained on well-edited text. More recently, Cahill et al. (2013) mined Wikipedia revisions to produce a large, publicly available error-annotated corpus and reported similar results on multiple, publicly available data sets.

Our goal in this paper is *not* to build a state-of-the-art system but rather to investigate the following research questions:

- Does a contextual classifier trained on error-annotated data outperform a language modeling approach?
- Can a classifier trained on error-annotated data and the language modeling approach be effectively combined?

With respect to the second question, Gamon (2010) previously reported that a combination of a contextual classifier trained on well-edited text and a language modeling approach outperformed each individual method. However, given that the performance of his classifier was lower than what has been reported on other datasets (Tetreault and Chodorow, 2008; Rozovskaya and Roth, 2011), we believe it is worth reinvestigating the merits of system combination but with publicly available data sets and with a classifier trained on error-annotated data instead of on well-edited text. This work differs from Susanto et al. (2014) in that we are interested in combining statistical models in order to more accurately correct individual preposition errors, while their work

*Michael Heilman is now a data scientist at Civis Analytics.

Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pages 136–141,
San Diego, California, June 16, 2016.

combined — at the sentence level — the outputs of multiple systems designed to correct *different types* of grammatical errors.

2 Task Description

In this paper, we focus on the task of detecting and correcting preposition selection errors in English writing — that is, errors where the writer selects the incorrect preposition for a given context. We consider 36 different prepositions (§3.1).

2.1 Evaluation Datasets

We use two data sets for evalution: (a) The CLC **FCE** dataset, which contains exam scripts written by English language learners for the Cambridge ESOL First Certificate in English (Yannakoudakis et al., 2011), with 20% held out for development, and (b) The **HOO** 2011 shared task data set, which contains excerpts of ACL papers manually annotated for grammatical errors (Dale and Kilgarriff, 2011). No HOO data was used for development.

2.2 Metrics

To evaluate performance, we compute precision, recall, and F_1 score for each dataset. Precision is the percentage of system corrections that are correct according to the gold standard, and recall is the percentage of the gold standard corrections that were correctly marked by the system. Our evaluation metric can be viewed as similar to a micro-averaged F_1 score for a multi-class document classification task where documents are the original prepositions, classes are the possible corrections, and only documents for ungrammatical prepositions have class labels. Our F_1 score is similar to the WAS evaluation scheme of Chodorow et al. (2012), except that we treat cases where the original preposition, system prediction, and gold standard all differ as false negatives. Chodorow et al. (2012) instead treat such cases as both false positives and false negatives, and as a result, the sum of true positives, false positives, true negatives, and false negatives does not equal the number of examples.

3 Methods

This section describes our implementations of the classifier, language modeling, and system combination approaches to preposition error correction.

3.1 Classifier

Our first system is a classifier trained on error-annotated data, following Cahill et al. (2013). The classifier uses logistic regression to solve a 36-way classification problem with one class per preposition (Tetreault and Chodorow, 2008). It includes 25 lexical and syntactic contextual features. It also includes a feature indicating the writer's original preposition. The classifier learns a conditional probability distribution $p_{CLS}(w|x)$ over prepositions w given the context x in which they appear.

To train this classifier, we used the preposition error corpus mined from revisions in an XML snapshot of Wikipedia (Cahill et al., 2013). The snapshot contained 8,735,890 articles and 288,583,063 revisions. The resulting data set consists of 7,125,317 prepositions and their sentence contexts. Of these prepositions, 1,027,643 were marked as errors and annotated with corrections.

We include a threshold parameter λ_{CLS}, tuned to maximize F_1 score on the development set. Let w_{orig} be the writer's preposition, let Q be the set of prepositions, and let $w_{alt} = \text{argmax}_{y \in Q - \{w_{orig}\}} p_{CLS}(y|x)$ (i.e., the best alternative that differs from the original). If $p_{CLS}(w_{alt}) - p_{CLS}(w_{orig}) > \lambda_{CLS}$, then the alternative is predicted (i.e., a correction is made). Otherwise, no change is made. We explored values ranging from 0 to 1, with steps of size .001.

3.2 Language Model

Our second system uses a language modeling approach. We use KenLM (Heafield, 2011) to estimate an unpruned model for $n = 6$ with modified Kneser-Ney smoothing (Chen and Goodman, 1998) on the text of all articles contained in a snapshot of English Wikipedia from June 2012 (68,356,743 sentences). We use this n-gram language model to obtain scores $g_{LM}(w, s, i) = \frac{\log_{10} p_{LM}(f(w,s,i))}{|s|+1}$, where w is the preposition to be scored, s is the writer's original sentence, i is the position of the original preposition in s, f is a function that returns a variant of s with the preposition at i replaced with w, and p_{LM} returns the probability for a sentence. We divide the language model log probability by $|s| + 1$, where $|s|$ is the number of tokens in the sentence, to account for

differences in sentence lengths.[1]

Again, we include a threshold λ_{LM} for deciding whether to replace the writer's original preposition with the best alternative preposition. This works similarly, except that it works with differences in the language model scores g_{LM} rather than differences in probabilities. We explored a grid with 45 manually selected points obtained by examining the percentiles for g_{LM} on the development set.[2]

3.3 System Combinations

We examine three methods for combining the language modeling and classifier approaches.

3.3.1 Heuristic

The first method is a simple heuristic combination intended to increase the recall of the classifier approach. We first tune the λ_{CLS} and λ_{LM} thresholds individually for the classifier and language model approaches to optimize F_1 score, as described above. Then, if the classifier predicts a correction, we return that as the final correction. If the classifier did not predict a correction but the language model did, then we return the language model's suggested correction. If neither predicts a correction, then we return the original preposition.

3.3.2 Interpolation

The second method combines the scores from the classifier and the language model, finds the best alternative to the original (i.e., a potential correction), and then applies a threshold to decide whether or not to make the correction.

Let $g_{LM}(w, s, i)$ be the language model score (§3.2) for the sentence s containing the preposition of interest w at position i, and let $g_{CLS}(w, s, i) = \log_{10} p_{CLS}(w|f_{CLS}(s, i))$, computed using the classifier, where f_{CLS} is a function that returns the contextual features for the classifier.[3]

For each original preposition w_{orig}, this method computes the difference between its classifier or language model score and the corresponding score for each alternative preposition w_{alt}:

$$\Delta_{CLS}(w_{orig}, w_{alt}, s, i) = g_{CLS}(w_{alt}, s, i) - g_{CLS}(w_{orig}, s, i)$$

$$\Delta_{LM}(w_{orig}, w_{alt}, s, i) = g_{LM}(w_{alt}, s, i) - g_{LM}(w_{orig}, s, i)$$

The method then computes an interpolated score for each alternative preposition as follows:

$$g_{INT}(w_{orig}, w_{alt}, s, i) = \alpha * \Delta_{CLS}(w_{orig}, w_{alt}, s, i) + (1 - \alpha) * \Delta_{LM}(w_{orig}, w_{alt}, s, i)$$

It then finds the best alternative $\hat{w}_{INT}$ to the writer's original preposition w_{orig} based on that score, i.e.,

$$\hat{w}_{INT} = \arg\max_{w_{alt}} g_{INT}(w_{orig}, w_{alt}, s, i),$$

and then predicts it as the final correction if $g_{INT}(w_{orig}, \hat{w}_{INT}, s, i) > \lambda_{INT}$. The interpolation parameter α and the threshold parameter λ_{INT} are tuned to maximize F_1 on the development set, with a search grid for α and λ_{INT} ranging from 0 to 1 with steps of size .01.

3.3.3 Ensemble Classifier

Finally, we evaluate an ensemble that uses scores from the classifier and language model as features.

Specifically, we use a logistic regression classifier to make binary predictions about whether or not to accept potential corrections from the revision classifier (§3.1). We give priority to the classifier since it had higher precision on the development set. Given an original preposition w_{orig} in a sentence s at position i, with $\hat{w}_{CLS}$ being the best alternative according to the revision classifier, the following features are considered by the logistic regression:

- binary features for each of the 36 possible values of the writer's original preposition. The feature for w_{orig} is set to 1, and the rest are set to 0.
- binary features for each of the 36 possible values of the best alternative to the writer's original preposition, according to the revision classifier. The feature for $\hat{w}_{CLS}$ is set to 1, and the rest are set to 0.

[1] Our language modeling approach differs from that of Rozovskaya and Roth (2011). We use a language model to compute probabilities for whole sentences, whereas they use one to derive feature weights for contexts around the writer's original preposition, which are used in a separate model.

[2] We also evaluated language models for n=3, 4, 5 on the development set, but we do not include them here since their performance was not as good as the 6-gram model.

[3] We take the logarithm of the classifier probability here to put it on a similar scale to the language model score.

- $\Delta_{CLS}(w_{orig}, \hat{w}_{CLS}, s, i)$ (see §3.3.2)
- $\Delta_{LM}(w_{orig}, \hat{w}_{CLS}, s, i)$

Once trained, we obtain the probability of making a change for any given preposition $p(\text{change} = 1|w_{orig}, s, i)$ according to the logistic regression. We output $\hat{w}_{CLS}$ as the final prediction if $p(\text{change} = 1|w_{orig}, s, i) > \lambda_{ENS}$.

To tune λ_{ENS}, we use a procedure based on 10-fold cross-validation to obtain probabilities for the development set. Each fold (i.e., tenth) of the development set is iteratively held out, and the ensemble is trained on the remaining folds. The ensemble is then used to obtain probabilities of corrections for the examples in the held-out fold. Once we have probabilities for the whole set, we tune λ_{ENS} to maximize F_1 score, using a search grid ranging from 0 to 1 with steps of .001.

4 Results

Figure 1 shows the development set precision-recall curves for the revision classifier, the language modeling approach, the interpolation approach, and the ensemble approach. The heuristic approach is shown as a single point in the figure since there is no threshold to tune. For each curve, the figure also shows the point corresponding to the threshold that yields the optimal F_1 score. From these results, all system combination approaches seem useful for combining the outputs of the classifier and language model to balance precision and recall. The ensemble appears particularly effective: the ensemble system's precision is generally higher than the classifier and language model systems at the same levels of recall, and the ensemble's recall is generally higher at the same levels of precision.

Table 1 shows the performance of all methods on the FCE and HOO test sets. Note that to apply the ensemble method to a test set, we trained the ensemble on the entire development set and then computed probabilities of corrections for the test set instances. We observe the following:

1. For both the FCE and the HOO test sets, the classifier approach yields results with higher precision whereas language model approach provides better recall.
2. For the FCE test set, we find that the ensemble approach attains the best F_1 score and that it performs significantly better than all other approaches, including both the classifier and the language model.
3. For the HOO test set — which is quite different from the FCE data in both its genre (ACL papers) and the distribution of grammatical errors — the ensemble approach still attains the best performance and is significantly better than the language model and the heuristic system combination approach.

Dataset	System	P	R	F_1	Sig.
FCE	Classifier	**65.63**	17.77	27.97	$*$
	LM	28.42	30.39	29.37	$*$
	Heuristic	30.91	37.26	33.79	$*$
	Interpolation	50.67	36.30	42.30	$*$
	Ensemble	51.72	**38.35**	**44.04**	
HOO	Classifier	**59.26**	19.75	29.63	
	LM	12.90	14.81	13.79	$*$
	Heuristic	21.24	29.63	24.74	$*$
	Interpolation	34.29	29.63	31.79	
	Ensemble	32.93	**33.33**	**33.13**	

Table 1: P, R and F_1 scores for the FCE and HOO test sets. Bold indicates the best result for each metric. "$*$" indicates that the F_1 score for a system was significantly different ($p < .05$) from that of the ensemble system as per the BC_a Bootstrap (Efron and Tibshirani, 1993) test with 10,000 replications.

Finally, we also compared the ensemble approach to a current state-of-the-art preposition error correction system. To do this, we evaluated on the CoNLL 2013 shared task test set (Ng et al., 2013), which contains essays written by students at the National University of Singapore and manually annotated with grammatical errors. We use the "revised" version of the annotations that includes revisions submitted by participants after the initial evaluation and only evaluate preposition selection errors. We did not use any of this data for development. We compared our ensemble system to the system submitted by the team that performed best on preposition errors ("NAIST, PC"). Our ensemble obtained $F_1 = 14.24$, whereas the NAIST system obtained $F_1 = 7.56$. This difference is statistically significant ($p < 0.05$).

Note that these results are not directly comparable with the official results of the preposition error correction component of the CoNLL shared task. First, we only measured the performance of the two systems on preposition selection errors since our system is

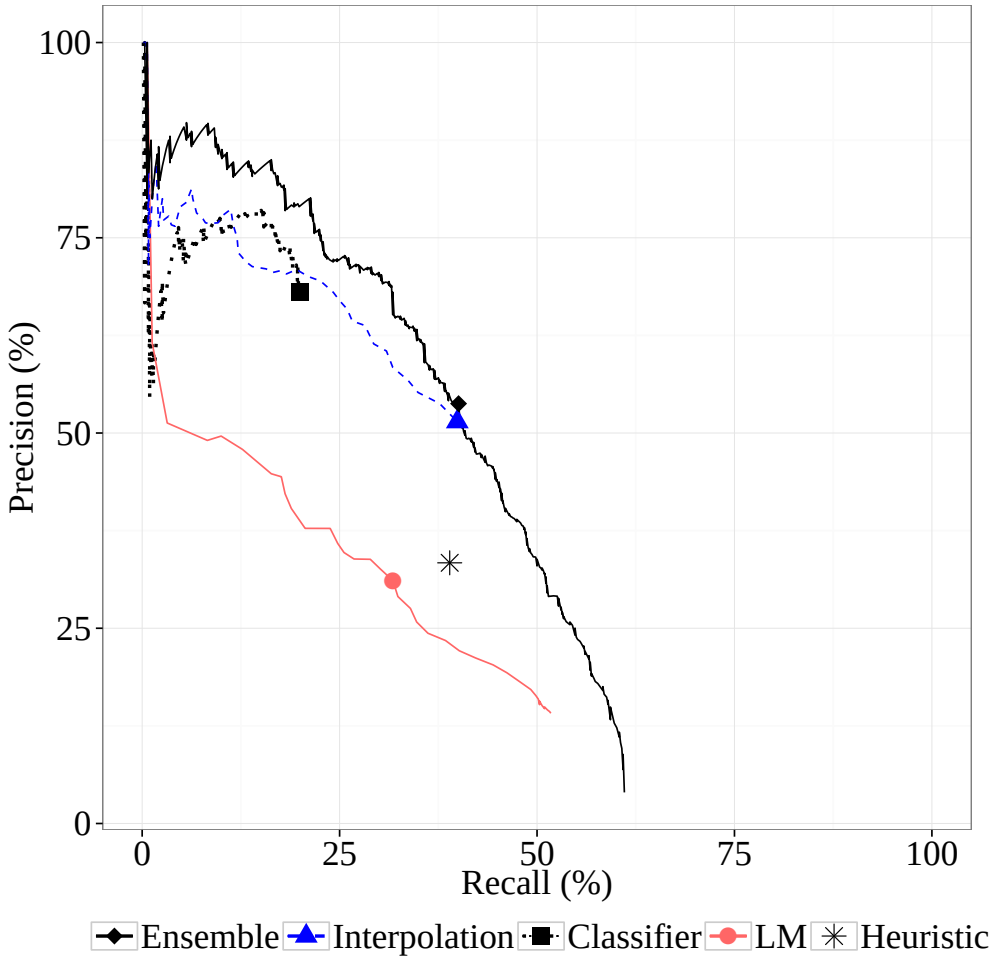

Figure 1: P and R values for the development set. Curves indicate performance at various values for the threshold λ for making a correction. Corresponding shaped dots indicate points at which F_1 is highest for each method.

not designed to correct either extraneous or missing preposition errors. Secondly, the F1 results on the CoNLL shared task used estimated types for computing precision, while were were certain of our error type. It is also not possible to compare to the published results on the HOO 2012 shared task (Dale et al., 2012) which used CLC-FCE data, because results for the three types of preposition errors were combined in one overall score.

5 Conclusions

Our goal in this paper was *not* to build a state-of-the-art preposition error correction system but rather to re-examine how well a simple language modeling approach performs on the task of correcting preposition selection errors, compared to the more typical approach that uses a classifier trained on a large error-annotated corpus (Cahill et al., 2013). We found that a language model does not generally perform as well as a classifier in terms of F_1, similar to a previous finding from Rozovskaya and Roth (2011). In addition, we also found that while the classifier has higher precision, the language model yields higher recall.

We also examined several methods for combining the classification and the language modeling approaches and found that a logistic regression ensemble is particularly effective. This ensemble significantly outperformed both the classifier and language modeling approaches on two publicly available test sets which indicates that more hybrid approaches should be investigated for grammatical error correction.

Acknowledgments

We would like to thank Lyle Ungar for his original suggestion, Kenneth Heafield for help with KenLM, Lei Chen, Anastassia Loukina, and the BEA reviewers.

References

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using wikipedia revisions. In *Proceedings of NAACL*, pages 507–517.

Stanley F. Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Harvard University.

Martin Chodorow, Markus Dickinson, Ross Israel, and Joel Tetreault. 2012. Problems in Evaluating Grammatical Error Detection Systems. In *Proceedings of COLING 2012*, pages 611–628.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical Error Correction with Alternating Structure Optimization. In *Proceedings of ACL*, pages 915–923.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 242–249.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada, June. Association for Computational Linguistics.

Rachele De Felice and Stephen G. Pulman. 2009. Automatic detection of preposition errors in learner writing. *CALICO Journal*, 26(3):512–528.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, Boca Raton, FL.

Michael Gamon. 2010. Using Mostly Native Data to Correct Errors in Learners' Writing. In *Proceedings of HLT-NAACL*, pages 163–171.

Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using Error-Annotated ESL Data to Develop an ESL Error Correction System. In *Proceedings of LREC*.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of CoNLL*.

Alla Rozovskaya and Dan Roth. 2010. Training Paradigms for Correcting Errors in Grammar and Usage. In *Proceedings of HLT-NAACL*, pages 154–162.

Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of ACL: HLT*, pages 924–933.

Hongsuck Seo, Jonghoon Lee, Seokhwan Kim, Kyusong Lee, Sechun Kang, and Gary Geunbae Lee. 2012. A Meta Learning Approach to Grammatical Error Correction. In *Proceedings of ACL (Short Papers)*, pages 328–332.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 951–962, Doha, Qatar, October. Association for Computational Linguistics.

Joel R. Tetreault and Martin Chodorow. 2008. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of COLING*, pages 865–872.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using Parse Features for Preposition Selection and Error Detection. In *Proceedings of ACL (Short Papers)*, pages 353–358.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of ACL*, pages 180–189.