The University of Illinois System in the CoNLL-2013 Shared Task

Alla Rozovskaya **Kai-Wei Chang** **Mark Sammons** **Dan Roth**
Cognitive Computation Group
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{rozovska,kchang10,mssammon,danr}@illinois.edu

Abstract

The CoNLL-2013 shared task focuses on correcting grammatical errors in essays written by non-native learners of English. In this paper, we describe the University of Illinois system that participated in the shared task. The system consists of five components and targets five types of common grammatical mistakes made by English as Second Language writers. We describe our underlying approach, which relates to our previous work, and describe the novel aspects of the system in more detail. Out of 17 participating teams, our system is ranked first based on both the original annotation and on the revised annotation.

1 Introduction

The task of correcting grammar and usage mistakes made by English as a Second Language (ESL) writers is difficult for several reasons. First, many of these errors are context-sensitive mistakes that confuse valid English words and thus cannot be detected without considering the context around the word. Second, the relative frequency of mistakes is quite low: for a given type of mistake, an ESL writer will typically make mistakes in only a small proportion of relevant structures. For example, determiner mistakes usually occur in 5% to 10% of noun phrases in various annotated ESL corpora (Rozovskaya and Roth, 2010a). Third, an ESL writer may make multiple mistakes in a single sentence, which may give misleading local cues for individual classifiers. In the example shown in Figure 1, the agreement error on the verb "tend" interacts with the noun number error on the word "equipments".

Therefore , the **equipments*/*equipment* of biometric identification **tend*/*tends* to be inexpensive .

Figure 1: Representative ESL errors in a sample sentence from the training data.

The CoNLL-2013 shared task (Ng et al., 2013) focuses on the following five common mistakes made by ESL writers:

- article/determiner
- preposition
- noun number
- subject-verb agreement
- verb form

Errors outside this target group are present in the task corpora, but are not evaluated.

In this paper, we present a system that combines a set of statistical models, where each model specializes in correcting one of the errors described above. Because the individual error types have different characteristics, we use several different approaches. The article system builds on the elements of the system described in (Rozovskaya and Roth, 2010c). The preposition classifier uses a combined system, building on work described in (Rozovskaya and Roth, 2011) and (Rozovskaya and Roth, 2010b). The remaining three models are all Naïve Bayes classifiers trained on the Google Web 1T 5-gram corpus (henceforth, Google corpus, (Brants and Franz, 2006)).

We first briefly discuss the task (Section 2) and give the overview of our system (Section 3). We then describe the error-specific components (Sections 3.1, 3.2 and 3.3). The sections describing individual components quantify their performance on splits of the training data. In Section 4,

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 13–19,
Sofia, Bulgaria, August 8-9 2013.

we evaluate the complete system on the training data using 5-fold cross-validation (hereafter, "*5-fold CV*") and in Section 5 we show the results we obtained on test.

We close with a discussion focused on error analysis (Section 6) and our conclusions (Section 7).

2 Task Description

The CoNLL-2013 shared task focuses on correcting five types of mistakes that are commonly made by non-native speakers of English. The training data released by the task organizers comes from the NUCLE corpus (Dahlmeier et al., 2013), which contains essays written by learners of English as a foreign language and is corrected by English teachers. The test data for the task consists of an additional set of 50 student essays. Table 1 illustrates the mistakes considered in the task and Table 2 illustrates the distribution of these errors in the released training data and the test data. We note that the test data contains a much larger proportion of annotated mistakes. For example, while only 2.4% of noun phrases in the training data have determiner errors, in the test data 10% of noun phrases have mistakes.

Error type	**Percentage of errors**	
	Training	**Test**
Articles	2.4%	10.0%
Prepositions	2.0%	10.7%
Noun number	1.6%	6.0%
Subject-verb agreement	2.0%	5.2%
Verb form	0.8%	2.5%

Table 2: **Statistics on error distribution in training and test data.** Percentage denotes the erroneous instances with respect to the total number of relevant instances in the data. For example, 10% of noun phrases in the test data have determiner errors.

Since the task focuses on five error types, only annotations marking these mistakes were kept. Note that while the other error annotations were removed, the errors still remain in the data.

3 System Components

Our system consists of five components that address individually article[1], preposition, noun verb form and subject-verb agreement errors.

Our article and preposition modules build on the elements of the systems described in Rozovskaya and Roth (2010b), Rozovskaya and Roth (2010c) and Rozovskaya and Roth (2011). The article system is trained using the Averaged Perceptron (AP) algorithm (Freund and Schapire, 1999), implemented within Learning Based Java (Rizzolo and Roth, 2010). The AP system is trained using the *inflation* method (Rozovskaya et al., 2012). Our preposition system is a Naïve Bayes (NB) classifier trained on the Google corpus and with prior parameters adapted to the learner data.

The other modules – those that correct noun and verb errors – are all NB models trained on the Google corpus.

All components take as input the corpus documents preprocessed with a part-of-speech tagger[2] and shallow parser[3] (Punyakanok and Roth, 2001). Note that the shared task data already contains comparable pre-processing information, in addition to other information, including dependency parse and constituency parse, but we chose to run our own pre-processing tools. The article module uses the POS and chunker output to generate some of its features and to generate candidates (likely contexts for missing articles). The other system components use the pre-processing tools only as part of candidate generation (e.g., to identify all nouns in the data for the noun classifier) because these components are trained on the Google corpus and thus only employ word n-gram features.

During development, we split the released training data into five parts. The results in Sections 3.1, 3.2, and 3.3 give performance of 5-fold CV on the training data. In Section 4 we report the development 5-fold CV results of the complete model and the performance on the test data. Note that the performance reported for the overall task on the test data in Section 4 reflects the system that makes use of the entire training corpus. It is also important to remark that only the determiner system is trained on the ESL data. The other models are trained on native data, and the ESL training data is only used to optimize the decision thresholds of the models.

[1]We will use the terms 'article-' and 'determiner errors' interchangeably: article errors constitute the majority of determiner errors, and we address only article mistakes.

[2]http://cogcomp.cs.illinois.edu/page/software_view/POS

[3]http://cogcomp.cs.illinois.edu/page/software_view/Chunker

Error type	Examples
Article	"It is also important to create *a/∅ better material that can support *the/∅ buildings despite any natural disaster like earthquakes."
Preposition	"As the number of people grows, the need *of/for habitable environment is unquestionably essential.
Noun number	Some countries are having difficulties in managing a place to live for their *citizen/citizens as they tend to get overpopulated."
Subject-verb agreement	"Therefore , the equipments of biometric identification *tend/tends to be inexpensive.
Verb form	"...countries with a lot of deserts can terraform their desert to increase their habitable land and *using/use irrigation.." "it was not *surprised/surprising to observe an increasing need for a convenient and cost effective platform."

Table 1: **Example errors.** Note that only the errors exemplifying the relevant phenomena are marked in the table; the sentences may contain other mistakes. Errors marked as verb form include multiple grammatical phenomena that may characterize verbs.

3.1 Determiners

There are three types of determiner error: omitting a determiner; choosing an incorrect determiner; and adding a spurious determiner. Even though the majority of determiner errors involve article mistakes, some of these errors involve personal and possessive pronouns.[4] Most of the determiner errors, however, involve omitting an article (these make up over 60% in the training data). Similar error patterns have been observed in other ESL corpora (Rozovskaya and Roth, 2010a).

Our system focuses on article errors. The system first extracts from the data all articles, and all spaces at the beginning of a noun phrase where an article is likely to be omitted (Han et al., 2006; Rozovskaya and Roth, 2010c). Then we train a multiclass classifier with features described in Table 3. These features were used successfully in previous tasks in error correction (Rozovskaya et al., 2012; Rozovskaya et al., 2011).

The original word choice (the source article) used by the writer is also used as a feature. Since the errors are sparse, this feature causes the model to abstain from flagging a mistake, which results in low recall. To avoid this problem, we adopt the approach proposed in (Rozovskaya et al., 2012), the *error inflation* method, and add artificial article errors in the training data based on the error distribution on the training set. This method prevents the source feature from dominating the context features, and improves the recall of the system.

We experimented with two types of classifiers: Averaged Perceptron (AP) and an L1-generalized logistic regression classifier (LR). Since the article system is trained on the ESL data, of which we have a limited amount, we also experimented with adding a language model (LM) feature to the LR learner. This feature indicates if the correction is accepted by a language model trained on the Google corpus. The performance of each classifier on 5-fold CV on the training data is shown in Table 4. The results show that AP performs better than LR. We observed that adding the LM feature improves precision but results in lower F1, so we chose the AP classifier without the LM feature for our final system.

Model	Precision	Recall	F1
AP (inflation)	0.17	0.31	0.22
AP (inflation+LM)	0.26	0.15	0.19
LR (inflation)	0.17	0.29	0.22
LR (inflation+LM)	0.24	0.21	0.22

Table 4: **Article development results** Results on 5-fold CV. AP With Inflation achieves the best development using an inflation constant of 0.85. AP achieves higher performance without using the language model feature.

3.2 Prepositions

The most common preposition errors are replacements, i.e., where the author correctly recognized the need for a preposition, but chose the wrong one to use.

[4] e.g. "Pat apologized to me for not keeping *the*/*my* secrets."

Feature Type	Description
Word n-grams	wB, w_2B, w_3B, wA, w_2A, w_3A, $wBwA$, w_2BwB, wAw_2A, w_3Bw_2BwB, w_2BwBwA, $wBwAw_2A$, wAw_2Aw_3A, $w_4Bw_3Bw_2BwB$, w_3w_2BwBwA, $w_2BwBwAw_2A$, $wBwAw_2Aw_3A$, $wAw_2Aw_3w_4A$
POS features	pB, p_2B, p_3B , pA, p_2A, p_3A, $pBpA$, p_2BpB, pAp_2A, $pBwB$, $pAwA$, p_2Bw_2B, p_2Aw_2A, p_2BpBpA, $pBpAp_2A$, pAp_2Ap_3A
NP_1	$headWord$, *npWords*, *NC*, *adj&headWord*, *adjTag&headWord*, *adj&NC*, *adjTag&NC*, *npTags&headWord*, *npTags&NC*
NP_2	*headWord&headPOS*, *headNumber*
wordsAfterNP	*headWord&wordAfterNP, npWords&wordAfterNP, headWord&2wordsAfterNP, npWords&2wordsAfterNP, headWord&3wordsAfterNP, npWords&3wordsAfterNP*
wordBeforeNP	$wB\&f_i \; \forall i \in NP_1$
Verb	$verb$, $verb\&f_i \; \forall i \in NP_1$
Preposition	$prep\&f_i \; \forall i \in NP_1$
Source	the word used by the original writer
LM	a binary feature assigned by a language model

Table 3: **Features used in the article error correction system.** wB and wA denote the word immediately before and after the target, respectively; and pB and pA denote the POS tag before and after the target. *headWord* denotes the head of the NP complement. NC stands for noun compound and is active if second to last word in the NP is tagged as a noun. *Verb* features are active if the NP is the direct object of a verb. *Preposition* features are active if the NP is immediately preceded by a preposition. *adj* feature is active if the first word (or the second word preceded by an adverb) in the NP is an adjective. *npWords* and *npTags* denote all words (POS tags) in the NP.

3.2.1 Preposition Features

All features used in the preposition module are lexical: word n-grams in the 4-word window around the target preposition. The NB-priors classifier, which is part of our model, can only make use of the word n-gram features; it uses n-gram features of lengths 3, 4, and 5. Note that since the NB model is trained on the Google corpus, the annotated ESL training data is used only to replace the prior parameters of the model (see Rozovskaya and Roth, 2011 for more details).

3.2.2 Training the Preposition System

Correcting preposition errors requires more data to achieve performance comparable to article error correction due to the task complexity (Gamon, 2010). We found that training an AP model on the ESL training data with more sophisticated features is not as effective as training on a native English dataset of larger size. The ESL training data contains slightly over 100K preposition examples, which is several orders of magnitude smaller than the Google n-gram corpus. We use the shared task training data to replace the prior parameters of the model (see Rozovskaya and Roth, 2011 for more details). The NB-priors model does not target preposition omissions and insertions: it corrects only preposition replacements that involve the 12 most common English prepositions. The task includes mistakes that cover 36 prepositions but we found that the model performance drops once the confusion set becomes too large. Table 5 shows the performance of the system on the 5-fold CV on the training data, where each time the classifier was trained on 80% of the documents.

Model	Precision	Recall	F1
NB-priors	0.14	0.14	0.14

Table 5: **Preposition results: NB with priors.** Results on 5-fold CV. The model is trained on the Google corpus.

3.3 Correcting Nouns and Verbs

The three remaining types of errors – noun number errors, subject-verb agreement, and the various verb form mistakes – are corrected using separate NB models also trained on the Google corpus. We focus here on the selection of candidates for correction, as this strongly affects performance.

3.3.1 Candidate Selection

This stage selects the set of words that are presented as input to the classifier. This is a crucial step because it limits the performance of any system: those errors that are missed at this stage have no chance of being detected by the later stages. This is also a challenging step as the class of verbs and nouns is open, with many English verbs and nouns being compatible with multiple parts of speech. This problem does not arise in preposition and article error correction, where candidates are determined by surface form (i.e. can be determined using a closed list of prepositions or articles).

We use the POS tag and the shallow parser output to identify the set of candidates that are input to the classifiers. In particular, for nouns, we collect all words tagged as NN or NNS. Since preprocessing tools are known to make more mistakes on ESL data than on native data, this procedure does not have a perfect result on the identification of all noun mistakes. For example, we

miss about 10% of noun errors due to POS/shallow parser errors. For verbs, we compared several candidate selection methods. Method (1) extracts all verbs heading a verb phrase, as identified by the shallow parser. Method (2) expands this set to words tagged with one of the verb POS tags {VB,VBN,VBG,VBD,VBP,VBZ}. However, generating candidates by selecting only those tagged as verbs is not good enough, since the POS tagger performance on ESL data is known to be suboptimal (Nagata et al., 2011), especially for verbs containing errors. For example, verbs lacking agreement markers are likely to be mistagged as nouns (Lee and Seneff, 2008). Erroneous verbs are exactly the cases that we wish to include. Method (3) adds words that are in the lemma list of common English verbs compiled using the Gigaword corpus. The last method has the highest recall on the candidate identification; it misses only 5% of verb errors, and also has better performance in the complete model. We thus use this method.

3.3.2 Noun-Verb Correction Performance

Table 6 shows the performance of the systems based on 5-fold CV on the training data. Each model is trained individually on the Google corpus, and is individually processed to optimize the respective thresholds.

Model	Precision	Recall	F1
Noun number	0.17	0.38	0.23
Subject-verb agr.	0.19	0.24	0.21
Verb form	0.07	0.20	0.10

Table 6: **Noun, subject-verb agreement and verb form results.** Results on 5-fold CV. The models are trained on the Google corpus.

4 Combined Model

In the previous sections, we described the individual components of the system developed to target specific error types. The combined model includes all of these modules, which are each applied to examples individually: there is no pipeline, and the individual predictions of the modules are then pooled.

The combined system also includes a post-processing step where we remove certain corrections of noun and verb forms that we found occur quite often but are never correct. This happens when both choices – the writer's selection and the correction – are valid but the latter is observed more frequently in the native training data. For example, the phrase "developing country" is changed to "developed country" even though both are legitimate English expressions. If a correction is frequently proposed but always results in a false alarm, we add it to a list of changes that is ignored when we generate the system output. When we generate the output on Test set, 8 unique pairs of such changes are ignored (36 pairs of changes in total).

We now show the combined results on the training data by conducting 5-fold CV, where we add one component at a time. Table 8 shows that the recall and the F1 scores improve when each component is added to the system. The final system achieves an F1 score of 0.21 on the training data in 5-fold CV.

Model	Precision	Recall	F1
Articles	0.16	0.12	0.14
+Prepositions	0.16	0.14	0.15
+Noun number	0.17	0.23	0.20
+Subject-verb agr.	0.18	0.25	0.21
+Verb form (All)	0.18	0.27	0.21

Table 7: **Results on 5-fold CV on the training data**. The article model is trained on the ESL data using AP. The other models are trained on the Google corpus. The last line shows the results, when all of the five modules are included.

5 Test Results

The previous section showed the performance of the system on the training data. In this section, we show the results on the test set. As previously, the performance improves when each component is added into the final system. However, we also note that the precision is much higher while the recall is only slightly lower. We attribute this increased precision to the observed differences in the percentage of annotated errors in training vs. test (see Section 3) and hypothesize that the training data may contain additional relevant errors that were not included in the annotation.

Besides the original official annotations announced by the organizers, another set of annotations is offered based on the combination of revised official annotations and accepted alternative annotations proposed by participants. We show in Table 8 when our system is scored based on the

revised annotations, both the precision and the recall are higher. Our system achieves the highest scores out of 17 participating teams based on both the original and revised annotations.

Model	Precision	Recall	F1
Scores based on the original annotations			
Articles	0.48	0.11	0.18
+Prepositions	0.45	0.12	0.19
+Noun number	0.48	0.21	0.29
+Subject-verb agr.	0.48	0.22	0.30
+Verb form (All)	**0.46**	**0.23**	**0.31**
Scores based on the revised annotations			
All	**0.62**	**0.32**	**0.42**

Table 8: **Results on Test**. The article model is trained on the ESL data using AP. The other models are trained on the Google corpus. *All* denotes the results of the complete model that includes all of the five modules.

6 Discussion and Error Analysis

Here, we present some interesting errors that our system makes.

6.1 Error Analysis

Incorrect verb form correction: *Safety is one of the crucial problems that many countries and companies* *concerned/concerns.

Here, the phrasing requires multiple changes; to maintain the same word order, this correction would be needed in tandem with the insertion of the auxiliary "*have*" to create a passive construction.

Incorrect determiner insertion: *In this era, Engineering designs can help to provide more habitable accommodation by designing a stronger material so it's possible to create a taller and safer building, a better and efficient sanitation system to prevent *∅/*the *disease, and also by designing a way to change the condition of the inhabitable environment.*

This example requires a model of discourse at the level of recognizing when a specific disease is a focus of the text, rather than disease in general. The use of a singular construction "*a taller and safer building*" in this context is somewhat unconventional and potentially makes this distinction even harder to detect.

Incorrect verb number correction: *One current human* *need/needs *that should be given priority is the search for renewable resources.*

This appears to be the result of the system heuristics intended to mitigate POS tagging errors on ESL text, where the word "*need*" is considered as a candidate verb rather thana noun; this results in an incorrect change to make the "verb" agree in number with the phrase "one human".

Incorrect determiner deletion: *This had shown that the engineering design process is essential in solving problems and it ensures that the problem is thoroughly looked into and ensure that the engineers are generating ideas that target the main problem, *the/∅ depletion and harmful fuel.*

In this example, local context may suggest a list structure, but the wider context indicates that the comma represents an appositive structure.

6.2 Discussion

Note that the presence of multiple errors can have very negative effects on preprocessing. For example, when an incorrect verb form is used that results in a word form commonly used as a noun, the outputs of the parsers tend to be incorrect. This limits the potential of rule-based approaches.

Machine learning approaches, on the other hand, require sufficient examples of each error type to allow robust statistical modeling of contextual features. Given the general sparsity of ESL errors, together with the additional noise introduced into more sophisticated preprocessing components by errors with overlapping contexts, it appears hard to leverage these more sophisticated tools to generate features for machine learning approaches. This motivates our use of just POS and shallow parse analysis, together with language-modeling approaches that can use counts derived from very large native corpora, to provide robust inputs for machine learning algorithms.

The interaction between errors suggests that constraints could be used to improve results by ensuring, for example, that verb number, noun number, and noun phrase determiner are consistent. This is more difficult than it may first appear for two reasons. First, the noun that is the subject of the verb under consideration may be relatively distant in the sentence (due to the presence of intervening relative clauses, for example). Second, the constraint only limits the possible correction options: the correct number for the noun in fo-

cus may depend on the form used in the preceding sentences – for example, to distinguish between a general statement about some type of entity, and a statement about a specific entity.

These observations suggest that achieving very high performance in the task of grammar correction requires sophisticated modeling of deep structure in natural language documents.

7 Conclusion

We have described our system that participated in the shared task on grammatical error correction and ranked first out of 17 participating teams. We built specialized models for the five types of mistakes that are the focus of the competition. We have also presented error analysis of the system output and discussed possible directions for future work.

Acknowledgments

This material is based on research sponsored by DARPA under agreement number FA8750-13-2-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This research is also supported by a grant from the U.S. Department of Education and by the DARPA Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-018.

References

T. Brants and A. Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA.

D. Dahlmeier, H.T. Ng, and S.M. Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proc. of the NAACL HLT 2013 Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, Atlanta, Georgia, June. Association for Computational Linguistics.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*.

M. Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *NAACL*, pages 163–171, Los Angeles, California, June.

N. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Journal of Natural Language Engineering*, 12(2):115–129.

J. Lee and S. Seneff. 2008. Correcting misuse of verb forms. In *ACL*, pages 174–182, Columbus, Ohio, June. Association for Computational Linguistics.

R. Nagata, E. Whittaker, and V. Sheinman. 2011. Creating a manually error-tagged and shallow-parsed learner corpus. In *ACL*, pages 1210–1219, Portland, Oregon, USA, June. Association for Computational Linguistics.

H. T. Ng, S. M. Wu, Y. Wu, Ch. Hadiwinoto, and J. Tetreault. 2013. The conll-2013 shared task on grammatical error correction. In *Proc. of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *NIPS*.

N. Rizzolo and D. Roth. 2010. Learning Based Java for Rapid Development of NLP Systems. In *LREC*.

A. Rozovskaya and D. Roth. 2010a. Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

A. Rozovskaya and D. Roth. 2010b. Generating confusion sets for context-sensitive error correction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A. Rozovskaya and D. Roth. 2010c. Training paradigms for correcting errors in grammar and usage. In *NAACL*.

A. Rozovskaya and D. Roth. 2011. Algorithm selection and model adaptation for esl correction tasks. In *ACL*.

A. Rozovskaya, M. Sammons, J. Gioja, and D. Roth. 2011. University of Illinois system in HOO text correction shared task.

A. Rozovskaya, M. Sammons, and D. Roth. 2012. The UI system in the hoo 2012 shared task on error correction.