The UI System in the HOO 2012 Shared Task on Error Correction

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

Abstract

We describe the University of Illinois (UI) system that participated in the Helping Our Own (HOO) 2012 shared task, which focuses on correcting preposition and determiner errors made by non-native English speakers. The task consisted of three metrics: Detection, Recognition, and Correction, and measured performance before and after additional revisions to the test data were made. Out of 14 teams that participated, our system scored first in Detection and Recognition and second in Correction before the revisions; and first in Detection and second in the other metrics after revisions. We describe our underlying approach, which relates to our previous work in this area, and propose an improvement to the earlier method, *error inflation*, which results in significant gains in performance.

1 Introduction

The task of correcting grammar and usage mistakes made by English as a Second Language (ESL) writers is difficult: 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.

Below we show examples of two common ESL mistakes considered in this paper:

1. "Nowadays ∅*/*the* Internet makes us closer and closer."
2. "I can see *at*/*on* the list a lot of interesting sports."

In (1), the definite article is incorrectly omitted. In (2), the writer uses an incorrect preposition.

This paper describes the University of Illinois system that participated in the HOO 2012 shared task on error detection and correction in the use of prepositions and determiners (Dale et al., 2012). Fourteen teams took part in the the competition. The scoring included three metrics: Detection, Recognition, and Correction, and our team scored first or second in each metric (see Dale et al. (2012) for details).

The UI system consists of two components, a determiner classifier and a preposition classifier, with a common pre-processing step that corrects spelling mistakes. The determiner system builds on the ideas 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).

Both the determiner and the preposition systems apply the method proposed in our earlier work, which uses the error distribution of the learner data to generate artificial errors in training data. The original method was proposed for adding artificial errors when training on native English data. In this task, however, we apply this method when training on annotated ESL data. Furthermore, we introduce an improvement that is conceptually simple but very effective and which also proved to be successful in an earlier error correction shared task (Dale and Kilgarriff, 2011; Rozovskaya et al., 2011). We identify the unique characteristics of the error correction task and analyze the limitations of existing approaches to error correction that are due to these characteristics. Based on this analysis, we propose the *error inflation* method (Sect. 6.2).

In this paper, we first briefly discuss the task (Sec-

The 7th Workshop on the Innovative Use of NLP for Building Educational Applications, pages 272–280,
Montréal, Canada, June 3-8, 2012.

tion 2) and present our overall approach (Section 3. Next, we describe the spelling correction module (Section 4). Section 5 provides an overview of the training approaches for error correction tasks. We present the inflation method in Section 6. Next, we describe the determiner error correction system (Section 7), and the preposition error correction module (Section 8). In Section 9, we present the performance results of our system in the competition. We conclude with a brief discussion (Section 10).

2 Task Description

The HOO 2012 shared task focuses on correcting determiner and preposition errors made by non-native speakers of English. These errors are some of the most common and also some of the most difficult for ESL learners (Leacock et al., 2010); even very advanced learners make these mistakes (Rozovskaya and Roth, 2010a).

The training data released by the task organizers comes from the publicly available FCE corpus (Yannakoudakis et al., 2011). The original FCE data set contains 1244 essays written by non-native English speakers and is corrected and error-tagged using a detailed error classification schema. The HOO training data contains 1000 of those files.[1] The test data for the task consists of an additional set of 100 student essays, different from the 1244 above.

Since the HOO task focuses on determiner and preposition mistakes, only annotations marking preposition and determiner mistakes were kept. Note that while the other error annotations were removed, the errors still remain in the HOO data. More details can be found in Dale et al. (2012).

3 System Overview

Our system consists of two components that address individually article[2] and preposition errors and use the same pre-processing.

[1] In addition, the participating teams were allowed to use for training the remaining 244 files of this corpus, as well as any other data. We also use a publicly available data set of native English, Google Web 1T corpus (Brants and Franz, 2006), in one of our models.

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

The first pre-processing step is correcting spelling errors. Since the essays were written by students of English as a Second language, and these essays were composed on-the-fly, they contain a large number of spelling errors. These errors add noise to the context around the target word (article or preposition). Good context is crucial for robust detection and correction of article and preposition mistakes.

After spelling errors are corrected, we run a sentence splitter, part-of-speech tagger[3] and shallow parser[4] (Punyakanok and Roth, 2001) on the data. Both the article and the preposition systems use features based on the output of these tools.

We made a 244-document subset of the FCE data a held-out set for development. The results in Sections 7 and 8 give performance on this held-out set, where we use the HOO data (1000 files) for training. The actual performance in the task (Section 9) reflects the system trained on the whole set of 1244 documents.

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). All article systems are trained using the Averaged Perceptron (AP) algorithm (Freund and Schapire, 1999), implemented within Learning Based Java (Rizzolo and Roth, 2010). Our preposition systems combine the AP algorithm with the Naïve Bayes (NB) classifier with prior parameters adapted to the learner data (see Section 5). The AP systems are trained using the *inflation* method (see Section 6.2).

We submitted 10 runs. All of our runs achieved comparable performance. Sections 7 and 8 describe our modules.

4 Correcting Spelling Errors

Analysis of the HOO data made clear the need for a variety of corrections beyond the immediate scope of the current evaluation. When a mistake occurs in the vicinity of a target (i.e. preposition or article) error, it may result in local cues that obscure the nature of the desired correction.

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

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

The following example illustrates such a problem: "*In my opinion your parents should be arrive in the first party of the month becouse we could be go in meeting with famous writer, travelled and journalist who wrote book about Ethiopia.*"

In this sample sentence, there are multiple errors in close proximity: the misspelled word *becouse*; the verb form *should be arrive*; the use of the word *party* instead of *part*; the verb *travelled* instead of a noun form; an incorrect preposition *in* (*in meeting*).

The context thus contains a considerable amount of noise that is likely to negatively affect system performance. To address some of these errors, we run a standard spell-checker over the data.

We use Jazzy[5], an open-source Java spell-checker. The distribution, however, comes only with a US English dictionary, which also has gaps in its coverage of the language. The FCE corpus prefers UK English spelling, so we use a mapping from US to UK English[6] to automatically correct the original dictionary. We also keep the converted US spelling, since our preposition module makes use of native English data, where the US spelling is prevalent.

The Jazzy API allows the client to query a word, and get a list of candidate corrections sorted in order of edit distance from the original term. We take the first suggestion and replace the original word. The resulting substitution may be incorrect, which may in turn mislead the downstream correction components. However, manual evaluation of the spelling corrections suggested about 80% were appropriate, and experimental evaluation on the corpus development set indicated a modest overall improvement when the spell-checked documents were used in place of the originals.

5 Training for Correction Tasks

The standard approach to correcting context-sensitive ESL mistakes follows the methodology of the *context-sensitive spelling correction* task that addresses such misspellings as *their* and *there* (Carlson et al., 2001; Golding and Roth, 1999; Golding and Roth, 1996; Carlson and Fette, 2007; Banko and Brill, 2001).

Following Rozovskaya and Roth (2010c), we distinguish between two training paradigms in ESL error correction, depending on whether the author's original word choice is used in training as a feature. In the standard *context-sensitive spelling correction* paradigm, the decision of the classifier depends only on the context around the author's word, e.g. article or preposition, and the author's word itself is not taken into consideration in training.

Mistakes made by non-native speakers obey certain regularities (Lee and Seneff, 2008; Rozovskaya and Roth, 2010a). Adding knowledge about *typical* errors to a model significantly improves its performance (Gamon, 2010; Rozovskaya and Roth, 2010c; Dahlmeier and Ng, 2011). *Typical* errors may refer both to speakers whose first language is L_1 and to specific authors. For example, non-native speakers whose first language does not have articles tend to make more articles errors in English (Rozovskaya and Roth, 2010a).

Since non-native speakers' mistakes are systematic, the author's word choice (the *source* word) carries a lot of information. Models that use the *source* word in training (Han et al., 2010; Gamon, 2010; Dahlmeier and Ng, 2011) learn which errors are typical for the learner and thus significantly outperform systems that only look at context. We call these models *adapted*. Training adapted models requires annotated data, since in native English data the source word is always correct and thus cannot be used by the classifier.

In this work, we use two methods of adapting a model to typical errors that have been proposed earlier. Both methods were originally developed for models trained on native English data: they use a small amount of annotated ESL data to generate error statistics. The **artificial errors** method is based on generating artificial errors[7] in correct native English training data. The method was implemented within the Averaged Perceptron (AP) algorithm (Rozovskaya and Roth, 2010c; Rozovskaya and Roth, 2010b), a discriminative learning algorithm, and this is the algorithm that we use in this work. The **NB-priors** method is a special adaptation technique for the Naïve Bayes algorithm (Rozovskaya and Roth, 2011). While **NB-priors** improves both precision

[5] http://jazzy.sourceforge.net/

[6] http://www.tysto.com/articles05/q1/20050324uk-us.shtml

[7] For each task, only relevant errors are generated – for example, article mistakes for the article correction task.

and recall, the **artificial errors** approach suffers from low recall due to error sparsity (Sec. 6.1).

In this work, in the preposition correction task, we use the **NB-priors** method without modifications (as described in the original paper). We use the **artificial errors** approach both for article and preposition error correction but with two important modifications: we train on annotated ESL data instead of native data, and use the proposed *error inflation* method (described in Section 6) to increase the error rate in training.

6 Error Inflation

In this section, we show why AP (Freund and Schapire, 1999), a discriminative classifier, is sensitive to the error sparsity of the data, and propose a method that addresses the problems raised by this sensitivity.

6.1 Error Sparsity and Low Recall

The low recall of the AP algorithm is related to the nature of the error correction tasks, which exhibit low error rates. Even for ESL writers, over 90% of their preposition and article usage is correct, which makes the errors very sparse (Rozovskaya and Roth, 2010c). The low recall problem is, in fact, a special case of a more general problem where there is one or a small group of dominant features that are very strongly correlated with the label. In this case, the system tends to predict the label that matches this feature, and tends to not predict it when that feature is absent. In error correction, which tends to have a very skewed label distribution, this results in very few errors being detected by the system: when training on annotated data with naturally occurring errors and using the source word as a feature, the system will learn that in the majority of cases the source word corresponds to the label, and will tend to over-predict it, which will result in low recall.

In the **artificial errors** approach, errors are simulated according to real observed mistakes. Table 1 shows a sample confusion matrix based on preposition mistakes in the FCE corpus; we show four rows, but the entire table contains 17 rows and columns, one for each preposition, and each entry shows $Prob(p_i|p_j)$, the probability that the author's preposition is p_i given that the correct preposition is p_j. The matrix also shows the preposition count for each source and label in the data set. Given the entire matrix and the counts, it is also possible to generate the matrix in the other direction and obtain $Prob(p_j|p_i)$, the probability that the correct preposition is p_j given that the author's preposition is p_i. This other matrix is used for adapting NB with the priors method.

The confusion matrix is sparse and shows that the distribution of alternatives for each source preposition is very different from that of the others. This strongly suggests that these errors are systematic. Additionally, most prepositions are used correctly, so the error rate is very low (the error rate can be estimated by looking at the matrix diagonal in the table; for example, the error rate for the preposition *about* is lower than for *into*, since 94.4% of the occurrences of label *about* are correct, but only 76.8% of label *into* are correct).

The artificial errors thus model the two properties that we mentioned: the confusability of different preposition pairs and the low error rate, and the artificial errors are similarly sparse.

6.2 The Error Inflation Method

Two extreme choices for solving the low recall problem due to error sparsity are: (1) training without the *source word* feature or (2) training with this feature, where the classifier relies on it too much. Models trained without the *source* feature have very poor precision. While the **NB-priors** method does have good recall, our expectation is that with the right approach, a discriminative classifier will also improve recall, but maintain higher precision as well.

We wish to reduce the confidence that the system has in the source word, while preserving the knowledge the model has about likely confusions and contexts of confused words. To accomplish this, we reduce the proportion of correct examples, i.e. examples where the *source* and the *label* are the same, by some positive constant < 1.0 and distribute the extra probability mass among the typical errors in an appropriate proportion by generating additional error examples. This inflates the proportion of artificial errors in the training data, and hence the error rate, while keeping the probability distribution among likely corrections the same. Increasing the error rate improves the recall, while the typical er-

Label	Sources												
	on (648)	about (700)	into (54)	with (733)	as (410)	at (880)	by (243)	for (1394)	from (515)	in (2213)	of (1954)	over (98)	to (1418)
on (598)	**0.846**	0.003	0.003	0.008	0.013	-	0.003	0.022	-	0.076	0.013	0.001	0.009
about (686)	0.004	**0.944**	-	0.007	-	-	-	0.022	0.005	0.002	0.016	0.001	-
into (55)	0.001	-	**0.768**	-	-	-	0.011	0.011	-	0.147	-	-	0.053
with (710)	0.001	0.006	-	**0.934**	-	0.001	0.007	0.004	0.001	0.027	0.003	-	0.015

Table 1: **Confusion matrix for preposition errors**. Based on data from the FCE corpus for top 17 most frequent English prepositions. The left column shows the correct preposition. Each row shows the author's preposition choices for that label and $Prob(source|label)$. The sources *among*, *between*, *under* and *within* are not shown for lack of space; they all have 0 probabilities in the matrix. The numbers next to the targets show the count of the label (or source) in the data set.

ror knowledge ensures that high precision is maintained. This method causes the classifier to rely on the *source* feature less and increases the contribution of the features based on context. The learning algorithm therefore finds a more optimal balance between the *source* feature and the context features.

Algorithm 1 shows the pseudo-code for generating training data; it takes as input training examples, the confusion matrix CM as shown in Table 1, and the inflation constant, and generates artificial source features for correct training examples.[8] An inflation constant value of 1.0 simulates learner mistakes without inflation. Table 2 shows the proportion of artificial errors created in training using the inflation method for different inflation rates.

Algorithm 1 Data Generation with Inflation

Input: Training examples E with correct sources, confusion matrix CM, inflation constant C
Output: Training examples E with artificial errors
for Example e in E **do**
  Initialize *lab* $\leftarrow$ *e.label*, *e.source* $\leftarrow$ *e.label*
  Randomize *targets* $\in CM[lab]$
  Initialize *flag* $\leftarrow$ *False*
  for target *t* in *targets* **do**
    if flag equals True **then**
      Break
    end if
    if t equals lab **then**
      $Prob(t) = CM[lab][t] \cdot C$
    else
      $Prob(t) = \frac{1.0 - CM[lab][lab] \cdot C}{1.0 - CM[lab][lab]} \cdot CM[lab][t]$
    end if
    $x \leftarrow$ *Random*$[0, 1]$
    if $x < Prob(t)$ **then**
      e.source $\leftarrow t$
      flag $\leftarrow$ *True*
    end if
  end for
end for
return E

[8] When training on native English data, all examples are correct. When training on annotated learner data, some examples will contain naturally occurring mistakes.

Inflation rate					
1.0 (Regular)	0.9	0.8	0.7	0.6	0.5
7.7%	15.1%	22.6%	30.1%	37.5%	45.0%

Table 2: **Artificial errors**. Proportion of generated artificial preposition errors in training using the inflation method (based on the FCE corpus).

7 Determiners

Table 4 shows the distribution of determiner errors in the HOO training set. Even though the majority of determiner errors involve article mistakes, 14% of errors are personal and possessive pronouns.[9] Most of the determiner errors involve omitting an article. Similar error patterns have been observed in other ESL corpora (Rozovskaya and Roth, 2010a).

Our system focuses on article errors. Because the majority of determiner errors are omissions, it is very important to target this subset of mistakes. One approach would be to consider every space as a possible article insertion point. However, this method will likely produce a lot of noise. The standard approach is to consider noun-phrase-initial contexts (Han et al., 2006; Rozovskaya and Roth, 2010c).

Error type	Example
Repl. 15.7%	"Can you send me *the*/a* letter back writing what happened to you recently."
Omis. 57.5%	"Nowadays ∅*/*the* Internet makes us closer and closer."
Unnec. 26.8%	"One of my hobbies is *the*/*∅ photography."

Table 4: **Distribution of determiner errors in the HOO training data.**

[9] 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*$\&f_i \; \forall i \in NP_1$

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.

7.1 Determiner Features

The features are presented in Table 3. The model also uses the *source* article as a feature.

7.2 Training the Determiner System

Model	Detection	Correction
AP (natural errors)	30.75	28.97
AP (inflation)	**34.62**	**32.02**

Table 5: **Article development results: AP with inflation.** The performance shows the F-Score for the 244 held-out documents of the original FCE data set. AP with inflation uses the constant value of 0.8.

The article classifier is based on the artificial errors approach (Rozovskaya and Roth, 2010c). The original method trains a system on native English data. The current setting is different, since the FCE corpus contains annotated learner errors. Since the errors are sparse, we use the *error inflation* method (Section 6.2) to boost the proportion of errors in training using the error distribution obtained from the same training set. The effectiveness of this method is demonstrated by the system performance: we obtain the top or second result in every metric. Note also that the article system does not use additional data for training.

Table 5 compares the performance of the system trained on natural errors with the performance of the system trained with the inflation method. We found that any value of the inflation constant between 0.9 and 0.5 will give a boost in performance. We use several values; the top determiner model uses the inflation constant of 0.8.

8 Prepositions

Table 6 shows the distribution of the three types of preposition errors in the HOO training data. The FCE annotation distinguishes between preposition mistakes and errors involving the infinitive marker *to*, e.g. "*He wants ∅*/to go there.*", which are annotated as verb errors. Since in the competition only article and preposition annotations are kept, these errors are not annotated, and thus we do not target these mistakes.

Error type	Example
Repl. 57.9%	"I can see *at*/on* the list a lot of interesting sports."
Omis. 24.0%	"I will be waiting *∅*/for* your call."
Unnec. 18.1%	"Despite *of*/∅* being tiring , it was rewarding"

Table 6: **Distribution of preposition errors in the HOO training data.**

To detect missing preposition errors, we use a set of rules, mined from the training data, to identify possible locations where a preposition might have been incorrectly omitted. Below we show examples of such contexts.

- "I will be waiting *∅*/for* your call."
- "But now we use planes to go *∅*/to* far places."

8.1 Preposition Features

All features used in the preposition module are lexical: word n-grams in the 4-word window around

Feature Type	Description
Word n-ngram features in the 4-word window around the target	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, $w_2BwBwAw_2A$, $wBwAw_2Aw_3A$, $wAw_2Aw_3w_4A$
Preposition complement features	*compHead*, *wB&compHead*, w_2BwB*&compHead*

Table 7: **Features used in the preposition error correction system.** wB and wA denote the word immediately before and after the target, respectively; the other features are defined similarly. *compHead* denotes the head of the preposition complement. *wB&compHead*, w_2*BwB&compHead* are feature conjunctions of *compHead* with wB and w_2BwB, respectively.

the target preposition, and three features that use the head of the preposition complement (see Table 7). 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. AP is trained on the HOO data and uses n-grams of lengths 2, 3, and 4, the head complement features, and the author's preposition as a feature.

Model	Detection	Correction
AP (inflation)	34.64	27.51
NB-priors	38.76	26.57
Combined	**41.27**	**29.35**

Table 8: **Preposition development results: performance of individual and combined systems.** The performance shows the F-Score for the 244 held-out documents of the original FCE data set.

8.2 Training the Preposition System

We train two systems. The first one is an AP model trained on the FCE data with *inflation* (similar to the article system). Correcting preposition errors requires more data to achieve performance comparable to article error correction, due to the task complexity (Gamon, 2010). Moreover, given that the development and test data are quite different,[10] it makes sense to use a model that is independent of those, to avoid overfitting. We combine the AP model with a model trained on native English data. Our second system is an NB-priors classifier trained on the the Google Web 1T 5-gram corpus (Brants and Franz, 2006). We use training data to replace the prior parameters of the model (see Rozovskaya and Roth, 2011 for more detail). The NB-priors model does not target preposition omissions.

[10]The data contains essays written on prompts, so that the training data may contain several essays written on the same prompt and thus will be very similar in content. In contrast, we expected that the test data will likely contain essays on a different set of prompts.

The NB-priors model outperforms the AP classifier. The two models are also very different due to the different learning algorithms and the type of the data used in training. Our final preposition model is thus a combination of these two, where we take as the base the decisions of the NB-priors classifier and add the AP model predictions for cases when the base model does not flag a mistake. Table 8 shows the results. The combined model improves both the detection and correction scores. Our preposition system ranked first in detection and recognition and second in correction.

Model	Detection	Correction
AP (natural errors)	13.50	12.73
AP (inflation)	**21.31**	**32.02**

Table 9: **Preposition development results: AP with inflation.** The performance shows the F-Score for the 244 held-out documents of the original FCE data set. AP with inflation uses the constant value of 0.7.

9 Test Performance

A number of revisions were made to the test data based on the input from the participating teams after the initial results were obtained, where each team submitted proposed edits to correct annotation mistakes. We show both results.

Table 10 shows results before the revisions were made. Row 1 shows the performance of the determiner system for the three metrics. This system achieved the best score in correction, and the second best scores in detection and recognition. The system is described in Section 7.2, with the exception that the final system for the article correction is trained on the entire FCE data set.

Table 10 (row 2) presents the results on preposition error correction. The system is described in Section 8.2 and is a combined model of AP trained with inflation on the FCE data set and NB-priors model trained on the Google Web 1T corpus. The

Model	Detection			Recognition			Correction		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
Articles	40.00	37.79	38.86^{2}	38.05	35.94	36.97^{2}	35.61	33.64	34.60^{1}
Prepositions	38.21	45.34	41.47^{1}	31.05	40.25	35.06^{1}	20.36	24.15	22.09^{2}
Combined	37.22	43.71	40.20^{1}	34.23	36.64	35.39^{1}	26.39	28.26	27.29^{2}

Table 10: **Performance on test before revisions.** Results are shown before revisions were made to the data. The rank of the system is shown as a superscript.

Model	Detection			Recognition			Correction		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
Articles	43.90	39.30	41.47^{2}	45.98	34.93	39.70^{2}	41.46	37.12	39.17^{2}
Prepositions	41.43	47.54	44.27^{1}	37.14	42.62	39.69^{1}	26.79	30.74	28.63^{2}
Combined	43.56	42.92	43.24^{1}	38.97	39.96	39.46^{2}	32.58	33.40	32.99^{2}

Table 11: **Performance on test after revisions.** Results are shown after revisions were made to the data. The rank of the system is shown as a superscript.

preposition system achieved the best scores in detection and recognition, scoring second in correction.

Row 3 shows the performance of the combined system. This system was ranked first in detection and recognition, and second in correction.

Table 11 shows our performance after the revisions were applied.

10 Discussion

The HOO 2012 shared task follows the HOO 2011 pilot shared task (Dale and Kilgarriff, 2011), where the data was fully corrected and error-tagged and the participants could address any types of mistakes. The current task allows for comparison of individual systems for each error type considered. This is important, since to date it has been difficult to compare different systems due to the lack of a benchmark data set.

The data used for the shared task has many errors besides the preposition and determiner errors; the annotations for these have been removed. One undesirable consequence of this approach is that some complex errors that involve either an article or a preposition mistake but depend on other corrections on neighboring words, e.g. a noun of a verb, may result in ungrammatical sequences.

Clearly, the task of annotating all requisite corrections is a daunting task, and it is preferable to identify subsets of these corrections that can be tackled somewhat independently of the rest, and these more complex cases present a problem.

To address these conflicting needs, we propose that the scope of all "final" corrections be marked, without necessarily specifying all individual corrections necessary to transform the original text into correct English. Edits that plausibly require corrections to their context to resolve correctly could then be treated as *out of scope*, and ignored by spelling correction systems even though in other contexts, those same edits would be *in* scope.

11 Conclusion

We have demonstrated how a competitive system for preposition and determiner error correction can be built using techniques that address the error sparsity of the data and the overfitting problem. We built on our previous work and presented the error inflation method that can be applied to the earlier proposed artificial errors approach to boost recall. Our determiner system used error inflation and trained a model using only the annotated FCE corpus. Our preposition system combined the FCE-trained system with a native-data model that was adapted to learner errors, using the NB-priors approach proposed earlier. Both of the systems showed competitive performance, scoring first or second in every task ranking.

Acknowledgments

The authors thank Jeff Pasternack for his assistance and Vivek Srikumar for helpful feedback. This research is supported by a grant from the U.S. Department of Education and is partly supported by the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-018.

References

M. Banko and E. Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proc.*

of 39th Annual Meeting of the Association for Computational Linguistics (ACL), pages 26–33, Toulouse, France, July.

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

A. Carlson and I. Fette. 2007. Memory-based context-sensitive spelling correction at web scale. In *Proc. of the IEEE International Conference on Machine Learning and Applications (ICMLA)*.

A. Carlson, J. Rosen, and D. Roth. 2001. Scaling up context sensitive text correction. In *Proceedings of the National Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 45–50.

D. Dahlmeier and H. T. Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 915–923, Portland, Oregon, USA, June. Association for Computational Linguistics.

R. Dale and A. Kilgarriff. 2011. Helping Our Own: The HOO 2011 pilot shared task. In *Proc. of the 13th European Workshop on Natural Language Generation (ENLG)*, pages 242–249, Nancy, France.

R. Dale, I. Anisimoff, and G. Narroway. 2012. A report on the preposition and determiner error correction shared task. In *Proc. of the NAACL HLT 2012 Seventh WorkshopWorkshop on Innovative Use of NLP for Building Educational Applications*, Montreal, Canada, June. Association for Computational Linguistics.

Y. Freund and R. E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

M. Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *Proc. of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 163–171, Los Angeles, California, June.

A. R. Golding and D. Roth. 1996. Applying Winnow to context-sensitive spelling correction. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 182–190.

A. R. Golding and D. Roth. 1999. A Winnow based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

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.

N. Han, J. Tetreault, S. Lee, and J. Ha. 2010. Using an error-annotated learner corpus to develop and ESL/EFL error correction system. In *Proc. of the Seventh conference on International Language Resources and Evaluation (LREC)*, Valletta, Malta, May. European Language Resources Association (ELRA).

C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers.

J. Lee and S. Seneff. 2008. An analysis of grammatical errors in non-native speech in English. In *Proc. of the 2008 Spoken Language Technology Workshop*, Goa.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 995–1001. MIT Press.

N. Rizzolo and D. Roth. 2010. Learning based java for rapid development of nlp systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, 5.

A. Rozovskaya and D. Roth. 2010a. Annotating ESL errors: Challenges and rewards. In *Proc. of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California, June. Association for Computational Linguistics.

A. Rozovskaya and D. Roth. 2010b. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 961–970, Cambridge, MA, October. Association for Computational Linguistics.

A. Rozovskaya and D. Roth. 2010c. Training paradigms for correcting errors in grammar and usage. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 154–162, Los Angeles, California, June. Association for Computational Linguistics.

A. Rozovskaya and D. Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pages 924–933, Portland, Oregon, USA, June. Association for Computational Linguistics.

A. Rozovskaya, M. Sammons, J. Gioja, and D. Roth. 2011. University of Illinois system in HOO text correction shared task. In *Proc. of the 13th European Workshop on Natural Language Generation (ENLG)*.

H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 180–189, Portland, Oregon, USA, June. Association for Computational Linguistics.