HOO 2012 Error Recognition and Correction Shared Task: Cambridge University Submission Report

Ekaterina Kochmar
Computer Laboratory
University of Cambridge
ek358@cl.cam.ac.uk

Øistein Andersen
iLexIR Ltd
Cambridge
and@ilexir.co.uk

Ted Briscoe
Computer Laboratory
University of Cambridge
ejb@cl.cam.ac.uk

Abstract

Previous work on automated error recognition and correction of texts written by learners of English as a Second Language has demonstrated experimentally that training classifiers on error-annotated ESL text generally outperforms training on native text alone and that adaptation of error correction models to the native language (L1) of the writer improves performance. Nevertheless, most extant models have poor precision, particularly when attempting error correction, and this limits their usefulness in practical applications requiring feedback.

We experiment with various feature types, varying quantities of error-corrected data, and generic versus L1-specific adaptation to typical errors using Naïve Bayes (NB) classifiers and develop one model which maximizes precision. We report and discuss the results for 8 models, 5 trained on the HOO data and 3 (partly) on the full error-coded Cambridge Learner Corpus, from which the HOO data is drawn.

1 Introduction

The task of detecting and correcting writing errors made by learners of English as a Second Language (ESL) has recently become a focus of research.

The majority of previous papers in this area have presented machine learning methods with models being trained on well-formed native English text (Eeg-Olofsson and Knutsson, 2003; De Felice and Pulman, 2008; Gamon et al., 2008; Han et al., 2006; Izumi et al., 2003; Tetreault and Chodorow, 2008; Tetreault et al., 2010). However, some recent approaches have explored ways of using annotated non-native text either by incorporating error-tagged data into the training process (Gamon, 2010; Han et al., 2010), or by using native language-specific error statistics (Rozovskaya and Roth, 2010b; Rozovskaya and Roth, 2010c; Rozovskaya and Roth, 2011). Both approaches show improvements over the models trained solely on well-formed native text.

Training a model on error-tagged non-native text is expensive, as it requires large amounts of manually-annotated data, not currently publically available. In contrast, using native language-specific error statistics to adapt a model to a writer's first or native language (L1) is less restricted by the amount of training data.

Rozovskaya and Roth (2010b; 2010c) show that adapting error corrections to the writer's L1 and incorporating artificial errors, in a way that mimics the typical error rates and confusion patterns of non-native text, improves both precision and recall compared to classifiers trained on native data only. The approach proposed in Rozovskaya and Roth (2011) uses L1-specific error correction patterns as a distribution on priors over the corrections, incorporating the appropriate priors into a generic Naïve Bayes (NB) model. This approach is both cheaper to implement, since it does not require a separate classifier to be trained for every L1, and more effective, since the priors condition on the writer's L1 as well as on the possible confusion sets.

Some extant approaches have achieved good results on error detection. However, error correction is much harder and on this task precision remains low. This is a disadvantage for applications such as self-tutoring or writing assistance, which require feedback to the user. A high proportion of errorful suggestions is likely to further confuse learners and/or non-native writers rather than improve their writing or assist learning. Instead a system which maximizes precision over recall returning accurate suggestions for a small proportion of errors is likely to be more helpful (Nagata and Nakatani, 2010).

In section 2 we describe the data used for training and testing the systems we developed. In section 3 we describe the preprocessing of the ESL text undertaken to provide a source of features for the classifiers. We also discuss the feature types that we exploit in our classifiers. In section 4 we describe and report results for a high precision system which makes no attempt to generalize from training data. In section 5 we describe our approach to adapting multiclass NB classifiers to characteristic errors and L1s. We also report the performance of some of these NB classifiers on the training and test data. In section 6 we report the official results of all our submitted runs on the test data and also on the HOO training data, cross-validated where appropriate. Finally, we briefly discuss our main results, further work, and lessons learnt.

2 Cambridge Learner Corpus

The Cambridge Learner Corpus[1] (CLC) is a large corpus of learner English. It has been developed by Cambridge University Press in collaboration with Cambridge Assessment, and contains examination scripts written by learners of English from 86 L1 backgrounds. The scripts have been produced by language learners taking Cambridge Assessment's ESL examinations.[2]

The linguistic errors committed by the learners have been manually annotated using a taxonomy of 86 error types (Nicholls, 2003). Each error has been manually identified and tagged with an appropriate code, specifying the error type, and a suggested correction. Additionally, the scripts are linked to metadata about examination and learner. This includes the year of examination, the question prompts, the learner's L1, as well as the grades obtained. The current version of the CLC contains about 20M words of error-annotated scripts from a wide variety of examinations.

The HOO training and test datasets are drawn from the CLC. The training dataset is a reformatted 1000-script subset of a publically-available subset of CLC scripts produced by learners sitting the First Certficate in English (FCE) examination.[3] This examination assesses English at an upper-intermediate level, so many learners sitting this exam still manifest a number of errors motivated by the conventions of their L1s. The CLC-FCE subcorpus was extracted, anonymized, and made available as a set of XML files by Yannakoudakis et al. (2011).[4]

The HOO training dataset contains scripts from FCE examinations undertaken in the years 2000 and 2001 written by speakers of 16 L1s. These scripts can be divided into two broad L1 typological groups, Asian (Chinese, Thai, Korean, Japanese) and European (French, Spanish, Italian, Portuguese, Catalan, Greek, Russian, Polish). The latter can be further subdivided into Slavic (Russian, Polish) and Romance. In turn, the Romance languages differ in typological relatedness with, for example, Portuguese and Spanish being closer than Spanish and French. Error coding which is not relevant to preposition or determiner errors has been removed from the training data so that only six error type annotations are retained for training: incorrect, missing or unnecessary determiners (RD, MD, UD) and prepositions (RT, MT, UT).

One consequence of this reformatting is that the contexts of these errors often contain further errors of different types that are no longer coded. The idea is that errors should be considered in their natural habitat, and that correcting and copy-editing the surrounding text would create an artificial task. On the other hand, not correcting anything makes it difficult in some cases and nigh impossible in others to determine whether a given determiner or preposition is correct or not. The error-coding in the CLC in such cases (provided the writer's intent is deemed recoverable) depends not only on the original text, but also on the correction of nearby errors.

[1] http://www.cup.cam.ac.uk/gb/elt/catalogue/subject/custom/item3646603/Cambridge-International-Corpus-Cambridge-Learner-Corpus

[2] http://www.cambridgeesol.org/

[3] http://www.cambridgeesol.org/exams/general-english/fce.html

[4] http://ilexir.co.uk/applications/clc- fce-dataset/

Certain errors even appear as a direct result of correcting others: for instance, the phrase *to sleep in tents* has been corrected to *to sleep in a tent* in the CLC; this ends up as a 'correction' to *to sleep in a tents* in the HOO dataset. This issue is difficult to avoid given that the potential solutions are all labour-intensive (explicit indication of dependencies between error annotations, completely separate error annotation for different types of errors, or manual removal of spurious errors after extraction of the types of error under consideration), and we mention it here mainly to explain the origin of some surprising annotations in the dataset.

A more HOO-specific problem is the '[removal of] elements [from] some of [the] files [...] to dispose of nested edits and other phenomena that caused difficulties in the preprocessing of the data' (Dale et al., 2012). This approach unfortunately leads to mutilated sentences such as *I think if we wear thistoevery wherespace ships. This mean.* replacing the original *I think if we wear this clothes we will travel to every where easier than we use cars, ships, planes and space ships. This mean the engineering will find the way to useless petrol for it, so it must useful in the future.*

The HOO test set consists of 100 responses to individual prompts from FCE examinations set between 1993 and 2009, also drawn from the CLC. As a side effect of removing the test data from the full CLC, we have found out that the distribution of L1s, examination years and exam prompts is different from the training data. There are 27 L1s exemplified, a superset of the 16 seen in the HOO training data; about half are Romance, and the rest are widely distributed with Asian and Slavic languages less well represented than in the training data.

In the experiments reported below, we make use of both the HOO training data and the full 20M words of error-annotated CLC, but with the HOO test data removed, to train our systems. Whenever we use the larger training set we refer to this as the *full CLC* below.

3 Data Preprocessing

We parsed the training and test data (see Section 2) using the Robust Accurate Statistical Parsing (RASP) system with the standard tokenization and sentence boundary detection modules and the unlexicalized version of the parser (Briscoe et al., 2006) in order to broaden the space of candidate features types. The features used in our experiments are mainly motivated by the fact that lexical and grammatical features have been shown in previous work to be effective for error detection and correction. We believe RASP is an appropriate tool to use with ESL text because the PoS tagger deploys a well-developed unknown word handling mechanism, which makes it relatively robust to noisy input such as misspellings, and because the parser deploys a hand-coded grammar which indicates ungrammaticality of sentences and markedness of constructions and is encoded entirely in terms of PoS tag sequences. We utilize the open-source version of RASP embedded in an XML-handling pipeline that allows XML-encoded metadata in the CLC and HOO training data to be preserved in the output, but ensures that unannotated text is passed to RASP (Andersen et al., 2008).

My friend was (MD: *a*) *good student*

Grammatical Relations (GRs):

```
 (ncsubj be+ed:3_VBDZ friend:2_NN1 _)
 (xcomp _ be+ed:3_VBDZ student:6_NN1)
 (ncmod _ student:6_NN1 good:5_JJ)
 (det friend:2_NN1 My:1_APP$)
*(det student:6_NN1 a:4_AT1)
```

Figure 1: RASP GR output

Relevant output of the system is shown in Figure 1 for a typical errorful example. The grammatical relations (GRs) form a connected, directed graph of typed bilexical head-dependent relations (where a non-fragmentary analysis is found). Nodes are lemmatized word tokens with associated PoS tag and sentence position number. Directed arcs are labelled with GR types. In the factored representation shown here, each line represents a GR type, the head node, the dependent node, and optional subtype information either after the GR type or after the dependent. In this example, the asterisked GR would be missing in the errorful version of the sentence. We extract the most likely analysis for each sentence based on the most probable tag sequence found by the tagger.

Extraction of the lexical and grammatical information from the parser output is easier when a determiner or preposition is present than when it is missing. During training, for all nouns, we checked for a `det` relation to a determiner, and whenever no `det` GR is present, we checked whether the noun is preceded by an MD annotation in the XML file. For missing prepositions, we have only extracted cases where a noun is governed by a verb with a `dobj` relation, and cases where a noun is governed by another noun with an `ncmod` (non-clausal modifier) relation. For example, in *It's been a long time since I last wrote you*, in absence of the preposition *to* the parser would 'recognize' a `dobj` relation between *you* and *wrote*, and this case would be used as a training example for a missing preposition, while *I trusted him* with the same `dobj` relation between *trusted* and *him* would be used as a training example to correct unwanted use of a preposition as in *I trusted *to him*.

3.1 Feature Types

In all the experiments and system configurations described below, we used a similar set of features based on the following feature templates.

For determiner errors:

- `Noun lemma`: lemma of the noun that governs the determiner
- `Noun PoS`: PoS tag of the noun
- `Distance from Noun`: distance in number of words to the governed determiner
- `Head lemma`: head lemma in the shortest grammatical relation in which the noun is dependent
- `Head PoS`: as defined above, but with PoS tag rather than lemma
- `Distance from Head`: distance in number of words to the determiner from head, as defined above (for `Head lemma`)
- `GR type to Noun`: a GR between `Head` and `Noun`.

For instance for the example shown in Figure 1, the noun lemma is *student*, the noun PoS is NN1, the distance from the noun is 2, the head lemma is *be*, the head PoS is VBDZ, and the distance from the head is 1, while the GR type to the noun is `xcomp`.

For preposition errors:

- `Preposition (P)`: target preposition
- `Head lemma (H)`: head lemma of the GR in which the preposition is dependent
- `Dependent lemma (D)`: dependent lemma of the GR in which the preposition is head.

For instance, in *I am looking forward to your reply*, `P` is *to*, `H` is *look* and `D` is *reply*.

In contrast to work by Rozovskaya and Roth, amongst others, we have not used word context features, but instead focused on grammatical context information for detecting and correcting errors. We also experimented with some other feature types, such as n-grams consisting of the head, preposition and dependent lemmas, but these did not improve performance on the cross-validated HOO training data, perhaps because they are sparser and the training set is small. However, there are many other potential feature types, such as PoS n-grams or syntactic rule types, and so forth that we don't explore here, despite their probable utility. Our main focus in these experiments is not on optimal feature engineering but rather on the issues of classifier adaption to errors and high precision error correction.

4 A Simple High Precision Correction System

We have experimented with a number of approaches to maximizing precision and have not outperformed a simple model that doesn't generalize from the training data using machine learning techniques. We leverage the large amount of error-corrected text in the full CLC to learn reliable contexts in which errors occur and their associated corrections. For the HOO shared task, we tested variants of this approach for missing determiner (MD) and incorrect preposition (RT) errors. Better performing features and thresholds used to define contexts were found by testing variants on the HOO training data. The feature types from section 3.1 deployed for the MD system submitted for the official run were `Noun` `lemma`, `Noun PoS`, `GR types to Noun` and `GR types from Noun` (set of GRs which has the noun as head). For the RT system, all three `P`, `H`, and `D` features were used to define contexts. A context is considered reliable if it occurs at least twice in the full CLC and more than 75% of the time it occurs with an error.

The performance of this system on the training data was very similar to performance on the test data (in contrast to our other runs). We also explored L1-specific and L1-group variants of these systems; for instance, we split the CLC data into Asian and European languages, trained separate systems on each, and then applied them according to the L1 metadata supplied with the HOO training data. However, all these systems performed worse than the best unadapted system.

The results for the generic, unadapted MD and RT systems appear as run 0 in Tables 4–9 below. These figures are artefactually low as we don't attempt to detect or correct UD, UT, RD or MT errors. The actual results computed from the official runs solely for MD errors are for detection, recognition and correction: 83.33 precision and 7.63 recall, which gives an F-measure of 13.99; the RT system performed at 66.67 precision, 8.05 recall and 14.37 F-measure on the detection, recognition and correction tasks. Despite the low recall, this was our best submitted system in terms of official correction F-score.

5 Naïve Bayes (NB) (Un)Adapted Multiclass Classifiers

Rozovskaya and Roth (2011) demonstrate on a different dataset that Naïve Bayes (NB) can outperform discriminative classifiers on preposition error detection and correction if the prior is adapted to L1-specific estimates of error-correction pairs. They compare the performance of an unadapted NB multiclass classifier, in which the prior for a preposition is defined as the relative probability of seeing a specific preposition compared to a predefined subset of the overall PoS class (which they call the Conf(usion) Set):

$$\text{prior}(p) = \frac{C(p)}{\sum_{q \in \text{ConfSet}} C(q)},$$

to the performance of the same NB classfier with an adapted prior which calculates the probability of a correct preposition as:

$$\text{prior}(c, p, \text{L1}) = \frac{C_{\text{L1}}(p, c)}{C_{\text{L1}}(p)},$$

where $C_{\text{L1}}(p)$ is the number of times preposition p is seen in texts written by learners with L1 as their native language, and $C_{\text{L1}}(p, c)$ is the number of times c is the correct preposition when p is used.

We applied Rozovskaya and Roth's approach to determiners as well as prepositions, and experimented with priors calculated in the same way for L1 groups as well as specific L1s. We also compared L1-adaptation to generic adaption to corrections, calculated as:

$$\text{prior}(c, p) = \frac{C(p, c)}{C(p)},$$

We have limited the set of determiners and prepositions that our classifiers aim to detect and correct, if necessary. Our confusions sets contain:

- `Determiners`: *no determiner*, *the*, *a*, *an*;
- `Prepositions`: *no preposition*, *in*, *of*, *for*, *to*, *at*, *with*, *on*, *about*, *from*, *by*, *after*.

Therefore, for determiners, our systems were only aimed at detecting and correcting errors in the use of articles, and we have not taken into account any errors in the use of possessive pronouns (*my*, *our*, etc.), demonstratives (*this*, *those*, etc.), and other types of determiners (*any*, *some*, etc.). For prepositions, it is well known that a set of about 10 of the most frequent prepositions account for more than 80% of all prepositional usage (Gamon, 2010).

We have calculated the upper bounds for the training and test sets when the determiner and preposition confusion sets are limited this way. The upper bound recall for *recognition* (i.e., ability of the classifier to recognize that there is an error, dependent on the fact that only the chosen determiners and prepositions are considered) is calculated as the proportion of cases where the incorrect, missing or unnecessary determiner or preposition is contained in our confusion set. For the training set, it is estimated at 91.95, and for the test at 93.20. Since for *correction*, the determiner or preposition suggested by the system should also be contained in our confusion set, upper bound recall for *correction* is slightly lower than that for *recognition*, and is estimated at 86.24 for the training set, and at 86.39 for the test set. These figures show that the chosen candidates distribute similarly in both datasets, and that a system aimed at recognition and correction of only these function words can obtain good performance on the full task.

The 1000 training scripts were divided into 5 portions pseudo-randomly to ensure that each portion contained approximately the same number of L1-specific scripts in order not to introduce any L1-related bias. The results on the training set presented below were averaged across 5 runs, where in each run 4 portions (about 800 scripts) were used for training, and one portion (about 200 scripts) was used for testing.

We treated the task as multi-class classification, where the number of classes equates to the size of our confusion set, and when the classifier's decision is different from the input, it is considered to be errorful. For determiners, we used the full set of features described in section 3.1, whereas for prepositions, we have tried two different feature sets: only head lemma (`H`), or `H` with the dependent lemma (`D`).

We ran the unadapted and L1-adapted NB classifiers on determiners and prepositions using the features defined above. The results of these preliminary experiments are presented below.

5.1 Unadapted and L1-adapted NB classifiers

Tables 1 to 3 below present results averaged over the 5 runs for the unadapted classifiers. We report the results in terms of recall, precision and F-score for detection, recognition and correction of errors as defined for the HOO shared task.[5]

We have experimented with two types of L1-specific classification: `classifier1` below is a combination of 16 separate multiclass NB classifiers, each trained on a specific L1 and applied to the corresponding parts of the data. `Classifier2` is a replication of the classifier presented in Rozovskaya and Roth (2011), which uses the priors adapted to the writer's L1 and to the chosen determiner or preposition at decision time. The priors used for these runs were estimated from the HOO training data.

We present only the results of the systems that use `H+D` features for prepositions, since these systems outperform systems using `H` only. Tables 1, 2 and 3 below show the comparative results of the three classifiers averaged over 5 runs, with all errors, determiner errors only, and preposition errors only, respectively.

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
U	60.69	**21.32**	**31.55**	**50.57**	**17.73**	**26.25**	**34.38**	**12.05**	**17.85**
C1	**64.51**	16.17	25.85	50.25	12.56	20.10	30.95	7.74	12.39
C2	33.74	16.51	22.15	28.50	13.96	18.72	16.51	8.10	10.85

Table 1: All errors included. Unadapted classifier (U) vs. two L1-adapted classifiers (C1 and C2). Results on the training set.

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
U	54.42	**33.25**	**41.25**	50.09	**30.60**	**30.83**	**40.70**	**24.84**	**30.83**
C1	**61.19**	20.25	30.42	**52.20**	17.27	25.94	40.57	13.43	20.17
C2	40.56	15.88	22.81	37.24	14.58	20.94	23.20	9.08	13.04

Table 2: Determiner errors. Unadapted classifier (U) vs. two L1-adapted classifiers (C1 and C2). Results on the training set.

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
U	65.71	16.89	**26.87**	**50.90**	**13.09**	**20.83**	**28.95**	**7.45**	**11.84**
C1	**66.96**	13.86	22.97	48.51	10.05	16.65	22.70	4.70	7.79
C2	27.45	**17.06**	21.00	21.00	13.07	16.09	10.79	6.73	8.27

Table 3: Preposition errors. Unadapted classifier (U) vs. two L1-adapted classifiers (C1 and C2). Results on the training set.

The results show some improvement with a combination of classifiers trained on L1-subsets in terms of recall for detection and recognition of errors, and a slight improvement in precision using L1-specific priors for preposition errors. However, in general, unadapted classifiers outperform L1-adapted classifiers with identical feature types. Therefore, we have not included L1-specific classifiers in the submitted set of runs.

5.2 Submitted systems

For the official runs, we trained various versions of the unadapted and generic adapted NB classifiers.

[5] For precise definitions of these measures see `www.correcttext.org/hoo2012`

We trained all the adapted priors on the full CLC dataset in the expectation that this would yield more accurate estimates. We trained the unadapted priors and the NB features as before on the HOO training dataset. We also trained the NB features on the full CLC dataset and tested the impact of the preposition feature `D` (dependent lemma of the GR from the preposition, i.e., the head of the preposition complement) with the different training set sizes. For all runs we used the full set of determiner features described in section 3.1.

The full set of multiclass NB classifiers submitted is described below:

- `Run1`: unadapted, trained on the HOO data. `H` feature for prepositions;
- `Run2`: unadapted, trained on the HOO data. `H` and `D` features for prepositions;
- `Run3`: a combination of the NB classifiers trained for each of the used candidate words separately. `H` and `D` features are used for prepositions;
- `Run4`: generic adapted, trained on HOO data. `H` feature for prepositions;
- `Run5`: generic adapted, trained on HOO data. `H` and `D` features for prepositions;
- `Run6`: unadapted, trained on the full CLC. `H` feature for prepositions;
- `Run7`: unadapted, trained on the full CLC. `H` and `D` features for prepositions.

The classifiers used for runs 1 and 2 differ from the ones used for runs 6 and 7 only in the amount of training data. None of these classifiers involve any adaptation. The classifiers used for runs 4 and 5 involve prior adaptation to the input determiner or preposition, adjusted at decision time. In run 3, a combination of classifiers trained on the input determiner- or preposition-specific partitions of the HOO training data are used. At test time, the appropriate classifier from this set is applied depending on the preposition or determiner chosen by the learner.

To limit the number of classes for the classifiers used in runs 1–3 and 6–7, we have combined the training cases for determiners *a* and *an* in one class *a/an*; after classification one of the variants is chosen depending on the first letter of the next word. However, for the classifiers used in runs 4–5, we used priors including confusions between *a* and *an*.

The results for these runs on the training data are shown in Tables 4 to 6 below.

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	5.54	**81.08**	10.37	5.32	**77.95**	9.97	4.90	**71.70**	9.17
1	60.14	18.57	28.37	48.21	14.88	22.74	32.71	10.09	15.43
2	60.69	21.32	31.55	50.57	17.73	26.25	**34.38**	12.05	17.85
3	50.09	27.54	**35.52**	45.99	25.23	**32.57**	28.78	15.80	**20.39**
4	25.39	25.48	25.39	22.10	22.23	22.13	12.23	12.33	12.26
5	31.17	22.33	25.94	26.28	18.88	21.90	14.50	10.46	12.11
6	62.41	10.73	18.31	49.95	8.57	14.63	32.66	5.60	9.57
7	**62.92**	11.60	19.59	**52.29**	9.61	16.24	34.32	6.31	10.66

Table 4: Training set results, all errors

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	5.02	**82.98**	9.46	5.02	**82.98**	9.46	4.81	**79.57**	9.07
1–2	54.42	33.25	41.25	50.09	30.60	30.83	40.70	24.84	30.83
3	58.50	62.22	**60.22**	**57.41**	61.07	**59.11**	**46.33**	49.25	**47.68**
4–5	34.93	31.09	32.68	33.66	30.01	31.52	19.74	17.66	18.51
6–7	**58.65**	8.11	14.24	53.90	7.43	13.06	40.61	5.60	9.84

Table 5: Training set results, determiner errors

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	5.87	**78.30**	10.93	5.59	**74.49**	10.40	4.97	**66.28**	9.25
1	64.71	14.04	23.06	46.54	10.11	16.61	25.86	5.61	9.22
2	**65.71**	16.89	**26.87**	**50.90**	13.09	**20.83**	**28.95**	7.45	**11.84**
3	42.63	16.53	23.81	36.18	14.04	20.22	13.74	5.35	7.70
4	16.85	19.27	17.97	12.24	14.03	13.06	5.81	6.67	6.21
5	27.49	16.89	20.88	19.96	12.30	15.19	10.03	6.20	7.65
6	64.69	14.03	23.06	46.51	10.10	16.60	25.83	5.61	9.22
7	65.68	16.89	**26.87**	50.87	13.09	**20.82**	28.92	7.44	**11.84**

Table 6: Training set results, preposition errors

The results on the training data show that use of the `D` feature improves the performance of all the preposition classifiers. Use of the full CLC for training improves recall, but does not improve precision for prepositions, while for determiners precision of the classifiers trained on the full CLC is much worse. Adaptation of the classifiers with determiner/preposition-specific priors slightly improves precision on prepositions, but is damaging for recall. Therefore, in terms of F-score, unadapted classifiers outperform adapted ones. The overall best-performing system on the cross-validated training data is Run3, which is trained on the determiner/preposition-specific data subsets and applies an input-specific classifier to test data. However, the result is due to improved performance on determiners, not prepositions.

6 Official Evaluation Results

The results presented below are calculated using the evaluation tool provided by the organizers, implementing the scheme specified in the HOO shared task. The results on the test set, presented in Tables 7–9 are from the final official run after correction of errors in the annotation and score calculation scripts.

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	4.86	**76.67**	9.15	4.65	**73.33**	8.75	4.65	**73.33**	8.75
1	34.46	13.04	18.92	22.83	8.64	12.54	13.53	5.12	7.43
2	35.73	14.04	**20.16**	23.47	9.22	13.24	12.26	4.82	6.92
3	19.24	12.10	14.86	14.59	9.18	11.27	5.71	3.59	4.41
4	9.51	14.95	11.63	7.19	11.30	8.79	5.29	8.31	6.46
5	15.43	14.31	14.85	10.78	10.00	10.38	6.77	6.28	6.51
6	55.60	11.15	18.58	41.86	8.40	13.99	**28.54**	5.73	**9.54**
7	**56.66**	11.59	19.24	**42.49**	8.69	**14.43**	27.27	5.58	9.26

Table 7: Test set results, all errors

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	4.37	**83.33**	8.30	4.37	**83.33**	8.30	4.37	**83.33**	8.30
1–2	8.73	7.61	8.13	4.80	4.18	4.47	4.37	3.80	4.07
3	6.11	11.29	7.93	5.24	9.68	6.80	5.24	9.68	6.80
4–5	6.11	9.72	7.51	4.80	7.64	5.90	4.80	7.64	5.90
6–7	**51.09**	8.53	**14.63**	**44.10**	7.37	**12.63**	**35.37**	5.91	**10.13**

Table 8: Test set results, determiner errors

	Detection			Recognition			Correction		
	R	P	F	R	P	F	R	P	F
0	5.33	**72.22**	9.92	4.92	**66.67**	9.16	4.92	**66.67**	**9.16**
1	57.79	14.29	22.91	39.75	9.83	15.76	**22.13**	5.47	8.77
2	**59.43**	15.41	**24.47**	**40.98**	10.63	**16.88**	19.67	5.10	8.10
3	29.10	11.31	16.28	23.36	9.08	13.07	6.15	2.39	3.44
4	12.71	19.75	15.46	9.43	14.65	11.47	5.74	8.92	6.98
5	24.18	16.12	19.34	16.39	10.93	13.12	8.61	5.74	6.89
6	57.79	14.29	22.91	39.75	9.83	15.76	**22.13**	5.47	8.77
7	**59.43**	15.41	**24.47**	**40.98**	10.63	**16.88**	19.67	5.10	8.10

Table 9: Test set results, preposition errors

The test set results for NB classifiers (Runs 1–7) are significantly worse than our preliminary results obtained on the training data partitions, especially for determiners. Use of additional training data (Runs 6 and 7) improves recall, but does not improve precision. Adaptation to the input preposition improves precision as compared to the unadapted classifier for prepositions (Run 4), whereas training on the determiner-specific subsets improves precision for determiners (Run 3). However, generally these results are worse than the results of the similar classifiers on the training data subsets.

We calculated the upper bound recall for our classifiers on the test data. The upper bound recall on the test data is 93.20 for recognition, and 86.39 for correction, given our confusion sets for both determiners and prepositions. However, the actual upper bound recall is 71.82, with upper bound recall on determiners at 71.74 and on prepositions at 71.90, because 65 out of 230 determiner errors, and 68 out of 243 preposition errors are not considered by our classifiers, primarily because when the parser fails to find a full analysis, the grammatical context is often not recovered accurately enough to identify missing input positions or relevant GRs. This is an inherent weakness of using only parser-extracted features from noisy and often ungrammatical input. Taking this into account, some models (Runs 1, 2, 6 and 7) achieved quite high recall.

We suspect the considerable drop in precision is explained by the differences in the training and test data. The training set contains answers from learners of a smaller group of L1s from one examination year to a much more restricted set of prompts. The well-known weaknesses of generative NB classifiers may prevent effective exploitation of the additional information in the full CLC over the HOO training data. Experimentation with count weighting schemes and optimized interpolation of adapted priors may well be beneficial (Rennie et al., 2003).

Acknowledgements

We thank Cambridge ESOL, a division of Cambridge Assessment for a partial grant to the first author and a research contract with iLexIR Ltd. We also thank them and Cambridge University Press for granting us access to the CLC for research purposes.

References

Øistein Andersen. 2011 Semi-automatic ESOL error annotation. *English Profile Journal, vol2:e1*. DOI: 10.1017/S2041536211000018, Cambridge University Press.

Øistein Andersen, Julian Nioche, Ted Briscoe, and John Carroll. 2008 The BNC parsed with

RASP4UIMA. In *6th Int. Conf. on Language Resources and Evaluation (LREC)*, Marrakech, Moroc-cco

Ted Briscoe, John A. Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of COLING/ACL*, vol 6.

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 Innovative Use of NLP for Building Educational Applications* Montreal, Canada, June.

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

Jens Eeg-Olofsson and Ola Knutsson. 2003. Automatic grammar checking for second language learners - the use of prepositions. In *Nodalida*.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexander Klementiev, William Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of IJCNLP*, Hyderabad, India, January.

Michael Gamon. 2010. Using Mostly Native Data to Correct Errors in Learners' Writing: A Meta-Classifier Approach. In *Proceedings of NAACL 2010*, pages 163–171, Los Angeles, USA, June.

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

Na-Rae Han, Joel R. Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an Error-Annotated Learner Corpus to Develop an ESL/EFL Error Correction System. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC-10)*, Valletta, Malta, May.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148, Sapporo, Japan, July.

Ekaterina Kochmar. 2011. Identification of a Writer's Native Language by Error Analysis University of Cambridge, MPhil Dissertation.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Morgan & Claypool Publishers.

John Lee and Stephanie Seneff. 2008. An analysis of grammatical errors in non-native speech in English. In *Proceedings of the 2008 Spoken Language Technology Workshop*.

Ryo Nagata and Kazuhide Nakatani. 2010 Evaluating performance of grammatical error detection to maximize learning effect. In *Proceedings of Int. Conf. on Computational Linguistics (Coling-10), Poster Session*, pages 894–900, Beijing, China.

Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics conference*, pages 572–581.

Jason Rennie, Lawrence Shih, Jaime Teevan, and David Karger. 2003 Tackling the Poor Assumtions of Naive Bayes Text Classifiers. 20th Int. Conference on Machine Learning (ICML-2003) Washington, DC

Alla Rozovskaya and Dan Roth. 2010a. Annotating ESL Errors: Challenges and Rewards. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Alla Rozovskaya and Dan Roth. 2010b. Generating Confusion Sets for Context-Sensitive Error Correction. In *of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alla Rozovskaya and Dan Roth. 2010c. Training Paradigms for Correcting Errors in Grammar and Usage. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872, Manchester, UK, August.

Joel R. Tetreault, Jennifer Foster, Martin Chodorow. 2010. Using Parse Features for Preposition Selection and Error Detection. In *ACL*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.