Training Paradigms for Correcting Errors in Grammar and Usage

Alla Rozovskaya and Dan Roth
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{rozovska,danr}@illinois.edu

Abstract

This paper proposes a novel approach to the problem of training classifiers to detect and correct grammar and usage errors in text by selectively introducing mistakes into the training data. When training a classifier, we would like the distribution of examples seen in training to be as similar as possible to the one seen in testing. In error correction problems, such as correcting mistakes made by second language learners, a system is generally trained on correct data, since annotating data for training is expensive. Error generation methods avoid expensive data annotation and create training data that resemble non-native data with errors.

We apply error generation methods and train classifiers for detecting and correcting article errors in essays written by non-native English speakers; we show that training on data that contain errors produces higher accuracy when compared to a system that is trained on clean native data. We propose several training paradigms with error generation and show that each such paradigm is superior to training a classifier on native data. We also show that the most successful error generation methods are those that use knowledge about the article distribution and error patterns observed in non-native text.

1 Introduction

This paper considers the problem of training classifiers to detect and correct errors in grammar and word usage in text. Both native and non-native speakers make a variety of errors that are not always easy to detect. Consider, for example, the problem of context-sensitive spelling correction (e.g., (Golding and Roth, 1996; Golding and Roth, 1999; Carlson et al., 2001)). Unlike spelling errors that result in non-words and are easy to detect, context-sensitive spelling correction task involves correcting spelling errors that result in legitimate words, such as confusing *peace* and *piece* or *your* and *you're*. The typical training paradigm for these context-sensitive ambiguities is to use text assumed to be error free, replacing each target word occurrence (e.g. *peace*) with a *confusion set* consisting of, say {*peace, piece*}, thus generating both positive and negative examples, respectively, from the same context.

This paper proposes a novel error generation approach to the problem of training classifiers for the purpose of detecting and correcting grammar and usage errors in text. Unlike previous work (e.g., (Sjöbergh and Knutsson, 2005; Brockett et al., 2006; Foster and Andersen, 2009)), we selectively introduce mistakes in an appropriate proportion. In particular, to create training data that closely resemble text with naturally occurring errors, we use error frequency information and error distribution statistics obtained from corrected non-native text. We apply the method to the problem of detecting and correcting article mistakes made by learners of English as a Second Language (ESL).

The problem of correcting article errors is generally viewed as that of article *selection*, cast as a classification problem and is trained as described above: a machine learning algorithm is used to train a classifier on native English data, where the possible selections are used to generate positive and negative

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 154–162,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

examples (e.g., (Izumi et al., 2003; Han et al., 2006; De Felice and Pulman, 2008; Gamon et al., 2008)). The classifier is then applied to non-native text to predict the correct article in context. But the article *correction* problem differs from the problem of article selection in that we know the original (*source*) article that the writer used. When proposing a correction, we would like to use information about the original article. One reason for this is that about 90% of articles are used correctly by ESL learners; this is higher than the performance of state-of-the-art classifiers for article selection. Consequently, not using the writer's article, when making a prediction, may result in making more mistakes than there are in the data. Another reason is that statistics on article errors (e.g., (Han et al., 2006; Lee and Seneff, 2008)) and in the annotation performed for the present study reveal that non-native English speakers make article mistakes in a consistent manner.

The system can consider the article used by the writer at evaluation time, by proposing a correction only when the confidence of the classifier is high enough, but the article cannot be used in training if the classifier is trained on clean native data that do not have errors. Learning Theory says that the distribution of examples seen in testing should be as similar as possible to the one seen in training, so one would like to train on errors similar to those observed in testing. Ideally, we would like to train using corrected non-native text. In that case, the original article of the writer can be used as a feature for the classifier and the correct article, as judged by a native English speaker, will be viewed as the label. However, obtaining annotated data for training is expensive and, since the native training data do not contain errors, we cannot use the writer's article as a feature for the classifier.

This paper compares the traditional training paradigm that uses native data to training paradigms that use data with artificial mistakes. We propose several methods of generating mistakes in native training data and demonstrate that they outperform the traditional training paradigm. We also show that the most successful error generation methods use knowledge about the article distribution and error patterns observed in the ESL data.

The rest of the paper is organized as follows. First, we discuss the baseline on the error correction task and show why the baselines used in selection tasks are not relevant for the error correction task. Next, we describe prior work in error generation and show the key difference of our approach. Section 4 presents the ESL data that we use and statistics on article errors. Section 5 describes training paradigms that employ error generation. In Sections 6 and 7 we present the results and discuss the results. The key findings are summarized in Table 7 in Section 6. We conclude with a brief discussion of directions for future work.

2 Measuring Success in Error Correction Tasks

The distinction between the selection and the error correction tasks alluded to earlier is important not only for training but also in determining an appropriate evaluation method.

The standard baseline used in selection tasks is the relative frequency of the most common class. For example, in word sense disambiguation, the baseline is the most frequent sense. In the task of article selection, the standard baseline used is to predict the article that occurs most frequently in the data (usually, it is the *zero* article, whose frequency is 60-70%). In this context, the performance of a state-of-the-art classifier (Knight and Chander, 1994; Minnen et al., 2000; Turner and Charniak, 2007; Gamon et al., 2008) whose accuracy is 85-87% is a significant improvement over the baseline. The majority has been used as the baseline also in the context-sensitive spelling task (e.g., (Golding and Roth, 1999)).

However, in article correction, spelling correction, and other text correction applications the split of the classes is not an appropriate baseline since the majority of the words in the confusion set are used correctly in the text. Han et al. (2006) report an average error rate of 13% on article data from TOEFL essays, which gives a baseline of 87%, versus the baseline of 60-70% used in the article selection task. Statistics on article mistakes in our data suggest a baseline of about 90%, depending on the source language of the writer. So the real baseline on the task is "do nothing". Therefore, to determine the baseline for a correction task, one needs to consider the error rate in the data.

Using the definitions of precision and recall and the "real" baseline, we can also relate the resulting accuracy of the classifier to the precision and recall on an error correction task as follows: Let P and R denote the precision and recall, respectively, of the system on an error correction task, and $Base$ denote the error rate in the data. Then the task baseline (i.e., accuracy of the data before running the system) is:

$$Baseline = 1 - Base$$

It can be shown that the error rate after running the classifier is:

$$Error = \frac{Base * (P + R - 2RP)}{P}$$

It follows that the accuracy of the system on the task is $1 - Error$.

For example, we can obtain a rough estimate on the accuracy of the system in Han et al. (2006), using precision and recall numbers by error type. Excluding the error type of category *other*, we can estimate that $Base = 0.1$, so the baseline is 0.9, average precision and recall are 0.85 and 0.25, respectively, and the resulting overall accuracy of the system is 92.2%.

3 Related Work

3.1 Generating Errors in Text

In text correction, adding mistakes in training has been explored before. Although the general approach has been to produce errors similar to those observed in the data to be corrected, mistakes were added in an ad-hoc way, without respecting the error frequencies and error patterns observed in non-native text. Izumi et al. (2003) train a maximum entropy model on error-tagged data from the Japanese Learners of English corpus (JLE, (Izumi et al., 2004)) to detect 8 error types in the same corpus. They show improvement when the training set is enhanced with sentences from the same corpus to which artificial article mistakes have been added. Though it is not clear in what proportion mistakes were added, it is also possible that the improvement was due to a larger training set. Foster and Andersen (2009) attempt to replicate naturally occurring learner mistakes in the Cambridge Learner Corpus (CLC)[1], but show a drop in accuracy when the original error-tagged data in training are replaced with corrected CLC sentences containing artificial errors.

Brockett et al. (2006) generate mass noun errors in native English data using relevant examples found in the Chinese Learners English Corpus (CLEC, (Gui and Yang, 2003)). Training data consist of an equal number of correct and incorrect sentences. Sjöbergh and Knutsson (2005) introduce split compound and agreement errors into native Swedish text: agreement errors are added in every sentence and for compound errors, the training set consists of an equal number of negative and positive examples. Their method gives higher recall at the expense of lower precision compared to rule-based grammar checkers.

To sum up, although the idea of using data with artificial mistakes is not new, the advantage of training on such data has not been investigated. Moreover, training on error-tagged data is currently unrealistic in the majority of error correction scenarios, which suggests that using text with artificial mistakes is the only alternative to using clean data. However, it has not been shown whether training on data with artificial errors is beneficial when compared to utilizing clean data. More importantly, error statistics have not been considered for error correction tasks. Lee and Seneff (2008) examine statistics on article and preposition mistakes in the JLE corpus. While they do not suggest a specific approach, they hypothesize that it might be helpful to incorporate this knowledge into a correction system that targets these two language phenomena.

3.2 Approaches to Detecting Article Mistakes

Automated methods for detecting article mistakes generally use a machine learning algorithm. Gamon et al. (2008) use a decision tree model and a 5-gram language model trained on the English Gigaword corpus (LDC2005T12) to correct errors in English article and preposition usage. Han et al. (2006) and De Felice and Pulman (2008) train a maximum entropy classifier. Yi et al. (2008) propose a web count-based system to correct determiner errors. In the above approaches, the classifiers are trained on native data. Therefore the classifiers cannot use the

[1] http://www.cambridge.org/elt

original article that the writer used as a feature. Han et al. (2006) use the *source* article at evaluation time and propose a correction only when the score of the classifier is high enough, but the *source* article is not used in training.

4 Article Errors in ESL Data

Article errors are one of the most common mistakes that non-native speakers make, especially those whose native language does not have an article system. For example, Han et al. (2006) report that in the annotated TOEFL data by Russian, Chinese, and Japanese speakers 13% of all noun phrases have an incorrect article. It is interesting to note that article errors are present even with very advanced speakers. While the TOEFL data include essays by students of different proficiency levels, we use data from very advanced learners and find that error rates on articles are similar to those reported by Han et al. (2006).

We use data from speakers of three first language backgrounds: Chinese, Czech, and Russian. None of these languages has an article system. The Czech and the Russian data come from the ICLE corpus (Granger et al., 2002), which is a collection of essays written by advanced learners of English. The Chinese data is a part of the CLEC corpus that contains essays by students of all levels of proficiency.

4.1 Data Annotation

A portion of data for each source language was corrected and error-tagged by native speakers. The annotation was performed at the sentence level: a sentence was presented to the annotator in the context of the entire essay. Essay context can become necessary, when an article is acceptable in the context of a sentence, but is incorrect in the context of the essay. Our goal was to correct all article errors, including those that, while acceptable in the context of the sentence, were not correct in the context of the essay. The annotators were also encouraged to propose more than one correction, as long as all of their suggestions were consistent with the essay context.

The annotators were asked to correct all mistakes in the sentence. The annotation schema included the following error types: mistakes in article and preposition usage, errors in *noun number*, *spelling*, *verb form*, and *word form*[2]. All other corrections were marked as *word replacement*, *word deletion*, and *word insertion*. For details about annotation and data selection, please refer to the companion paper (Rozovskaya and Roth, 2010).

4.2 Statistics on Article Errors

Traditionally, three article classes are distinguished: *the*, *a*(*an*)[3] and *None* (no article). The training and the test data are thus composed of two types of events:

1. All articles in the data

2. Spaces in front of a noun phrase if that noun phrase does not start with an article. To identify the beginning of a noun phrase, we ran a part-of-speech tagger and a phrase chunker[4] and excluded all noun phrases not headed[5] by a personal or demonstrative pronoun.

Table 1 shows the size of the test data by source language, proportion of errors and distribution of article classes before and after annotation and compares these distributions to the distribution of articles in English Wikipedia. The distribution before annotation shows statistics on article usage by the writers and the distribution after annotation shows statistics after the corrections made by the annotators were applied. As the table shows, the distribution of articles is quite different for native data (Wikipedia) and non-native text. In particular, non-native data have a lower proportion of *the*.

The annotation statistics also reveal that learners do not confuse articles randomly. From Table 2, which shows the distribution of article errors by type, we observe that the majority of mistakes are omissions and extraneous articles. Table 3 shows statistics on corrections by *source* and *label*, where *source* refers to the article used by the writer, and *label* refers to the article chosen by the annotator. Each entry in the table indicates $Prob(source =$

[2] Our classification, was inspired by the classification presented in Tetreault and Chodorow (2008)

[3] Henceforth, we will use *a* to refer to both *a* and *an*

[4] The tagger and the chunker are available at `http://L2R.cs.uiuc.edu/~cogcomp/software.php`

[5] We assume that the last word of the noun phrase is its head.

Source language	Number of test examples	Proportion of errors	Errors total	Article distribution	Classes		
					a	*the*	*None*
Chinese	1713	9.2%	158	Before annotation	8.5	28.2	63.3
				After annotation	9.9	24.9	65.2
Czech	1061	9.6%	102	Before annotation	9.1	22.9	68.0
				After annotation	9.9	22.3	67.8
Russian	2146	10.4%	224	Before annotation	10.5	21.7	67.9
				After annotation	12.5	20.1	67.4
English Wikipedia					9.6	29.1	61.4

Table 1: Statistics on articles in the annotated data before and after annotation.

Source language	Proportion of errors in the data	Errors total	Errors by Type			
			Extraneous	Missing a	Missing the	Confusion
Chinese	9.2%	158	57.0%	13.3%	22.8%	7.0%
Czech	9.6%	102	45.1%	14.7%	33.3%	6.9%
Russian	10.4%	224	41.5%	20.1%	25.5%	12.3%

Table 2: Distribution of article errors in the annotated data by error type. *Extraneous* refers to using *a* or *the* where *None* (no article) is correct. *Confusion* is using *a* instead of *the* or vice versa.

Label	Source language	Source		
		a	*the*	*None*
a	Chinese	81.7%	5.9%	12.4%
	Czech	81.0%	4.8%	14.3%
	Russian	75.3%	7.9%	16.9%
the	Chinese	0.2%	91.3%	8.5%
	Czech	0.9%	84.7%	14.4%
	Russian	1.9%	84.9%	13.2%
None	Chinese	0.6%	7.4%%	92.0%
	Czech	1.3%	5.2%	93.6%
	Russian	1.0%	5.4%%	93.6%

Table 3: Statistics on article corrections by the original article (*source*) and the annotator's choice (*label*). Each entry in the table indicates $Prob(source = s|label = l)$ for each article pair.

$s|label = l$) for each article pair. We can also observe specific error patterns. For example, *the* is more likely than *a* to be used superfluously.

5 Introducing Article Errors into Training Data

This section describes experiments with error generation methods. We conduct four sets of experiments. Each set differs in how article errors are generated in the training data. We now give a description of error generation paradigms in each experimental set.

5.1 Methods of error generation

We refer to the article that the writer used in the ESL data as *source*, and *label* refers to the article that the annotator chose. Similarly, when we introduce errors into the training data, we refer to the original article as *label* and to the replacement as *source*. This is because the original article is the correct article choice, and the replacement that the classifier will see as a feature can be an error. We call this feature *source* feature. In other words, both for training (native data) and test (ESL data), *source* denotes the form that the classifier sees as a feature (which could be an error) and *label* denotes the correct article. Below we describe how errors are generated in each set of experiments.

Method 1: General With probability x each article in the training data is replaced with a different article uniformly at random, and with probability $(1 - x)$ it remains unchanged. We build six classifiers, where $x \in \{5\%, 10\%, 12\%, 14\%, 16\%, 18\%\}$. We call this method *general* since it uses no information about article distribution in the ESL data.

Method 2: ArticleDistrBeforeAnnot We use the distribution of articles in the ESL data before the annotation to change the distribution of articles in the training. Specifically, we change the articles so that their distribution approximates the distribution of articles in the ESL data. For example, the relative frequency of *the* in English Wikipedia data is 29.1%, while in the writing by Czech speakers it is 22.3%. It should be noted that this method changes the distribution only of source articles, but the distribution of labels is not affected. An additional constraint that we impose is the minimum error rate r for each article class, so that $Prob(s|l) \geq r \; \forall l \in labels$. In this fashion, for each source language we train four classifiers, where we use article distribution from Chinese, Czech, and Russian, and where we set the minimum error rate r to be $\in \{2\%, 3\%, 4\%, 5\%\}$.

Method 3: ArticleDistrAfterAnnot This method is similar to the one above but we use the distribution of articles in the ESL data after the corrections have been made by the annotators.

Method 4: ErrorDistr This method uses information about error patterns in the annotated ESL data. For example, in the Czech annotated subcorpus, label *the* corresponds to source *the* in 85% of the cases and corresponds to source *None* in 14% of the cases. In other words, in 14% of the cases where the article *the* should have been used, the writer used no article at all. Thus, with probability 14% we change *the* in the training data to *None*.

6 Experimental Results

In this section, we compare the quality of the system trained on clean native English data to the quality of the systems trained on data with errors. The errors were introduced into the training data using error generation methods presented in Section 5.

In each training paradigm, we follow a discriminative approach, using an online learning paradigm and making use of the Averaged Perceptron Algorithm (Freund and Schapire, 1999) implemented within the Sparse Network of Winnow framework (Carlson et al., 1999) – we use the regularized version in Learning Based Java[6] (LBJ, (Rizzolo and Roth, 2007)). While classical Perceptron comes with generalization bound related to the margin of the data, Averaged Perceptron also comes with a PAC-like generalization bound (Freund and Schapire, 1999). This linear learning algorithm is known, both theoretically and experimentally, to be among the best linear learning approaches and is competitive with SVM and Logistic Regression, while being more efficient in training. It also has been shown to produce state-of-the-art results on many natural language applications (Punyakanok et al., 2008).

Since the methods of error generation described in Section 5 rely on the distribution of articles and article mistakes and these statistics are specific to the first language of the writer, we conduct evaluation separately for each source language. Thus, for each language group, we train five system types: one system is trained on clean English data without errors (the same classifier for the three language groups) and four systems are trained on data with errors, where errors are produced using the four methods described in Section 5. Training data are extracted from English Wikipedia.

All of the five systems employ the same set of features based on three tokens to the right and to the left of the target article. For each context word, we use its relative position, its part-of-speech tag and the word token itself. We also use the head of the noun phrase and the conjunctions of the pairs and triples of the six tokens and their part-of-speech tags[7]. In addition to these features, the classifiers trained on data with errors also use the *source* article as a feature. The classifier that is trained on clean English data cannot use the *source* feature, since in training the *source* always corresponds to the *label*. By contrast, when the training data contain mistakes, the *source* is not always the same as the *label*, the situation that we also have with the test (ESL) data.

We refer to the classifier trained on clean data as $TrainClean$. We refer to the classifiers trained on data with mistakes as TWE (TrainWithErrors). There are four types of TWE systems for each language group, one for each of the methods of error generation described in Section 5. All results are the averaged results of training on three random samples from Wikipedia with two million training examples on each round. All five classifiers are trained on exactly the same set of Wikipedia examples, except that we add article mistakes to the data used by the TWE systems. The $TrainClean$ system achieves an accuracy of 87.10% on data from English Wikipedia. This performance is state-of-the-

[6] LBJ code is available at `http://L2R.cs.uiuc.edu/~cogcomp/asoftware.php?skey=LBJ`

[7] Details about the features are given in the paper's web page, accessible from `http://L2R.cs.uiuc.edu/~cogcomp/`

art compared to other systems reported in the literature (Knight and Chander, 1994; Minnen et al., 2000; Turner and Charniak, 2007; Han et al., 2006; De Felice and Pulman, 2008). The best results of 92.15% are reported by De Felice and Pulman (2008). But their system uses sophisticated syntactic features and they observe that the parser does not perform well on non-native data.

As mentioned in Section 4, the annotation of the ESL data consisted of correcting all errors in the sentence. We exclude from evaluation examples that have spelling errors in the 3-word window around the target article and errors on words that immediately precede or immediately follow the article, as such examples would obscure the evaluation of the training paradigms.

Tables 4, 5 and 6 show performance by language group. The tables show the accuracy and the error reduction on the test set. The results of systems TWE (methods 2 and 3) that use the distribution of articles before and after annotation are merged and appear as $ArtDistr$ in the tables, since, as shown in Table 1, these distributions are very similar and thus produce similar results. Each table compares the performance of the $TrainClean$ system to the performance of the four systems trained on data with errors.

For all language groups, all classifiers of type TWE outperform the $TrainClean$ system. The reduction in error rate is consistent when the TWE classifiers are compared to the $TrainClean$ system.

Table 7 shows results for all three languages, comparing for each language group the $TrainClean$ classifier to the best performing system of type TWE.

Training paradigm	**Errors in training**	**Accuracy**	**Error reduction**
$TrainClean$	0.0%	91.85%	-2.26%
$TWE(General)$	10.0%	92.57%	6.78%
$TWE(ArtDistr)$	13.2%	92.67%	**8.33%**
$TWE(ErrorDistr)$	9.2%	92.31%	3.51%
Baseline		92.03%	

Table 4: **Chinese speakers**: Performance of the $TrainClean$ system (without errors in training) and of the best classifiers of type TWE. Rows 2-4 show the performance of the systems trained with error generation methods described in 5. *Error reduction* denotes the percentage reduction in the number of errors when compared to the number of errors in the ESL data.

Training paradigm	**Errors in training**	**Accuracy**	**Error reduction**
$TrainClean$	0.0%	91.82%	10.31%
$TWE(General)$	18.0%	92.22%	**14.69%**
$TWE(ArtDistr)$	21.6%	92.00%	12.28%
$TWE(ErrorDistr)$	10.2%	92.15%	13.93%
Baseline		90.88%	

Table 5: **Czech speakers**: Performance of the $TrainClean$ system (without errors in training) and of the best classifiers of type TWE. Rows 2-4 show the performance of the systems trained with error generation methods described in 5. *Error reduction* denotes the percentage reduction in the number of errors when compared to the number of errors in the ESL data.

Training paradigm	**Errors in training**	**Accuracy**	**Error reduction**
$TrainClean$	0.0%	90.62%	5.92%
$TWE(General)$	14.0%	91.25%	12.24%
$TWE(ArtDistr)$	18.8%	91.52%	14.94%
$TWE(ErrorDistr)$	10.7%	91.63%	**16.05%**
Baseline		90.03%	

Table 6: **Russian speakers**: Performance of the $TrainClean$ system (without errors in training) and of the best classifiers of type TWE. Rows 2-4 show the performance of the systems trained with error generation methods described in 5. *Error reduction* denotes the percentage reduction in the number of errors when compared to the number of errors in the ESL data.

7 Discussion

As shown in Section 6, training a classifier on data that contain errors produces better results when compared to the $TrainClean$ classifier trained on clean native data. The key results for all language groups are summarized in Table 7. It should be noted that the $TrainClean$ system also makes use of the article chosen by the author through a confidence threshold[8]; it prefers to keep the article chosen by the user. The difference is that the $TrainClean$ system does not consider the author's article in training. The results of training with error generation are better, which shows that training on automatically corrupted data indeed helps. While the performance is different by language group, there is an observable reduction in error rate for each language group when TWE systems are used compared to $TrainClean$ approach. The reduction in error rate

[8] The decision threshold is found empirically on a subset of the ESL data set aside for development.

achieved by the best performing TWE system when compared to the error rate of the $TrainClean$ system is 10.06% for Chinese, 4.89% for Czech and 10.77% for Russian, as shown in Table 7. We also note that the best performing TWE systems for Chinese and Russian speakers are those that rely on the distribution of articles (Chinese) and the distribution of errors (Russian), but for Czech it is the $General$ TWE system that performs the best, maybe because we had less data for Czech speakers, so their statistics are less reliable.

There are several additional observations to be made. First, training paradigms that use error generation methods work better than the training approach of using clean data. Every system of type TWE outperforms the $TrainClean$ system, as evidenced by Tables 4, 5, and 6. Second, the proportion of errors in the training data should be similar to the error rate in the test data. The proportion of errors in training is shown in Tables 4, 5 and 6 in column 2. Furthermore, TWE systems $ArtDistr$ and $ErrorDistr$ that use specific knowledge about article and error distributions, respectively, work better for Russian and Chinese groups than the $General$ method that adds errors to the data uniformly at random. Since $ArtDistr$ and $ErrorDistr$ depend on the statistics of learner mistakes, the success of the systems that use these methods for error generation depends on the accuracy of these statistics, and we only have between 100 and 250 errors for each language group. It would be interesting to see whether better results can be achieved with these methods if more annotated data are available. Finally, for the same reason, there is no significant difference in the performance of methods $ArtDistrBeforeAnnot$ and $ArtDistrAfterAnnot$: With small sizes of annotated data there is no difference in article distributions before and after annotation.

Source language	**Accuracy**		**Error reduction**
	$Train$ $Clean$	TWE	
Chinese	91.85%	92.67%	**10.06%**
Czech	91.82%	92.22%	**4.89%**
Russian	90.62%	91.63%	**10.77%**

Table 7: **Improvement due to training with errors**. For each source language, the last column of the table shows the reduction in error rate achieved by the best performing TWE system when compared to the error rate of the $TrainClean$ system. The error rate for each system is computed by subtracting the accuracy achieved by the system, as shown in columns 2 and 3.

8 Conclusion and Future Work

We have shown that error correction training paradigms that introduce artificial errors are superior to training classifiers on clean data. We proposed several methods of error generation that account for error frequency information and error distribution statistics from non-native text and demonstrated that the methods that work best are those that result in a training corpus that statistically resembles the non-native text. Adding information about article distribution in non-native data and statistics on specific error types is even more helpful.

We have also argued that the baselines used earlier in the relevant literature – all based on the majority of the most commonly used class – suit selection tasks, but are inappropriate for error correction. Instead, the error rate in the data should be taken into account when determining the baseline.

The focus of the present study was on training paradigms. While it is quite possible that the article correction system presented here can be improved – we would like to explore improving the system by using a more sophisticated feature set – we believe that the performance gap due to the error driven training paradigms shown here will remain. The reason is that even with better features, some of the features that hold in the native data will not be active in in the ESL writing.

Finally, while this study focused on the problem of correcting article mistakes, we plan to apply the proposed training paradigms to similar text correction problems.

Acknowledgments

We thank Nick Rizzolo for helpful discussions on LBJ. We also thank Peter Chew and the anonymous reviewers for their insightful comments. This research is partly supported by a grant from the U.S. Department of Education.

References

C. Brockett, W. B. Dolan, and M. Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st COLING and the 44th ACL*, Sydney.

A. Carlson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. *Technical report*.

A. J. Carlson and J. Rosen and D. Roth. 2001. Scaling Up Context Sensitive Text Correction. *IAAI*, 45–50.

R. De Felice and S. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of COLING-08*.

J. Foster and Ø. Andersen. 2009. GenERRate: Generating Errors for Use in Grammatical Error Detection. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

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

M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. Dolan, D. Belenko and L. Vanderwende. 2008. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. *Proceedings of IJCNLP*.

A. R. Golding and D. Roth. 1996. Applying Winnow to Context-Sensitive Spelling Correction. *ICML*, 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.

S. Granger, E. Dagneaux and F. Meunier 2002. *International Corpus of Learner English*.

S. Gui and H. Yang. 2003. *Zhongguo Xuexizhe Yingyu Yuliaohu. (Chinese Learner English Corpus)*. Shanghai Waiyu Jiaoyu Chubanshe. (In Chinese).

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.

E. Izumi, K. Uchimoto, T. Saiga and H. Isahara. 2003. Automatic Error Detection in the Japanese Leaners English Spoken Data. *ACL*.

E. Izumi, K. Uchimoto and H. Isahara. 2004. The NICT JLE Corpus: Exploiting the Language Learner's Speech Database for Research and Education. *International Journal of the Computer, the Internet and Management*, 12(2):119–125.

K. Knight and I. Chander. 1994. Automatic Postediting of Documents. In *Proceedings of the American Association of Artificial Intelligence*, pp 779–784.

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

G. Minnen, F. Bond and A. Copestake 2000. Memory-Based Learning for Article Generation. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop*, pp 43–48.

V. Punyakanok, D. Roth, and W. Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

N. Rizzolo and D. Roth 2007. Modeling Discriminative Global Inference. In *Proceedings of the First International Conference on Semantic Computing (ICSC)*, pp 597–604.

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

J. Sjöbergh and O. Knutsson. 2005. Faking errors to avoid making errors. In *Proceedings of RANLP 2005*, Borovets.

J. Tetreault and M. Chodorow. 2008. Native Judgments of Non-Native Usage: Experiments in Preposition Error Detection. *COLING Workshop on Human Judgments in Computational Linguistics*, Manchester, UK.

J. Turner and E. Charniak. 2007. Language Modeling for Determiner Selection. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pp 177–180.

J. Wagner, J. Foster, and J. van Genabith. 2009. Judging grammaticality: Experiments in sentence classification. *CALICO Journal*. Special Issue on the 2008 Automatic Analysis of Learner Language CALICO Workshop.

Y. Xing, J. Gao, and W. Dolan. 2009. A web-based English proofing system for ESL users. In *Proceedings of IJCNLP*.