Building a State-of-the-Art Grammatical Error Correction System

Alla Rozovskaya
Center for Computational Learning Systems
Columbia University
New York, NY 10115
alla@ccls.columbia.edu

Dan Roth
Department of Computer Science
University of Illinois
Urbana, IL 61801
danr@illinois.edu

Abstract

This paper identifies and examines the key principles underlying building a state-of-the-art grammatical error correction system. We do this by analyzing the Illinois system that placed first among seventeen teams in the recent CoNLL-2013 shared task on grammatical error correction.

The system focuses on five different types of errors common among non-native English writers. We describe four design principles that are relevant for correcting all of these errors, analyze the system along these dimensions, and show how each of these dimensions contributes to the performance.

1 Introduction

The field of text correction has seen an increased interest in the past several years, with a focus on correcting grammatical errors made by English as a Second Language (ESL) learners. Three competitions devoted to error correction for non-native writers took place recently: HOO-2011 (Dale and Kilgarriff, 2011), HOO-2012 (Dale et al., 2012), and the CoNLL-2013 shared task (Ng et al., 2013). The most recent and most prominent among these, the CoNLL-2013 shared task, covers several common ESL errors, including article and preposition usage mistakes, mistakes in noun number, and various verb errors, as illustrated in Fig. 1.[1] Seventeen teams that participated in the task developed a wide array of approaches that include discriminative classifiers, language models, statistical machine-translation systems, and rule-based modules. Many of the systems also made use of linguistic resources such as additional annotated learner corpora, and defined high-level features that take into account syntactic and semantic knowledge.

Nowadays **phone/phones* **has/have* many functionalities, **included/including* **∅/a* camera and **∅/a* Wi-Fi receiver.

Figure 1: Examples of representative ESL errors.

Even though the systems incorporated similar resources, the scores varied widely. The top system, from the University of Illinois, obtained an F1 score of 31.20[2], while the second team scored 25.01 and the median result was 8.48 points.[3] These results suggest that there is not enough understanding of what works best and what elements are essential for building a state-of-the-art error correction system.

In this paper, we identify key principles for building a robust grammatical error correction system and show their importance in the context of the shared task. We do this by analyzing the Illinois system and evaluating it along several dimensions: choice

[1] The CoNLL-2014 shared task that completed at the time of writing this paper was an extension of the CoNLL-2013 competition (Ng et al., 2014) but addressed all types of errors. The Illinois-Columbia submission, a slightly extended version of the Illinois CoNLL-2013 system, ranked at the top. For a description of the Illinois-Columbia submission, we refer the reader to Rozovskaya et al. (2014a).

[2] The state-of-the-art performance of the Illinois system discussed here is with respect to individual components for different errors. Improvements in Rozovskaya and Roth (2013) over the Illinois system that are due to joint learning and inference are orthogonal, and the analysis in this paper still applies there.

[3] F1 might not be the ideal metric for this task but this was the one chosen in the evaluation. See more in Sec. 6.

Transactions of the Association for Computational Linguistics, 2 (2014) 419–434. Action Editor: Alexander Koller.
Submitted 10/2013; Revised 6/2014; Published 10/2014.

of learning algorithm; choice of training data (native or annotated learner data); model adaptation to the mistakes made by the writers; and the use of linguistic knowledge. For each dimension, several implementations are compared, including, when possible, approaches chosen by other teams. We also validate the obtained results on another learner corpus. Overall, this paper makes two contributions: (1) we explain the success of the Illinois system, and (2) we provide an understanding and qualitative analysis of different dimensions that are essential for success in this task, with the goal of aiding future research on it. Given that the Illinois system has been the top system in four competitive evaluations over the last few years (HOO and CoNLL), we believe that the analysis we propose will be useful for researchers in this area.

In the next section, we present the CoNLL-2013 competition. Sec. 3 gives an overview of the approaches adopted by the top five teams. Sec. 4 describes the Illinois system. In Sec. 5, the analysis of the Illinois system is presented. Sec. 6 offers a brief discussion, and Sec. 7 concludes the paper.

2 Task Description

The CoNLL-2013 shared task focuses on five common mistakes made by ESL writers: *article/determiner*, *preposition*, *noun number*, *verb agreement*, *verb form*. The training data of the shared task is the NUCLE corpus (Dahlmeier et al., 2013), which contains essays written by learners of English (we also refer to it as *learner data* or *shared task training data*). The test data consists of 50 essays by students from the same linguistic background. The training and the test data contain 1.2M and 29K words, respectively.

Table 1 shows the number of errors by type and the error rates. Determiner errors are the most common and account for 42.1% of all errors in training. Note that the test data contains a much larger proportion of annotated mistakes; e.g. determiner errors occur four times more often in the test data than in the training data (only 2.4% of noun phrases in the training data have determiner errors, versus 10% in the test data). The differences might be attributed to differences in annotation standards, annotators, or writers, as the test data was annotated at a later time. The shared task provided two sets of test annotations: the original annotated data and a set with additional revisions that also includes alternative annotations proposed by participants. Clearly, having alternative answers is the right approach as there are typically multiple ways to correct an error. However, because the alternatives are based on the error analysis of the participating systems, the revised set may be biased (Ng et al., 2013). Consequently, we report results on the original set.

Error	**Number of errors and error rates**	
	Train	**Test**
Art.	6658 (2.4%)	690 (10.0%)
Prep.	2404 (2.0%)	311 (10.7%)
Noun	3779 (1.6%)	396 (6.0%)
Verb agr.	1527 (2.0%)	124 (5.2%)
Verb form	1453 (0.8%)	122 (2.5%)

Table 1: **Statistics on annotated errors in the CoNLL-2013 shared task data.** Percentage denotes the error rates, i.e. the number of erroneous instances with respect to the total number of relevant instances in the data.

3 Model Dimensions

Table 2 summarizes approaches and methodologies of the top five systems. The prevailing approach consists in building a statistical model either on learner data or on a much larger corpus of native English data. For native data, several teams make use of the Web 1T 5-gram corpus (henceforth Web1T, (Brants and Franz, 2006)). NARA employs a statistical machine translation model for two error types; two systems have rule-based components for selected errors. Based on the analysis of the Illinois system, we identify the following, inter-dependent, dimensions that will be examined in this work:

1. Learning algorithm: Most of the teams, including Illinois, built statistical models. We show that the choice of the learning algorithm is very important and affects the performance of the system.

2. Adaptation to learner errors: Previous studies, e.g. (Rozovskaya and Roth, 2011) showed that adaptation, i.e. developing models that utilize knowledge about error patterns of the non-native writers, is extremely important. We summarize adaptation techniques proposed earlier and examine their impact on the performance of the system.

3. Linguistic knowledge: It is essential to use some linguistic knowledge when developing error correction modules, e.g., to identify which type of verb

System	Error	Approach
Illinois (Rozovskaya et al., 2013)	Art.	AP model on NUCLE with word, POS, shallow parse features
	Prep.	NB model trained on Web1T and adapted to learner errors
	Noun/Agr./Form	NB model trained on Web1T
NTHU (Kao et al., 2013)	All	Count model with backoff trained on Web1T
HIT (Xiang et al., 2013)	Art./Prep./Noun	ME on NUCLE with word, POS, dependency features
	Agr./Form	Rule-based
NARA (Yoshimoto et al., 2013)	Art./Prep.	SMT model trained on learner data from Lang-8 corpus
	Noun	ME model on NUCLE with word, POS and dependency features
	Agr./Form	Treelet LM on Gigaword and Penn TreeBank corpora
UMC (Xing et al., 2013)	Art./Prep.	Two LMs – on NUCLE and Web1T corpus – with voting
	Noun	Rules and ME model on NUCLE + LM trained on Web1T
	Agr./Form	ME model on NUCLE (agr.) and rules (form)

Table 2: **Top systems in the CoNLL-2013 shared task.** The second column indicates the error type; the third column describes the approach adopted by the system. *ME* stands for Maximum Entropy; *LM* stands for language model; *SMT* stands for Statistical Machine Translation; *AP* stands for Averaged Perceptron; *NB* stands for Naïve Bayes.

	Classifier				
	Art.	**Prep.**	**Noun**	**Agr.**	**Form**
Train	254K	103K	240K	75K	175K
Test	6K	2.5K	2.6K	2.4K	4.8K

Table 3: **Number of candidate words by classifier type.**

error occurs in a given context, before the appropriate correction module is employed. We describe and evaluate the contribution of these elements.

4. Training data: We discuss the advantages of training on learner data or native English data in the context of the shared task and in broader context.

4 The Illinois System

The Illinois system consists of five machine-learning models, each specializing in correcting one of the errors described above. The words that are selected as input to a classifier are called *candidates* (Table 3). In the preposition system, for example, candidates are determined by surface forms. In other systems, determining the candidates might be more involved.

All modules take as input the corpus documents pre-processed with a part-of-speech tagger[4] (Even-Zohar and Roth, 2001) and shallow parser[5] (Punyakanok and Roth, 2001). In the Illinois submission, some modules are trained on native data, others on learner data. The modules trained on learner data make use of a discriminative algorithm, while native-trained modules make use of the Naïve Bayes (NB) algorithm. The Illinois system has an option for a post-processing step where corrections that always result in a false positive in training are ignored but this option is not used here.

4.1 Determiner Errors

The majority of determiner errors involve articles, although some errors also involve pronouns. The Illinois system addresses only article errors. Candidates include articles ("a","an","the")[6] and omissions, by considering noun-phrase-initial contexts where an article is likely to be omitted. The *confusion set* for articles is thus $\{a, the, \varnothing\}$. The article classifier is the same as the one in the HOO shared tasks (Rozovskaya et al., 2012; Rozovskaya et al., 2011), where it demonstrated superior performance. It is a discriminative model that makes use of the Averaged Perceptron algorithm (AP, (Freund and Schapire, 1996)) implemented with LBJava (Rizzolo and Roth, 2010) and is trained on learner data with rich features and adaptation to learner errors. See Sec. 5.2 and Sec. 5.3.

4.2 Preposition Errors

Similar to determiners, we distinguish three types of preposition mistakes: choosing an incorrect preposition, using a superfluous preposition, and omitting a preposition. In contrast to determiners, for learners of many first language backgrounds, most of the preposition errors are replacements, i.e., where the

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

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

[6] The variants "a" and "an" are collapsed to one class.

"Hence, the environmental factors also ***contributes/ contribute** to various difficulties, ***giving/given** problems in nuclear technology."	
Error	**Confusion set**
Agr.	{INF=contribute, S=contributes}
Form	{INF=give, ED=given, ING=giving, S=gives }

Table 4: **Confusion sets for agreement and form.** For irregular verbs, the second candidate in the confusion set for *Verb form* is the past participle.

author correctly recognized the need for a preposition, but chose the wrong one (Leacock et al., 2010). However, learner errors depend on the first language; in NUCLE, spurious prepositions occur more frequently: 29% versus 18% of all preposition mistakes in other learner corpora (Rozovskaya and Roth, 2010a; Yannakoudakis et al., 2011).

The Illinois preposition classifier is a NB model trained on Web1T that uses word n-gram features in the 4-word window around the preposition. The 4-word window refers to the four words before and the four words after the preposition, e.g. "problem as the search *of* alternative resources to the" for the preposition "of". Features consist of word n-grams of various lengths spanning the target preposition. For example, "the search of" is a 3-gram feature. The model is adapted to likely preposition confusions using the priors method (see Sec. 5.2). The Illinois model targets replacement errors of the 12 most common English prepositions. Here we augment it to identify spurious prepositions. The *confusion set* for prepositions is as follows: {*in, of, on, for, to, at, about, with, from, by, into, during,* ∅}.

4.3 Agreement and Form Errors

The Illinois system implements two verb modules – agreement and form – that consist of the following components: (1) candidate identification; (2) determining the relevant module for each candidate based on *verb finiteness*; (3) correction modules for each error type. The *confusion set* for verbs depends on the target word and includes its morphological variants (Table 4). For irregular verbs, the past participle form is included, while the past tense form is not (i.e. "given" is included but "gave" is not), since tense errors are not part of the task. To generate morphological variants, the system makes use of a morphological analyzer *verbMorph*; it assumes (1) a list of valid verb lemmas (compiled using a POS-tagged version of the NYT section of the Gigaword corpus) and (2) a list of irregular English verbs.[7]

Dimension	**Systems used in the comparison**
Learn. alg. (Sec. 5.1)	NTHU, UMC
Adaptation (Sec. 5.2)	Error inflation: HIT
Ling. knowledge (Sec. 5.3)	Cand. identification: NTHU, HIT Verb finiteness: NTHU
Train. data (Sec. 5.4)	HIT, NARA

Table 5: **System comparisons.** Column 1 indicates the dimension, and column 2 lists systems whose approaches provide a relevant point of comparison.

Candidate Identification stage selects the set of words that are presented as input to the classifier. This is a crucial step: errors missed at this stage will not be detected by the later stages. See Sec. 5.3.

Verb Finiteness is used in the Illinois system to separately process verbs that fulfill different grammatical functions and thus are marked for different grammatical properties. See Sec. 5.3.

Correction Modules The agreement module is a binary classifier. The form module is a 4-class system. Both classifiers are trained on the Web1T corpus.

4.4 Noun Errors

Noun number errors involve confusing singular and plural noun forms (e.g. "phone" instead of "phones" in Fig. 1) and are the second most common error type in the NUCLE corpus after determiner mistakes (Table 1). The Illinois noun module is trained on the Web1T corpus using NB. Similar to verbs, *candidate identification* is an important step in the noun classifier. See Sec. 5.3.

5 System Analysis

In this section, we evaluate the Illinois system along the four dimensions identified in Sec. 3, compare its components to alternative configurations implemented by other teams, and present additional experiments that further analyze each dimension. While a direct comparison with other systems is not always possible due to other differences between the systems, we believe that these results are still useful. Table 5 lists systems used for comparion. It is important to note that the dimensions are not independent. For instance, there is a correlation between algorithm choice and training data.

[7] The tool and more detail about it can be found at http://cogcomp.cs.illinois.edu/page/publication_view/743

Results are reported on the test data using F1 computed with the CoNLL scorer (Dahlmeier and Ng, 2012). Error-specific results are generated based on the output of individual modules. Note that these are not directly comparable to error-specific results in the CoNLL overview paper: the latter are approximate as the organizers did not have the error type information for corrections in the output. The complete system includes the union of corrections made by each of these modules, where the corrections are applied in order. Ordering overlapping candidates[8] might potentially affect the final output, when modules correctly identify an error but propose different corrections, but this does not happen in practice. Modules that are part of the Illinois submission are marked with an asterisk in all tables.

To demonstrate that our findings are not specific to CoNLL, we also show results on the FCE dataset. It is produced by learners from seventeen first language backgrounds and contains 500,000 words from the Cambridge Learner Corpus (CLC) (Yannakoudakis et al., 2011). We split the corpus into two equal parts – training and test. The statistics are shown in Appendix Tables A.16 and A.17.

5.1 Dim. 1: Learning Algorithm

Rozovskaya and Roth (2011, Sec. 3) discuss the relations between the amount of training data, learning algorithms, and the resulting performance. They show that on training sets of similar sizes, discriminative classifiers outperform other machine learning methods on this task. Following these results, the Illinois article module that is trained on the NUCLE corpus uses the discriminative approach AP. Most of the other teams that train on the NUCLE corpus also use a discriminative method.

However, when a very large native training set such as the Web1T corpus is available, it is often advantageous to use it. The Web1T corpus is a collection of n-gram counts of length one to five over a corpus of 10^{12} words. Since the corpus does not come with complete sentences, it is not straightforward to make use of a discriminative classifier because of the limited window provided around each example: training a discriminative model would limit the surrounding context features to a 2-word window. Because we wish to make use of the context features that extend beyond the 2-word window, it is only possible to use count-based methods, such as NB or LM. Several teams make use of the Web1T corpus: UMC uses a count-based LM for article, preposition, and noun number errors; NTHU addresses all errors with a count-based model with backoff, which is essentially a variation of a language model with backoff. The Illinois system employs the Web1T corpus for all errors, except articles, using NB.

Training Naïve Bayes for Deletions and Insertions The reason for not using the Web1T corpus for article errors is that training NB on Web1T for deletions and insertions presents a problem, and the majority of article errors are of this type. Recall that Web1T contains only n-gram counts, which makes it difficult to estimate the prior count for the $\varnothing$ candidate. (With access to complete sentences, the prior of $\varnothing$ is estimated by counting the total number of $\varnothing$ candidates; e.g., in case of articles, the number of NPs with $\varnothing$ article is computed.) We solve this problem by treating the article and the word following it as one target. For instance, to estimate prior counts for the article candidates in front of the word "camera" in "including camera", we obtain counts for "camera", "a camera", "the camera". In the case of the $\varnothing$ candidate, the word "camera" acts as the target. Thus, the confusion set for the article classifier is modified as follows: instead of the three articles (as shown in Sec. 4.1), each member of the confusion set is a concatenation of the article and the word that follows it, e.g. {a_camera, the_camera, camera}. The counts for contextual features are obtained similarly, e.g. a feature that includes a preceding word would correspond to the count of "including x", where x can take any value from the confusion set. The above solution allows us to train NB for article errors and to extend the preposition classifier to handle extraneous preposition errors (Table 6).

Rozovskaya and Roth (2011) study several algorithms trained on the Web1T corpus and observe that, when evaluated with the same context window size, NB performs better than other count-based methods. In order to show the impact of the algorithm choice, in Table 6, we compare LM and NB models. Both models use word n-grams spanning the target word in the 4-word window. We train LMs

[8]Overlapping candidates are included in more than one module: if "work" is tagged as *NN*, it is included in the noun module, but also in the form module (as a valid verb lemma).

Error	Model	F1	
		CoNLL	FCE
Art.	LM	21.11	24.15
	NB	**32.45**	**30.78**
Prep.	LM	12.09	**30.01**
	NB	**14.04**	29.40
Noun	LM	40.72	32.41
	NB*	**42.60**	**34.40**
Agr.	LM	20.65	33.53
	NB*	**26.46**	**36.42**
Form	LM	13.40	08.46
	NB*	**14.50**	**12.16**

Table 6: **Comparison of learning models.** Web1T corpus. Modules that are part of the Illinois submission are marked with an asterisk.

Source	Candidates			
	ED	INF	ING	S
ED	0.99675	0.00192	0.00103	0.00030
INF	0.00177	0.99630	0.00168	0.00025
ING	0.00124	0.00447	0.99407	0.00022
S	0.00054	0.00544	0.00132	0.99269

Table 7: **Priors confusion matrix used for adapting NB.** Each entry shows Prob(candidate|source), where source corresponds to the verb form chosen by the author.

with SRILM (Stolcke, 2002) using Jelinek-Mercer linear interpolation as a smoothing method (Chen and Goodman, 1996). On the CoNLL test data, NB outperforms LM on all errors; on the FCE corpus, NB is superior on all errors, except preposition errors, where LM outperforms NB only very slightly. We attribute this to the fact that the preposition problem has more labels; when there is a big confusion set, more features have default smooth weights, so there is no advantage to running NB. We found that with fewer classes (6 rather than 12 prepositions), NB outperforms LM. It is also possible that when we have a lot of labels, the theoretical difference between the algorithms disappears. Note that NB can be improved via adaptation (next section) and then it outperforms the LM also for preposition errors.

5.2 Dim. 2: Adaptation to Learner Errors

In the previous section, the models were trained on native data. These models have no notion of the error patterns of the learners. Here we discuss *model adaptation to learner errors*, i.e. developing models that utilize the knowledge about the types of mistakes learners make. Adaptation is based on the fact that learners make mistakes in a systematic manner, e.g. errors are influenced by the writer's first language (Gass and Selinker, 1992; Ionin et al., 2008).

There are different ways to adapt a model that depend on the type of training data (learner or native) and the algorithm choice. The key application of adaptation is for models trained on native English data, because the learned models do not know anything about the errors learners make. With adaptation, models trained on native data can use the author's word (the source word) as a feature and thus propose a correction based on what the author originally wrote. This is crucial, as the source word is an important piece of information (Rozovskaya and Roth, 2010b). Below, several adaptation techniques are summarized and evaluated. The Illinois system makes use of adaptation in the article model via the inflation method and adapts its NB preposition classifier trained on Web1T with the priors method.

Adapting NB The *priors method* (Rozovskaya and Roth, 2011, Sec. 4) is an adaptation technique for a NB model trained on native English data; it is based on changing the distribution of priors over the correction candidates. Candidate prior is a special parameter in NB; when NB is trained on native data, candidate priors correspond to the relative frequencies of the candidates in the native corpus and do not provide any information on the real distribution of mistakes and the dependence of the correction on the word used by the author.

In the priors method, candidate priors are changed using an error confusion matrix based on learner data that specifies how likely each confusion pair is. Table 7 shows the confusion matrix for verb form errors, computed on the NUCLE data. Adapted priors are dependent on the author's original verb form used: let *s* be a form of the verb appearing in the source text, and *c* a correction candidate. Then the adapted prior of *c* given *s* is:

$$prior(c|s) = \frac{C(s,c)}{C(s)}$$

where $C(s)$ denotes the number of times *s* appeared in the learner data, and $C(s,c)$ denotes the number of times *c* was the correct form when *s* was used by a writer. The adapted priors differ by the source: the probability of candidate *INF* when the source form is *S*, is more than twice than when the source form is

Error	Model	F1		
		CoNLL		FCE
		Train	Test	
Art.	NB	18.28	32.45	30.78
	NB-adapted	**19.18**	**34.49**	**31.76**
Prep.	NB	09.03	**14.04**	29.40
	NB-adapted*	**10.94**	12.14	**32.22**
Noun	NB*	**23.06**	**42.60**	**34.40**
	NB-adapted	22.89	42.31	32.38
Agr.	NB*	16.72	**26.46**	36.42
	NB-adapted	**17.62**	23.46	**38.57**
Form	NB*	11.93	14.50	12.16
	NB-adapted	**14.63**	**18.35**	**16.67**

Table 8: **Adapting NB with the priors method.** All models are trained on the Web1T corpus. Modules that are part of the Illinois submission are marked with an asterisk.

ED; the probability that S is the correct form is very high, which reflects the low error rates.

Table 8 compares NB and NB-adapted models. Because of the dichotomy in the error rates in CoNLL training and test data, we also show experiments using 5-fold cross-validation on the training data. Adaptation always helps on the CoNLL training data and the FCE data (except noun errors), but on the test data it only helps on article and verb form errors. This is due to discrepancies in the error rates, as adaptation exploits the property that learner errors are systematic. Indeed, when priors are estimated on the test data (in 5-fold cross-validation), the performance improves, e.g. the preposition module attains an F1 of 18.05 instead of 12.14.

Concerning lack of improvement on noun number errors, we hypothesize that these errors differ from the other mistakes in that the appropriate form strongly depends on the surface form of the noun, which would, in turn, suggest that the dependency of the label on the grammatical form of the source that the adaptation is trying to discover is weak. Indeed, the prior distribution of {singular, plural} label space does not change much when the source feature is taken into account. The unadapted priors for "singular" and "plural" are 0.75 and 0.25, respectively. Similarly, the adapted priors (singular|plural) and (plural|singular) are 0.034 and 0.016, respectively. In other words, the unadapted prior probability for "plural" is three times lower than for "singular", which does not change much with adaptation. This is different for other errors. For instance, in case of verb agreement, the unadapted prior for "plural" is 0.617, more than three times than the "singular" prior of 0.20. With adaptation, these priors become almost the same (0.016 and 0.012).

Adapting AP The AP is a discriminative learning algorithm and does not use priors on the set of candidates. In order to reflect our estimate of the error distribution, the AP algorithm is adapted differently, by introducing into the native data artificial errors, in a rate that reflects the errors made by the ESL writers (Rozovskaya and Roth, 2010b). The idea is to simulate learner errors in training, through artificial mistakes (also produced using an error confusion matrix).[9] The original method was proposed for models trained on native data. This technique can be further enhanced using *the error inflation method* (Rozovskaya et al., 2012, Sec. 6) applied to models trained on native or learner data.

The Illinois system uses error inflation in its article classifier. Because this classifier is trained on learner data, the *source* article can be used as a feature. However, since learner errors are sparse, the *source* feature encourages the model to abstain from flagging a mistake, which results in low recall. The error inflation technique addresses this problem by boosting the proportion of errors in the training data. It does this by generating additional artificial errors using the error distribution from the training set.

Table 9 shows the results of adapting the AP classifier using error inflation. (We omit noun results, since the noun AP model performs better without the source feature, which is similar to the noun NB model, as discussed above.) The inflation method improves recall and, consequently, F1. It should be noted that although inflation also decreases precision it is still helpful. In fact, because of the low error rates, performance on the CoNLL dataset with natural errors is very poor, often resulting in F1 being equal to 0 due to no errors being detected.

Inflation vs. Sampling To demonstrate the impact of error inflation, we compare it against sampling, an approach used by other teams – e.g. HIT – that improves recall by removing correct examples in training. The HIT article model is similar to the

[9]The idea of using artificial errors goes back to Izumi et al. (2003) and was also used in Foster and Andersen (2009). The approach discussed here refers to the *adaptation* method in Rozovskaya and Roth (2010b) that generates artificial errors using the distribution of naturally-occurring errors.

Error	Model	F1	
		CoNLL	FCE
Art.	AP (natural errors)	07.06	27.65
	AP (infl. const. 0.9)*	24.61	30.96
Prep.	AP (natural errors)	0.0	14.69
	AP (infl. const. 0.7)	07.37	34.77
Agr.	AP (natural errors)	0.0	08.05
	AP (infl. const. 0.8)	17.06	31.03
Form	AP (natural errors)	0.0	01.56
	AP (infl. const. 0.9)	10.53	09.43

Table 9: **Adapting AP using error inflation.** Models are trained on learner data with word n-gram features and the source feature. *Inflation constant* shows how many correct instances remain (e.g. 0.9 indicates that 90% of correct examples are unchanged, while 10% are converted to mistakes.) Modules that are part of the Illinois submission are marked with an asterisk.

Infl. constant	F1	
	Sampling	Inflation
0.90	23.22	24.61
0.85	27.75	29.29
0.80	30.04	33.47
0.70	33.02	35.52
0.60	32.78	35.03

Table 10: **Comparison of the inflation and sampling methods on article errors (CoNLL).** The proportion of errors in training in each row is identical.

Illinois model but scored three points below. Table 10 shows that sampling falls behind the inflation method, since it considerably reduces the training size to achieve similar error rates. The proportion of errors in training in each row is identical: sampling achieves the error rates by removing correct examples, whereas the inflation method converts some positive examples to artificial mistakes. *Inflation constant* shows how many correct instances remain; smaller inflation values correspond to more erroneous instances in training; the sampling approach, correspondingly, removes more positive examples.

To summarize, we have demonstrated the impact of error inflation by comparing it to a similar method used by another team; we have also shown that further improvements can be obtained by adapting NB to learner errors using the priors method, when training and test data exhibit similar error patterns.

5.3 Dim. 3: Linguistic Knowledge

The use of linguistic knowledge is important in several components of the error correction system: feature engineering, candidate identification, and special techniques for correcting verb errors.

Error	Features	F1	
		CoNLL	FCE
Art.	n-gram	24.61	30.96
	n-gram+POS+chunk*	**33.50**	**35.66**
Agr.	n-gram	17.06	31.03
	n-gram+POS	24.14	35.29
	n-gram+POS+syntax	**27.93**	**41.23**

Table 11: **Feature evaluation.** Models are trained on learner data, use the source word and error inflation. Modules that are part of the Illinois submission are marked with an asterisk.

Features It is known from many NLP tasks that feature engineering is important, and this is the case here. Note that this is relevant only when training on learner data, as models trained on Web1T can make use of n-gram features only but for the NUCLE corpus we have several layers of linguistic annotation.[10] We found that for article and agreement errors, using deeper linguistic knowledge is especially beneficial. The article features in the Illinois module, in addition to the surface form of the context, encode POS and shallow parse properties. These features are presented in Rozovskaya et al. (2013, Table 3) and Appendix Table A.19. The Illinois agreement module is trained on Web1T but further analysis reveals that it is better to train on learner data with rich features. The word n-gram and POS agreement features are the same as those in the article module. Syntactic features encode properties of the subject of the verb and are presented in Rozovskaya et al. (2014b, Table 7) and Appendix Table A.18; these are based on the syntactic parser (Klein and Manning, 2003) and the dependency converter (Marneffe et al., 2006).

Table 11 shows that adding rich features is helpful. Notably, adding deeper syntactic knowledge to the agreement module is useful, although parse features are likely to contain more noise.[11] Foster (2007) and Lee and Seneff (2008) observe a degrade in performance on syntactic parsers due to grammatical noise that also includes agreement errors. For articles, we chose to add syntactic knowledge from shallow parse as it is likely to be sufficient for articles and more accurate than full-parse features.

Candidate Identification for errors on open-class

[10] Feature engineering will also be relevant when training on a native corpus that has linguistic annotation.

[11] Parse features have also been found useful in preposition error correction (Tetreault et al., 2010).

words is rarely discussed but is a crucial step: it is not possible to identify the relevant candidates using a closed list of words, and the procedure needs to rely on pre-processing tools, whose performance on learner data is suboptimal.[12] Rozovskaya et al. (2014b, Sec. 5.1) describe and evaluate several candidate selection methods for verbs. The Illinois system implements their best method that addresses pre-processing errors, by selecting words tagged as verbs as well as words tagged as *NN*, whose lemma is on the list of valid verb lemmas (Sec. 4.3).

Following descriptions provided by several teams, we evaluate several candidate selection methods for nouns. The first method includes words tagged as *NN* or *NNS* that head an NP. NTHU and HIT use this method; NTHU obtained the second best noun score, after the Illinois system; its model is also trained on Web1T. The second method includes all words tagged as *NN* and *NNS* and is used in several other systems, e.g. SZEG, (Berend et al., 2013).

The above procedures suffer from pre-processing errors. The Illinois method addresses this problem by adding words that end in common noun suffixes, e.g. "ment", "ments", and "ist". The percentage of noun errors selected as candidates by each method and the impact of each method on the performance are shown in Table 12. The Illinois method has the best result on both datasets; on CoNLL, it improves F1 score by 2 points and recovers 43% of the candidates that are missed by the first approach. On FCE, the second method is able to recover more erroneous candidates, but it does not perform as well as the last method, possibly, due to the number of noisy candidates it generates. To conclude, pre-processing mistakes should be taken into consideration, when correcting errors, especially on open-class words.

Using Verb Finiteness to Correct Verb Errors As shown in Table 4, the surface realizations that correspond to the agreement candidates are a subset of the possible surface realizations of the form classifier. One natural approach, thus, is to train one classifier to predict the correct surface form of the verb. However, the same surface realization may correspond to multiple grammatical properties. This observation motivates the approach that corrects agreement and form errors separately (Rozovskaya et al., 2014b). It uses the linguistic notion of *verb finiteness* (Radford, 1988) that distinguishes between finite and non-finite verbs, each of which fulfill different grammatical functions and thus are marked for different grammatical properties.

[12] Candidate selection is also difficult for closed-class errors in the case of omissions, e.g. articles, but article errors have been studied rather extensively, e.g. (Han et al., 2006), and we have no room to elaborate on it here.

Candidate ident. method	**Error recall (%)**		**F1**	
	CoNLL	**FCE**	**CoNLL**	**FCE**
NP heads	87.72	92.32	40.47	34.16
All nouns	89.50	95.29	41.08	33.16
Nouns+heuristics*	92.84	94.86	**42.60**	**34.40**

Table 12: **Nouns: effect of candidate identification methods on the correction performance.** Models are trained using NB. *Error recall* denotes the percentage of nouns containing number errors that are selected as candidates. Modules that are part of the Illinois submission are marked with an asterisk.

Training method	**F1**	
	CoNLL	**FCE**
One classifier	16.43	21.14
Finiteness-based training (I)	18.59	27.72
Finiteness-based training (II)	**21.08**	**29.98**

Table 13: **Improvement due to separate training for verb errors.** Models are trained using the AP algorithm.

Verb finiteness is used to direct each verb to the appropriate classifier. The candidates for the agreement module are verbs that take agreement markers: the finite surface forms of the be-verbs ("is", "are", "was", and "were"), auxiliaries "have" and "has", and finite verbs tagged as *VB* and *VBZ* that have explicit subjects (identified with the parser). The form candidates are non-finite verbs and some of the verbs whose finiteness is ambiguous.

Table 13 compares the two approaches: when all verbs are handled together; and when verbs are processed separately. All of the classifiers use surface form and POS features of the words in the 4-word window around the verb. Several subsets of these features were tried; the single classifier uses the best combination, which is the same word and POS features shown in Appendix Table A.19. Finiteness-based classifier (I) uses the same features for agreement and form as the single classifier.

When training separately, we can also explore whether different errors benefit from different features; finiteness-based classifier (II) optimizes features for each classifier. The differences in the feature sets are minor and consist of removing several

unigram word and POS features of tokens that do not appear immediately next to the verb. Recall from the discussion on features that the agreement module can be further improved by adding syntactic knowledge. In the next section, it is shown that an even better approach is to train on learner data for agreement mistakes and on native data for form errors. The results in Table 13 are for AP models but similar improvements due to separate training are observed for NB models trained on Web1T. Note that the NTHU system also corrects all verb errors using a model trained on Web1T but handles all these errors together; its verb module scored 8 F1 points below the Illinois one. While there are other differences between the two systems, the results suggest that part of the improvement within the Illinois system is indeed due to handling the two errors separately.

5.4 Dim. 4: Training Data

NUCLE is a large corpus produced by learners of the same language background as the test data. Because of its large size, training on this corpus is a natural choice. Indeed, many teams follow this approach. On the other hand, an important issue in the CoNLL task is the difference between the training and test sets, which has impact on the selection of the training set – the large Web1T has more coverage and allows for better generalization. We show that for some errors it is especially advantageous to train on a larger corpus of native data. It should be noted that while we refer to the Web1T corpus as "native", it certainly contains data from language learners; we assume that the noise can be neglected.

Table 14 compares models trained on native and learner data in their best configurations based on the training data. Overall, we find that Web1T is clearly preferable for noun errors. We attribute this to the observation that noun number usage strongly depends on the surface form of the noun, and not just the contextual cues and syntactic structure. For example, certain nouns in English tend to be used exclusively in singular or plural form. Thus, considerably more data compared to other error types is required to learn model parameters.

On article and preposition errors, native-trained models perform slightly better on CoNLL, while learner-trained models are better on FCE. We conjecture that the FCE training set is more similar to the respective test data and thus provides an advantage over training on native data.

Error	Train. data	Learning algorithm	Features	F1	
				CoNLL	FCE
Art.	Native	NB-adapt.	n-gram	**34.49**	31.76
	Learner	AP-infl.*	+POS+chunk	33.50	**35.66**
Prep.	Native	LM; NB-adapt.	n-gram	**12.09**	32.22
	Learner	AP-infl.	n-gram	10.26	**33.93**
Noun	Native	NB*	n-gram	**42.60**	**32.38**
	Learner	AP-infl.	+POS	19.22	17.28
Agr.	Native	NB-adapt.	n-gram	23.46	38.57
	Learner	AP-infl.	+POS+syntax	**27.93**	**41.23**
Form	Native	NB-adapt.	n-gram	**18.35**	**16.67**
	Learner	AP-infl.	+POS	12.32	12.02

Table 14: **Choice of training data: learner vs. native (Web1T).** For prepositions, LM is chosen for CoNLL, and NB-adapted for FCE. Modules that are part of the Illinois submission are marked with an asterisk.

On verb agreement errors, native-trained models perform better than those trained on learner data, when the same n-gram features are used. However, when we add POS and syntactic knowledge, training on learner data is advantageous. Finally, for verb form errors, there is an advantage when training on a lot of native data, although the difference is not as substantial as for noun errors. This suggests that unlike agreement mistakes that are better addressed using syntax, form errors, similarly to nouns, benefit from training on a lot of data with n-gram features.

To summarize, choice of the training data is an important consideration for building a robust system. Researchers compared native- and learner-trained models for prepositions (Han et al., 2010; Cahill et al., 2013), while the analysis in this work addresses five error types – showing that errors behave differently – and evaluates on two corpora.[13]

6 Discussion

In Table 15, we show the results of the system, where the best modules are selected based on the performance on the training data. We also show the Illinois modules (without post-processing). The following changes are made with respect to the Illinois submission: the preposition system is based on an LM and enhanced to handle spurious preposition errors (thus the Illinois result of 7.10 shown here is

[13]For studies that directly combine native and learner data in training, see Gamon (2010) and Dahlmeier and Ng (2011).

Error	Illinois submission		This work	
	Model	F1	Model	F1
Art.	AP-infl.	33.50	AP-infl.	33.50
Prep.	NB-adapt.	07.10	LM	**12.09**
Noun	NB	42.60	NB	42.60
Agr.	NB	26.14	AP-infl.	**27.93**
Form	NB	14.50	NB-adapt.	**18.35**
All		31.43		**31.75**

Table 15: **Results on CoNLL of the Illinois system (without post-processing) and this work.** NB and LM models are trained on Web1T; AP models are trained on NUCLE. Modules different from the Illinois submission are in bold.

different from the 12.14 in Table 8); the agreement classifier is trained on the learner data using AP with rich features and error inflation; the form classifier is adapted to learner mistakes, whereas the Illinois submission trains NB without adaptation. The key improvements are observed with respect to least frequent errors, so the overall improvement is small. Importantly, the Illinois system already takes into account the four dimensions analyzed in this paper.

In CoNLL-2013, systems were compared using F1. Practical systems, however, should be tuned for good precision to guarantee that the overall quality of the text does not go down. Clearly, optimizing for F1 does not ensure that the system improves the quality of the text (see Appendix B). A different evaluation metric based on the *accuracy* of the data is proposed in Rozovskaya and Roth (2010b). For further discussion of evaluation metrics, see also Wagner (2012) and Chodorow et al. (2012).

It is also worth noting that the obtained results underestimate the performance because the agreement on what constitutes a mistake can be quite low (Madnani et al., 2011), so providing alternative corrections is important. The revised annotations address this problem. The Illinois system improves its F1 from 31.20 to 42.14 on revised annotations. However, these numbers are still an underestimation because the analysis typically eliminates precision errors but not recall errors. This is not specific to CoNLL: an error analysis of the false positives in CLC that includes the FCE showed an increase in precision from 33% to 85% and 33% to 75% for preposition and article errors (Gamon, 2010).

An error analysis of the training data also allows us to determine prominent groups of system errors and identify areas for potential improvement, which we outline below. **Cascading NLP errors:** In the example below, the Illinois system incorrectly changes "need" to "needs" as it considers "victim" to be the subject of that verb: "*Also, not only the kidnappers and the victim **needs** to be tracked down, but also jailbreakers.*" **Errors in interacting linguistic structures:** The Illinois system considers every word independently and thus cannot handle interacting phenomena. In the example below, the article and the noun number classifiers propose corrections that result in an ungrammatical structure "such a situations": "*In **such situation**, individuals will lose their basic privacy.*" This problem is addressed via global models (Rozovskaya and Roth, 2013) and results in an improvement over the Illinois system. **Errors due to limited context:** The Illinois system does not consider context beyond sentence level. In the example below, the system incorrectly proposes to delete "the" but the wider context indicates that the definite article is more appropriate here: "*We have to admit that how to prevent **the** abuse and how to use it reasonably depend on a sound legal system, and it means surveillance has its own restriction.*"

7 Conclusion

We identified key design principles in developing a state-of-the-art error correction system. We did this through analysis of the top system in the CoNLL-2013 shared task along several dimensions. The key dimensions that we identified and analyzed concern the choice of a learning algorithm, adaptation to learner mistakes, linguistic knowledge, and the choice of the training data. We showed that the decisions in each case depend both on the type of a mistake and the specific setting, e.g. how much annotated learner data is available. Furthermore, we provided points of comparison with other systems along these four dimensions.

Acknowledgments

We thank Peter Chew and the anonymous reviewers for the feedback. Most of this work was done while the first author was at the University of Illinois. This material is based on research sponsored by DARPA under agreement number FA8750-13-2-0008 and by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the view of the agencies.

References

G. Berend, V. Vincze, S. Zarrieß, and R. Farkas. 2013. Lfg-based features for noun number and article grammatical errors. In *Proceedings of CoNLL: Shared Task*.

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

A. Cahill, N. Madnani, J. Tetreault, and D. Napolitano. 2013. Robust systems for preposition error correction using wikipedia revisions. In *Proceedings of NAACL*.

S. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of ACL*.

M. Chodorow, M. Dickinson, R. Israel, and J. Tetreault. 2012. Problems in evaluating grammatical error detection systems. In *Proceedings of COLING*.

D. Dahlmeier and H. T. Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of ACL*.

D. Dahlmeier and H.T Ng. 2012. A beam-search decoder for grammatical error correction. In *Proceedings of EMNLP-CoNLL*.

D. Dahlmeier, H.T. Ng, and S.M. Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

R. Dale and A. Kilgarriff. 2011. Helping Our Own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*.

R. Dale, I. Anisimoff, and G. Narroway. 2012. A report on the preposition and determiner error correction shared task. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Y. Even-Zohar and D. Roth. 2001. A sequential model for multi class classification. In *Proceedings of EMNLP*.

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*.

J. Foster. 2007. Treebanks gone bad: Generating a treebank of ungrammatical english. In *Proceedings of the IJCAI Workshop on Analytics for Noisy Unstructures Data*.

Y. Freund and R. E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*.

M. Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *Proceedings of NAACL*.

S. Gass and L. Selinker. 1992. *Language transfer in language learning*. John Benjamins.

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 *Proceedings of LREC*.

T. Ionin, M.L. Zubizarreta, and S. Bautista. 2008. Sources of linguistic knowledge in the second language acquisition of English articles. *Lingua*, 118:554–576.

E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *Proceedings of ACL*.

T.-H. Kao, Y.-W. Chang, H.-W. Chiu, T-.H. Yen, J. Boisson, J.-C. Wu, and J.S. Chang. 2013. CoNLL-2013 shared task: Grammatical error correction NTHU system description. In *Proceedings of CoNLL: Shared Task*.

D. Klein and C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Proceedings of NIPS*.

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. Correcting misuse of verb forms. In *Proceedings of ACL*.

N. Madnani, M. Chodorow, J. Tetreault, and A. Rozovskaya. 2011. They can help: Using crowdsourcing to improve the evaluation of grammatical error detection systems. In *Proceedings of ACL*.

M. Marneffe, B. MacCartney, and Ch. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.

H. T. Ng, S. M. Wu, Y. Wu, Ch. Hadiwinoto, and J. Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

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

A. Radford. 1988. *Transformational Grammar*. Cambridge University Press.

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

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

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

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

A. Rozovskaya and D. Roth. 2013. Joint learning and inference for grammatical error correction. In *Proceedings of EMNLP*.

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

A. Rozovskaya, M. Sammons, and D. Roth. 2012. The UI system in the HOO 2012 shared task on error correction. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

A. Rozovskaya, K.-W. Chang, M. Sammons, and D. Roth. 2013. The University of Illinois system in the CoNLL-2013 shared task. In *Proceedings of CoNLL Shared Task*.

A. Rozovskaya, K.-W. Chang, M. Sammons, D. Roth, and N. Habash. 2014a. The University of Illinois and Columbia system in the CoNLL-2014 shared task. In *Proceedings of CoNLL Shared Task*.

A. Rozovskaya, D. Roth, and V. Srikumar. 2014b. Correcting grammatical verb errors. In *Proceedings of EACL*.

A. Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

J. Tetreault, J. Foster, and M. Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of ACL*.

J. Wagner. 2012. *Detecting Grammatical Errors with Treebank-Induced, Probabilistic Parsers*. Ph.D. thesis.

Y. Xiang, B. Yuan, Y. Zhang, X. Wang, W. Zheng, and C. Wei. 2013. A hybrid model for grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

J. Xing, L. Wang, D.F. Wong, L.S. Chao, and X. Zeng. 2013. UM-Checker: A hybrid system for English grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

H. Yannakoudakis, T. Briscoe, and B. Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of ACL*.

I. Yoshimoto, T. Kose, K. Mitsuzawa, K. Sakaguchi, T. Mizumoto, Y. Hayashibe, M. Komachi, and Y. Matsumoto. 2013. NAIST at 2013 CoNLL grammatical error correction shared task. In *Proceedings of CoNLL: Shared Task*.

Appendix A Features and Additional Information about the Data

	Classifier				
	Art.	**Prep.**	**Noun**	**Agr.**	**Form**
Train	43K	20K	39K	22K	37K
Test	43K	20K	39K	22K	37K

Table A.16: **Number of candidate words by classifier type in training and test data (FCE).**

Error	**Number of errors and error rate**	
	Train	**Test**
Art.	2336 (5.4%)	2290 (5.3%)
Prep.	1263 (6.4%)	1205 (6.1%)
Noun	858 (2.2%)	805 (2.0%)
Verb agr.	319 (1.5%)	330 (1.4%)
Verb form	104 (0.3%)	127 (0.3%)

Table A.17: **Statistics on annotated errors in the FCE corpus.** Percentage denotes the error rates, i.e. the number of erroneous instances with respect to the total number of relevant instances in the data.

	Features	**Description**
(1)	subjHead, subjPOS	the surface form and the POS tag of the subject head
(2)	subjDet	determiner of the subject NP
(3)	subjDistance	distance between the verb and the subject head
(4)	subjNumber	*Sing* – singular pronouns and nouns; *Pl* – plural pronouns and nouns
(5)	subjPerson	*3rdSing* – "she", "he", "it", singular nouns; *Not3rdSing* – "we", "you", "they", plural nouns; *1stSing* – "I"
(6)	conjunctions	(1)&(3); (4)&(5)

Table A.18: **Verb agreement features that use syntactic knowledge.**

Appendix B Evaluation Metrics

Here, we discuss the CoNLL-2013 shared task evaluation metric and provide a little bit more detail on

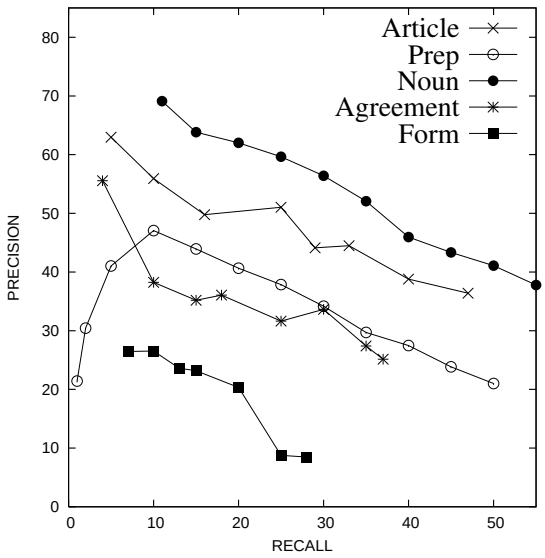

Figure 2: **Precision/Recall curves by error type.**

the performance of the Illinois modules in this context. As shown in Table 1 in Sec. 2, over 90% of words (about 98% in training) are used correctly. The low error rates are the key reason the error correction task is so difficult: it is quite challenging for a system to improve over a writer that already performs at the level of over 90%. Indeed, very few NLP tasks already have systems that perform at that level. The error sparsity makes it very challenging to identify mistakes accurately. In fact, the highest precision of 46.45%, as calculated by the shared task evaluation metric, is achieved by the Illinois system. However, once the precision drops below 50%, the system introduces more mistakes than it identifies.

We can look at individual modules and see whether for any type of mistake the system improves the quality of the text. Fig. 2 shows Precision/Recall curves for the system in Table 15. It is interesting to note that performance varies widely by error type. The easiest are noun and article usage errors: for nouns, we can do pretty well at the recall point 20% (with the corresponding precision of over 60%); for articles, the precision is around 50% at the recall value of 20%. For agreement errors, we can get a precision of 55% with a very high threshold (identifying only 5% of mistakes). Finally, on two mistakes – preposition and verb form – the system never achieves a precision over 50%.

Feature type	**Feature group**	**Features**
Word n-gram		*wB*, w_2B, w_3B, *wA*, w_2A, w_3A, *wBwA*, w_2BwB, wAw_2A, w_3Bw_2BwB, w_2BwBwA, *wBwA*w_2A, *wA*$w_2A w_3A$, $w_4Bw_3Bw_2BwB$, w_3Bw_2BwBwA, $w_2BwBwAw_2A$, *wBwA*$w_2A w_3A$, *wA*$w_2A w_3 w_4A$
POS		*pB*, p_2B, p_3B, *pA*, p_2A, p_3A, *pBpA*, p_2BpB, *pA*p_2A, *pBwB*, *pAwA*, p_2Bw_2B, p_2Aw_2A, p_2BpBpA, *pBpA*p_2A, *pA*p_2Ap_3A
Chunk	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 A.19: **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.