Correcting Grammatical Verb Errors

Alla Rozovskaya
Columbia University
New York, NY 10115
ar3366@columbia.edu

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

Vivek Srikumar
Stanford University
Stanford, CA 94305
svivek@cs.stanford.edu

Abstract

Verb errors are some of the most common mistakes made by non-native writers of English but some of the least studied. The reason is that dealing with verb errors requires a new paradigm; essentially all research done on correcting grammatical errors assumes a closed set of triggers – e.g., correcting the use of prepositions or articles – but identifying mistakes in verbs necessitates identifying potentially ambiguous triggers first, and then determining the type of mistake made and correcting it. Moreover, once the verb is identified, modeling verb errors is challenging because verbs fulfill many grammatical functions, resulting in a variety of mistakes. Consequently, the little earlier work done on verb errors assumed that the error type is known in advance.

We propose a linguistically-motivated approach to verb error correction that makes use of the notion of *verb finiteness* to identify triggers and types of mistakes, before using a statistical machine learning approach to correct these mistakes. We show that the linguistically-informed model significantly improves the accuracy of the verb correction approach.

1 Introduction

We address the problem of correcting grammatical verb mistakes made by English as a Second Language (ESL) learners. Recent work in ESL error correction has focused on errors in article and preposition usage (Han et al., 2006; Felice and Pulman, 2008; Gamon et al., 2008; Tetreault et al., 2010; Gamon, 2010; Rozovskaya and Roth, 2010b; Dahlmeier and Ng, 2011).

While verb errors occur as often as article and preposition mistakes, with a few exceptions (Lee and Seneff, 2008; Gamon et al., 2009; Tajiri et al., 2012), there has been little work on verbs. There are two reasons for why it is difficult to deal with verb mistakes. First, in contrast to articles and prepositions, verbs are more difficult to identify in text, as they can often be confused with other parts of speech, and processing tools are known to make more errors on noisy ESL data (Nagata et al., 2011). Second, verbs are more complex linguistically: they fulfill several grammatical functions, and these different roles imply different types of errors.

These difficulties have led all previous work on verb mistakes to assume prior knowledge of the mistake type; however, identifying the specific category of a verb error is nontrivial, since the surface form of the verb may be ambiguous, especially when that verb is used incorrectly. Consider the following examples of verb mistakes:

1. "We *discusses*/discuss* this every time."
2. "I will be lucky if I {*will find*}*/*find* something that fits."
3. "They wanted to visit many places without *spend*/spending* a lot of money."
4. "They arrived early to organized*/organize everything".

These examples illustrate three grammatical verb properties: *Agreement*, *Tense*, and non-finite *Form* choice that encompass the most common grammatical verb problems for ESL learners. The first two examples show mistakes on verbs that function as main verbs in a clause: sentence (1) shows an example of subject-verb *Agreement* error; (2) is an example of a *Tense* mistake where the ambiguity is between {*will find*} (Future tense)

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 358–367,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

and *find* (Present tense). Examples (3) and (4) display *Form* mistakes: confusing the infinitive and gerund forms in (3) and including an inflection on an infinitive verb in (4).

This paper addresses the specific challenges of verb error correction that have not been addressed previously – identifying candidates for mistakes and determining which class of errors is present, before proceeding to correct the error. The experimental results show that our linguistically-motivated approach benefits verb error correction. In particular, in order to determine the error type, we build on the notion of *verb finiteness* to distinguish between *finite* and *non-finite* verbs (Quirk et al., 1985), that correspond to *Agreement* and *Tense* mistakes (examples (1) and (2) above) and *Form* mistakes (examples (3) and (4) above), respectively (see Sec. 3). The approach presented in this work was evaluated empirically and competitively in the context of the CoNLL shared task on error correction (Ng et al., 2013) where it was implemented as part of the highest-scoring University of Illinois system (Rozovskaya et al., 2013) and demonstrated superior performance on the verb error correction sub-task.

This paper makes the following contributions:

- We present a holistic, linguistically-motivated framework for correcting grammatical verb mistakes; our approach "starts from scratch" without any knowledge of which mistakes should be corrected or of the mistake type; in doing that we show that the specific challenges of verb error correction are better addressed by first identifying the *finiteness* of the verb in the error identification stage.
- Within the proposed model, we describe and evaluate several methods of *selecting verb candidates*, an algorithm for *determining the verb type*, and a type-driven *verb error correction system.*
- We annotate a subset of the FCE data set with gold verb candidates and gold verb type.[1]

2 Related Work

Earlier work in ESL error correction follows the methodology of the *context-sensitive spelling correction* task (Golding and Roth, 1996; Golding and Roth, 1999; Banko and Brill, 2001; Carlson et al., 2001; Carlson and Fette, 2007). Most of the effort in ESL error correction so far has been on article and preposition usage errors, as these are some of the most common mistakes among non-native English speakers (Dalgish, 1985; Leacock et al., 2010). These phenomena are generally modeled as multiclass classification problems: a single classifier is trained for a given error type where the set of classes includes all articles or the top *n* most frequent English prepositions (Izumi et al., 2003; Han et al., 2006; Felice and Pulman, 2008; Gamon et al., 2008; Tetreault et al., 2010; Rozovskaya and Roth, 2010b; Rozovskaya and Roth, 2011; Dahlmeier and Ng, 2011).

Mistakes on verbs have attracted significantly less attention in the error correction literature. Moreover, the little earlier work done on verb errors only considered subsets of these errors and assumed the error sub-type is known in advance. Gamon et al. (2009) mentioned a model for learning gerund/infinitive confusions and auxiliary verb presence/choice. Lee and Seneff (2008) proposed an approach based on pattern matching on trees combined with word n-gram counts for correcting agreement misuse and some types of verb form errors. However, they excluded tense mistakes, which is the most common error category for ESL learners (40% of all verb errors, Sec. 3). Tajiri et al. (2012) considered only tense mistakes. In the above studies, it was assumed that the type of mistake that needs to be corrected is known, and irrelevant verb errors were excluded (e.g., Tajiri et al. (2012) addressed only tense mistakes and excluded from the evaluation other kinds of verb errors). In other words, it was assumed that part of the task was solved. But, unlike in article and preposition error correction where the type of mistake is known based on the surface form of the word, in verb error correction, it is not obvious.

The key distinction of our work is that we propose a holistic approach that starts from "scratch" and, given an instance, first detects a mistake and identifies its type, and then proceeds to correct it. We also evaluate several methods for selecting verb candidates and show the significance of this step for improving verb error correction performance, while earlier studies do not discuss this aspect of the problem. In the CoNLL shared task (Ng et al., 2013) that included verb errors in agreement and form, the participating teams did not provide details on how specific challenges were handled, but the University of Illinois system obtained the highest score on the verb sub-task, even though

[1] The annotation is available at http://cogcomp.cs.illinois.edu/page/publication_view/743

Tag	Error type	Rel. freq. (%)
TV	Tense	40.0
FV	Form	22.3
AGV	Verb-subject agreement	11.5
MV	Missing verb	11.7
UV	Unneccesary verb	7.3
IV	Inflection	5.4
DV	Derivation	1.8
Total		6640

Table 1: **Grammatical verb errors in FCE.**

all teams used similar resources (Ng et al., 2013).

3 Verb Errors in ESL Writing

Verb-related errors are very prominent among non-native English speakers: grammatical misuse of verbs constitutes one of the most common errors in several learner corpora, including those previously used (Izumi et al., 2003; Lee and Seneff, 2008) and the one employed in this work. We study verb errors using the FCE corpus (Yannakoudakis et al., 2011). The corpus possesses several desirable characteristics: it is large (500,000 words), has been annotated by native English speakers, and contains data by learners of multiple first-language backgrounds. The FCE corpus contains 5056 determiner errors, 5347 preposition errors, and 6640 grammatical verb mistakes (Table 1).

3.1 Verb Finiteness

There are many grammatical categories for which English verbs can be marked. The linguistic notion of *verb finiteness* or *verb type* (Radford, 1988; Quirk et al., 1985) distinguishes between verbs that function on their own in a clause as main verbs (finite) and those that do not (non-finite). Grammatical properties associated with each group are mutually exclusive: tense and agreement markers, for example, do not apply to non-finite verbs; non-finite verbs are not marked for many grammatical functions but may appear in several forms.

The most common verb problems for ESL learners – *Tense*, *Agreement*, non-finite *Form* – involve verbs both in finite and non-finite roles. Table 2 illustrates contexts that license finite and non-finite verbs.

Our intuition is that, because properties associated with each verb type are mutually exclusive, verb finiteness should benefit verb error correction models: an observed verb error may be due to several grammatical phenomena, and knowing which phenomena are active depends on the function of the verb in the current context. Note that *Agreement*, *Tense*, and *Form* errors account for about 74% of all grammatical verb errors in Table 1 but the finiteness distinction applies to all English verbs – every verb is either finite or non-finite in a specific syntactic context – and is also relevant for the remaining mistakes not addressed here.[2]

Category	Agreement	Kappa	Random
Correct verbs	0.97	0.95	0.51
Erroneous verbs	0.88	0.81	0.41

Table 3: **Inter-annotator agreement** based on 250 verb errors and 250 correct verbs, randomly selected.

4 Annotation for Verb Finiteness

In order to evaluate the quality of the *algorithm for verb finiteness* and of the *candidate selection* methods, we annotated all verbs – correct and erroneous – in a random set of 124 documents from our corpus with the information about verb finiteness. We refer to these 124 documents as *gold subset*. We also annotated erroneous verbs in the remaining 1120 documents of the corpus. The annotation was performed by two students with background in Linguistics. The inter-annotator agreement is shown in Table 3 and is high.

Annotating Verb Errors For each verb error that was tagged as *Tense* (TV), *Agreement* (AGV), and *Form* (FV), the annotators marked verb finiteness. Additionally, the annotators also specified the type of error (*Tense*, *Agreement*, or *Form*) (Table 4), since the FCE tags do not always correspond to the three error types we study here. For example, the FV tag may mark errors on finite verbs. Overall, about 7% of verb errors have to do with phenomena different from the three verb properties considered in this work and thus are excluded from the present study.

Annotating Correct Verbs Correct verbs were identified in text using an automated procedure that relies on part-of-speech information (Sec. 5.1). Valid candidates were specified for verb finiteness. The candidates that were identified incorrectly due to mistakes by the part-of-speech tagger were marked as invalid.

5 The Computational Model

The verb error correction problem is formulated as a classification task in the spirit of the learn-

[2] For instance, the missing verb errors (MV, 11.7%) require an additional step to identify contexts for missing verbs, and then appropriate verb properties need to be determined based on verb finiteness.

Verb type	Example	Verb properties		
		Agreement	Tense	Form
Finite	“He _discussed_ this with me last week”	-	Past Simple	-
	“He _discusses_ this with me every week.”	3rd person,Sing.	Present Simple	-
Non-finite	“He left without _discussing_ it with me.”	-	-	Gerund
	“They let him _discuss_ this with me.”	-	-	Infinitive
	“_To discuss_ this now would be ill-advised.”	-	-	to-Infinitive

Table 2: **Contexts that license finite and non-finite verbs and the corresponding active properties.**

Error on Verb Type	Subcategory	Example
Finite (67.7%)	Agreement (20%)	“We _discusses*/discuss_ this every time.”
	Tense (80%)	“If you buy something, you {_would be_}*/{_will be_} happy.”
Non-finite (25.3%)		“If one is famous he has to accept the disadvantages of _be*/being_ famous.” “I am very glad {_for receiving_}*/{_to receive_} it.” “They arrived early to _organized*/organize_ everything.”
Other errors (7.0%)	Passive/Active(42.3%)	“Our end-of-conference party {_is included_}*_/includes_ dinner and dancing.”
	Compound (40.7%)	“You ask me for some _informations*/information_- here _they*/it are*/is_.”
	Other (16.8%)	“Nobody {_has to be_}*/{_should be_} late.”

Table 4: **Verb error classification** based on 4864 mistakes marked as TV, AGV, and FV errors in the FCE corpus.

ing paradigm commonly used for correcting other ESL errors (Sec. 2), with the exception that the verb model includes additional components. All of the components are listed below:

1. Candidate selection (5.1)
2. Verb finiteness prediction (5.2)
3. Feature generation (5.3)
4. Error identification (5.4)
5. Error correction (5.5)

After verb candidates are selected, verb finiteness is determined and features are generated for each candidate. The *finiteness* prediction is used in the *error identification* component. Given the output of the error identification stage, the corresponding classifiers for each error type are invoked to propose an appropriate correction.

We split the corpus documents into two equal parts – training and test. We chose a train-test split and not cross-validation, since the FCE data set is quite large to allow for such a split. The training data is also used to develop the components for candidate selection and verb finiteness prediction.

5.1 Candidate Selection

This stage selects the set of verb instances that are presented as input to the classifier. A *verb instance* refers to the verb, including its auxiliaries or the infinitive marker (e.g. “found”, “will find”, “to find”). Candidate selection is a crucial step for models that correct mistakes on open-class words because those errors that are missed at this stage have no chance of being detected. We implement four candidate selection methods. Method (1) extracts all verbs heading a verb phrase, as identified by a shallow parser (Punyakanok and Roth, 2001).[3] Method (2) also includes words tagged with one of the verb tags: {VB, VBN, VBG, VBD, VBP, VBZ} predicted by the POS tagger.[4] However, relying on the POS information is not good enough, since the POS tagger performance on ESL data is known to be suboptimal (Nagata et al., 2011). For example, verbs lacking agreement markers are likely to be mistagged as nouns (Lee and Seneff, 2008). Methods (3) and (4) address the problem of pre-processing errors. Method (3) adds words that are on the list of valid English verb lemmas; the lemma list is constructed using a POS-tagged version of the NYT section of the Gigaword corpus and contains about 2,600 of frequently-occurring words tagged as *VB*; for example, (3) will add *shop* but not *shopping*, but (4) will add both.

For methods (3) and (4), we developed *verbMorph*,[5] a tool that performs morphological analysis on verbs and is used to lemmatize verbs and to generate morphological variants. The module makes uses of (1) the verb lemma list and (2) a list of irregular English verbs.

The quality of the candidate selection methods is evaluated in Table 5 on the gold subset by computing the recall, i.e. the percentage of erroneous verbs that have been selected as candidates. Methods that address pre-processing mistakes are able to recover more erroneous verb candidates in text. It is also interesting to note that across all methods, the highest recall is obtained for tense errors. This suggests that the POS tagger is more prone to fail-

[3] http://cogcomp.cs.illinois.edu/demo/shallowparse

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

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

Method	Recall	Recall by error group (%)		
	(%)	Agr.	Tense	Form
(1) All verb phrases	83.00	86.62	93.55	59.08
(2) + tokens tagged as verbs	91.96	90.30	94.33	87.79
(3) + tokens that are valid verb lemmas	95.50	95.99	96.46	93.23
(4) + tokens with inflections that are valid verb lemmas	96.09	96.32	96.62	94.84

Table 5: **Candidate selection methods performance.**

ure due to errors in agreement and form. The evaluation in Table 5 uses recall, as the goal is to assess the ability of the methods to select erroneous verbs as candidates. In Sec. 6.1, the contribution of each method to error identification is evaluated.

5.2 Predicting Verb Finiteness

Predicting verb finiteness is not trivial, as almost all English verbs can occur in both finite and non-finite form and the surface forms of a verb in finite and non-finite form may be the same (see Table 2).

While we cannot learn verb type automatically due to lack of annotation, we show, however, that, for the majority of verbs, finiteness can be reliably predicted using linguistic knowledge. We implement a decision-list classifier that makes use of linguistically-motivated rules (Table 6). The algorithm covers about 92% of all verb candidates, abstaining on the remaining highly-ambiguous 8%.

The evaluation of the method on the gold subset (last column in Table 6) shows that despite its simplicity, this method is highly effective: 98% on correct verbs and over 89% on errors.

5.3 Features

The *baseline* features are word n-grams in the 4-word window around the verb instance. Additional features are intended to characterize a given error type and are selected based on previous studies: for *Agreement* and *Form* errors, we use a parser (Klein and Manning, 2003) and define features that reflect dependency relations between the verb and its neighbors. We denote these features by *syntax*. Syntactic knowledge via tree patterns has been shown useful for *Agreement* mistakes (Lee and Seneff, 2008). Features for *Tense* include temporal adverbs in the sentence and tenses of other verbs in the sentence and are similar to the features used in other verb classification tasks (Reichart and Rappoport, 2010; Lee, 2011; Tajiri et al., 2012). The features are shown in Table 7.

5.4 Error Identification

The goal of this stage is to identify errors and to predict their type. We define a linear model where, given a verb, a weight vector $\mathbf{w}$ assigns a score to each label in the label space $\{$Correct, Form, Agreement, Tense$\}$. The prediction of the classifier is the label with the highest score.

The baseline error identification model, called *combined*, is agnostic to the type of the verb. In the *combined* model, for each verb v and label l, we generate a feature vector, $\phi(v, l)$ and the best label is predicted as

$$\arg\max_{l} \mathbf{w}^T \phi(v, l).$$

The *combined* model makes use of *all* the features we have defined earlier for each verb.

The *type-based* model uses the verb finiteness prediction made by the verb finiteness classifier. A *soft* way to use the finiteness prediction is to add the predicted finiteness value as a feature. The other – *hard*-decision approach – is to use only a subset of the features depending on the predicted finiteness: *Agreement* and *Tense* for the finite verbs, and *Form* features for non-finite. The hard-decision *type-driven* approach defines a feature vector for a verb based on its type. Thus, given the verb v *and* its type t, we define features $\phi(v, t, l)$ for each label l. Thus, the label is predicted as

$$\arg\max_{l} \mathbf{w}^T \phi(v, t, l).$$

5.5 Error Correction

The correction module consists of three components, one for each type of mistake. Given the output of the error identification model, the appropriate correction component is run for each instance predicted to be a mistake.[6] The verb finiteness prediction is used to select finite instances for training the *Agreement* and *Tense* components and non-finite – for the *Form* component. The label space for *Tense* specifies tense and aspect properties of the English verbs (see Tajiri et al., 2012 for more detail), the *Agreement* component specifies the person and number properties, while the *Form* component includes the commonly confusable non-finite English forms (see Table 2). These components are trained as multiclass classifiers.

[6] We assume that each verb contains at most one mistake. Less than 1% of all erroneous verbs have more than one error present.

A verb is Non-Finite if any of the following hold:	A verb is Finite if any of the following hold	Accuracy on	
		Correct verbs	Erroneous verbs
(1) $[numTokens = 2] \wedge [firstToken = to]$ (2) $firstToken = be$ (3) $[numTokens = 1] \wedge [pos = VBG]$	(1) All verbs identified by shallow parser (2) *can*; *could* (3) $[numTokens = 1] \wedge [pos \in \{VBD, VBP, VBZ\}]$ (4) $[numTokens = 2] \wedge [firstToken! = to]$ (5) $numTokens > 2$	98.01	89.4

Table 6: **Algorithm for determining verb type.** *numTokens* denotes the number of tokens in the verb instance, e.g., for the verb instance "to go", $numTokens = 2$. Verbs not covered by the rules, e.g. those that are not tagged with a verb-related POS in methods (3) and (4), are not assigned any verb type. The last column shows algorithm accuracy on the gold subset separately for correct and incorrect verbs.

	Agreement	**Description**
(1)	subjHead, subjPOS	The surface form and the POS tag of the subject head
(2)	subjDet {those,this,..}	Determiner of the subject phrase
(3)	subjDistance	Distance between the verb and the subject head
(4)	subjNumber {*Sing, Pl*}	*Sing* – singular pronouns and nouns; *Pl* – plural pronouns and nouns
(5)	subjPerson {*3rdSing, Not3rdSing, 1stSing*}	*3rdSing* – she,he,it,singular nouns; *Not3rdSing* – we,you,they, plural nouns; *1stSing* – "I"
(6)	conjunctions	$(1)\&(3);(4)\&(5)$
	Tense	**Description**
(1)	verb phrase (VP)	verb lemma, negation, surface forms and POS tags of all words in the verb phrase
(2)	verbs in sentence(4 features)	tenses and lemmas of the finite verbs preceding and following the verb instance
(3)	time adverbs (2 features)	temporal adverb before and after the verb instance
(4)	bag-of-words (BOW) (8 features)	Includes the following words in the sentence: {if, when, since, then, wish, hope, when, since, after}
	Form	**Description**
(1)	closest word	surface form, lemma, POS tag, and distance of the closest open-class word to the left of the verb
(2)	governor	surface form, POS tag and dependency type of the target
(3)	preposition	if the verb is preceded by a preposition: preposition itself and the surface form, POS tag and dependency of the governor of the preposition
(4)	pos and lemma	POS tag and lemma of the verb and their conjunctions with features in (2) and (3) and word ngrams

Table 7: **Features used, grouped by error type.**

6 Experiments

The main goal of this work is to propose a unified framework for correcting verb mistakes and to address the specific challenges of the problem. We thus do not focus on features or on the specific learning algorithm. Our experimental study addresses the following research questions:

I. **Linguistic questions**: (i) candidate selection methods; (ii) verb finiteness contribution to error identification

II. **Computational Framework**: error identification vs. correction

III. **Gold annotation**: (i) using gold candidates and verb type vs. automatic; (ii) performance comparison by error type

Learning Framework There is a lot of understanding for which algorithmic methods work best for ESL correction tasks, how they compare among themselves, and how they compare to n-gram based methods. Specifically, despite their intuitive appeal, language models were shown to not work well on these tasks, while the discriminative learning framework has been shown to be superior to other approaches and thus is commonly used for error correction tasks (see Sec. 2). Since we do not address the algorithmic aspect of the problem, we refer the reader to Rozovskaya and Roth (2011) for a discussion of these issues. We train all our models with the SVM learning algorithm implemented in JLIS (Chang et al., 2010).

Evaluation We report both Precision/Recall curves and AAUC (as a summary). Error correction is generally evaluated using F1 (Dale et al., 2012); Precision and Recall (Gamon, 2010; Tajiri et al., 2012); or Average Area Under Curve (AAUC) (Rozovskaya and Roth, 2011). For a discussion on these metrics with respect to error correction tasks, we refer the reader to Rozovskaya (2013). AAUC (Hanley and McNeil, 1983)) is a measure commonly used to generate a summary statistic, computed as an average precision value over a range of recall points. In this paper, AAUC is computed over the first 15 recall points:

$$AAUC = \frac{1}{15} \cdot \sum_{i=1}^{15} Precision(i).$$

6.1 Linguistic Questions

Candidate Selection Methods The contribution of the candidate selection component with respect to error identification is evaluated in Table 8, using the methods presented in Sec. 5.1. Overall,

Recall of candidate	AAUC	
selection method (%)	Combined	Type-based
(1) (83.00)	73.38	79.49
(2) (91.96)	80.36	86.48
(3) (95.50)	**81.39**	**87.05**
(4) (96.09)	81.27	86.81

Table 8: **Impact of candidate selection methods on error identification performance.** The first column shows the percentage of erroneous verbs selected by each method. *Type-based* models are discussed in Sec. 6.1.

	Correct verbs	Erroneous verbs	Error rate
Training	41721	1981	4.75%
Test	41836	2014	4.81%

Table 9: **Training and test data statistics.** Candidates are selected using method (3).

better performance is achieved by methods with higher recall, with the exception of method (4); its performance on error identification is behind that of method (3), perhaps due to the amount of noise that is also added. While the difference is small, method (3) is also simpler than method (4). We thus use method (3) in the rest of the paper. Table 9 shows the number of verb instances in training and test selected with this method.

Verb Finiteness Sec. 5.4 presented two ways of adding verb finiteness: (1) adding the predicted verb type as a feature and (2) selecting only the relevant features depending on the finiteness of the verb. Table 10 shows the results of using verb type in the error identification stage. While the first approach does not provide improvement over the *combined* model, the second method is very effective. We conjecture that because verb type prediction is quite accurate, the second, hard-decision approach is preferred, as it provides knowledge in a direct way. Henceforth, we will use the second method in the *type-based* model.

Fig. 1 compares the performance of the *combined* and the hard-decision *type-based* models shown in Table 10. Precision/Recall curves are generated by varying the threshold on the confidence of the classifier. This graph reveals the behavior of the systems at multiple recall points: we observe that at every recall point the *type-based* classifier has higher precision.

So far, the models used all features defined in Sec. 5.3. Table 11 reveals that the type-driven approach is superior to the combined approach across different feature sets, and the performance gap increases with more sophisticated feature sets, which is to be expected, since more complex features are tailored toward relevant verb errors. Furthermore, adding features specific to each error type significantly improves the performance over the word n-gram features. The rest of the experiments use all features (denoted *Full* feature set).

Model	AAUC
Combined	81.39
Type-based I (soft)	81.11
Type-based II (hard)	**87.05**

Table 10: **Verb finiteness contribution to error identification.**

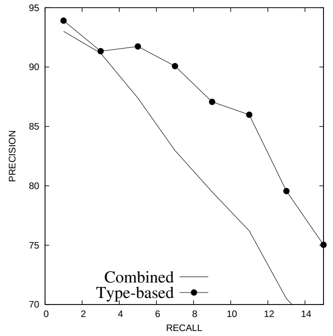

Figure 1: **Verb finiteness contribution to error identification: key result**. AAUC shown in Table 10. The *combined* model uses no verb type information. In the hard-decision *type-based* model, each verb uses the features according to its finiteness. The differences are statistically significant (McNemar's test, $p < 0.0001$).

Feature set	AAUC	
	Combined	Type-based
Baseline	46.62	49.72
All−Syntax	79.47	84.88
Full feature set	**81.39**	**87.05**

Table 11: **Verb finiteness contribution to error identification for different features.**

6.2 Identification vs. Correction

After running the error identification component, we apply the appropriate correction models to those instances identified as errors. The results for identification and correction are shown in Table 12. The correction models are also finiteness-aware models trained on the relevant verb instances (finite or non-finite), as predicted by the verb finiteness classifier.

We evaluate the correction components by fixing a recall point in the error identification stage.[7] We observe the relatively low recall obtained by the models. Error correction models tend to have low recall (see, for example, the recent shared tasks on ESL error correction (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013)). The key reason for the low recall is the error sparsity: over 95% of verbs are correct, as shown in Table 9.

[7] We can increase recall using a different threshold but higher precision is preferred in error correction tasks.

Error type	Correction			Identification		
	P	R	F1	P	R	F1
Agreement	90.62	9.70	17.52	90.62	9.70	17.52
Tense	60.51	7.47	13.31	86.62	10.70	19.05
Form	81.82	16.34	27.24	83.47	16.67	27.79
Total	71.94	10.24	17.94	85.81	12.22	21.20

Table 12: **Performance of the complete model after the correction stage.** The results on *Agreement* mistakes are the same, since *Agreement* errors are always binary decisions, unlike *Tense* and *Form* mistakes.

The only way to improve over this 95% baseline is by forcing the system to have very good precision (at the expense of recall). The performance shown in Table 12 corresponds to an accuracy of 95.60% in identification (**error reduction of 8.7%**) and 95.40% in correction (**error reduction of 4.5%**) over the baseline of 95.19%.

6.3 Analysis on Gold Data

To further study the impact of each step of the system, we analyze our model on the gold subset of the data. The gold subset contains two additional pieces of information not available for the rest of the corpus: gold verb candidates and gold verb finiteness (Sec. 4). The set contains 7784 gold verbs, including 464 errors. Experiments are run in 10-fold cross-validation where on each run 90% of the documents are used for training and the remaining 10% are used for evaluation. The gold annotation can be used instead of automatic predictions in two system components: (1) candidate selection and (2) verb finiteness.

Table 13 shows the performance on error identification when gold vs. automatic settings are used. As expected, using the gold verb type is more effective than using the automatic one, both with automatic and gold candidates. The same is true for candidate selection. For instance, the *combined* model improves by 14 AAUC points (from 55.90 to 69.86) with gold candidates. These results indicate that candidate selection is an important component of the verb error correction system.

Note that compared to the performance on the entire data set (Table 10), the performance of the models shown here that use automatic components is lower, since the training size is smaller. On the other hand, because of the smaller training size, the gain due to the type-based approach is larger on the gold subset (19 vs. 6 AAUC points).

Finally, in Table 14, we evaluate the contribution of verb finiteness to error identification by error type. While performance varies by error, it is clear that all errors benefit from verb typing.

Candidate selection	Verb type prediction	AAUC
Automatic	None	55.90
	Automatic	74.72
	Gold	**89.45**
Gold	None	69.86
	Automatic	90.89
	Gold	**96.42**

Table 13: **Gold subset: error identification with gold vs. automatic candidates and finiteness information**. Value *None* for verb type prediction denotes the *combined* model.

Error type	AAUC		
	Combined	Type-based Automatic	Type-based Gold
Agreement	86.80	88.43	**89.21**
Tense	18.07	25.62	**26.87**
Form	97.08	98.23	**98.36**

Table 14: **Gold subset: gold vs. automatic finiteness contribution to error identification by error type.**

7 Conclusion

Verb errors are commonly made by ESL writers but difficult to address due to to their diversity and the fact that identifying verbs in (noisy) text may itself be difficult. We develop a linguistically-inspired approach that first identifies verb candidates in noisy learner text and then makes use of *verb finiteness* to identify errors and characterize the type of mistake. This is important, since most errors made by non-native speakers cannot be identified by considering only closed classes (e.g., prepositions and articles). Our model integrates a statistical machine learning approach with a rule-based system that encodes linguistic knowledge to yield the first general correction approach to verb errors (that is, one that does not assume prior knowledge of which mistake was made). This work thus provides a first step in considering more general algorithmic paradigms for correcting grammatical errors and paves the way for developing models to address other “open-class” mistakes.

Acknowledgments

The authors thank Graeme Hirst, Julia Hockenmaier, Mark Sammons, and the anonymous reviewers for their helpful feedback. This work was done while the first and the third authors were 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

M. Banko and E. Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France, July.

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

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

M. Chang, V. Srikumar, D. Goldwasser, and D. Roth. 2010. Structured output learning with indirect supervision. In *Proc. of the International Conference on Machine Learning (ICML)*.

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

R. Dale and A. Kilgarriff. 2011. Helping Our Own: The HOO 2011 pilot shared task. In *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 *Proc. of the NAACL HLT 2012 Seventh Workshop on Innovative Use of NLP for Building Educational Applications*, Montreal, Canada, June. Association for Computational Linguistics.

G. Dalgish. 1985. Computer-assisted ESL research. *CALICO Journal*, 2(2).

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

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. In *Proceedings of IJCNLP*.

M. Gamon, C. Leacock, C. Brockett, W. B. Dolan, J. Gao, D. Belenko, and A. Klementiev. 2009. Using statistical techniques and web search to correct ESL errors. *CALICO Journal, Special Issue on Automatic Analysis of Learner Language*, 26(3):491–511.

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

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

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

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

J. Hanley and B. McNeil. 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843.

E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148, Sapporo, Japan, July.

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 the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 20–25, Sofia, Bulgaria, August. Association for Computational Linguistics.

D. Klein and C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 NIPS*, pages 3–10. MIT Press.

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 *ACL*, pages 174–182, Columbus, Ohio, June. Association for Computational Linguistics.

J. Lee. 2011. Verb tense generation. *Social and Behavioral Sciences*, 27:122–130.

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

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

Language Learning. Association for Computational Linguistics.

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

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, New York.

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

R. Reichart and A. Rappoport. 2010. Tense sense disambiguation: A new syntactic polysemy task. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 325–334, Cambridge, MA, October. Association for Computational Linguistics.

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 the NAACL-HLT*.

A. Rozovskaya and D. Roth. 2011. Algorithm selection and model adaptation for esl correction tasks. In *ACL*, Portland, Oregon, 6. Association for Computational Linguistics.

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

A. Rozovskaya. 2013. *Automated Methods for Text Correction*. Ph.D. thesis.

T. Tajiri, M. Komachi, and Y. Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea, July. Association for Computational Linguistics.

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

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