Grammatical Error Correction: Machine Translation and Classifiers

Alla Rozovskaya
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
alla@vt.edu

Dan Roth
Department of Computer Science
University of Illinois
Urbana, IL 61820
danr@illinois.edu

Abstract

We focus on two leading state-of-the-art approaches to grammatical error correction – machine learning classification and machine translation. Based on the comparative study of the two learning frameworks and through error analysis of the output of the state-of-the-art systems, we identify key strengths and weaknesses of each of these approaches and demonstrate their complementarity. In particular, the machine translation method learns from parallel data without requiring further linguistic input and is better at correcting complex mistakes. The classification approach possesses other desirable characteristics, such as the ability to easily generalize beyond what was seen in training, the ability to train without human-annotated data, and the flexibility to adjust knowledge sources for individual error types.

Based on this analysis, we develop an algorithmic approach that combines the strengths of both methods. We present several systems based on resources used in previous work with a relative improvement of over 20% (and 7.4 F score points) over the previous state-of-the-art.

1 Introduction

For the majority of English speakers today, English is not the first language. These writers make a variety of grammar and usage mistakes that are not addressed by standard proofing tools. Recently, there has been a spike in research on grammatical error correction (GEC), correcting writing mistakes made by learners of English as a Second Language, including four shared tasks: HOO (Dale and Kilgarriff, 2011; Dale et al., 2012) and CoNLL (Ng et al., 2013; Ng et al., 2014). These shared tasks facilitated progress on the problem within the framework of two leading methods – machine learning classification and statistical machine translation (MT).

System	**Method**	**Performance**		
		P	**R**	**F0.5**
CoNLL-2014 top 3	MT	41.62	21.40	35.01
CoNLL-2014 top 2	Classif.	41.78	24.88	36.79
CoNLL-2014 top 1	MT, rules	39.71	30.10	37.33
Susanto et al. (2014)	MT, classif.	53.55	19.14	39.39
Miz. & Mats. (2016)	MT	45.80	26.60	40.00
This work	MT, classif.	60.17	25.64	**47.40**

Table 1: **(Lack of) progress in GEC over the last few years**.

The top CoNLL system combined a rule-based module with MT (Felice et al., 2014). The second system that scored almost as highly used machine learning classification (Rozovskaya et al., 2014), and the third system used MT (Junczys-Dowmunt and Grundkiewicz, 2014). Furthermore, Susanto et al. (2014) showed that a combination of the two methods is beneficial, but the advantages of each method have not been fully exploited.

Despite success of various methods and the growing interest in the task, the key differences between the leading approaches have not been identified or made explicit, which could explain the lack of progress on the task. Table 1 shows existing state-of-the-art since CoNLL-2014. The top results are close, suggesting that several groups have competitive systems. Two improvements (of <3 points) were published since then (Susanto et al., 2014; Mizumoto and Matsumoto, 2016).

The purpose of this work is to gain a better understanding of the values offered by each method and to facilitate progress on the task, building on the advantages of each approach. Through better understanding of the methods, we exploit the strengths of each technique and, building on existing architecture, develop superior systems within each framework. Further combination of these systems yields even more significant improvements over existing state-of-the-art. We make the following contributions:

• We examine two state-of-the-art approaches to GEC and identify strengths and weaknesses of the respective learning frameworks.

• We perform an error analysis of the output of two state-of-the-art systems, and demonstrate how the methods differ with respect to the types of language misuse handled by each.

• We exploit the strengths of each framework: with classifiers, we explore the ability to learn from native data, i.e. without supervision, and the flexibility to adjust knowledge sources to specific error types; with MT, we leverage the ability to learn without further linguistic input and to better identify complex mistakes that cannot be easily defined in a classifier framework.

• As a result, we build several systems that combine the strengths of both frameworks and demonstrate substantial progress on the task. Specifically, the best system outperforms the previous best result by 7.4 F score points.

Section 2 describes related work. Section 3 presents error analysis. In Section 4, we develop classifier and MT systems that make use of the strengths of each framework. Section 5 shows how to combine the two approaches. Section 6 concludes.

2 Related Work

We first introduce the CoNLL-2014 shared task and briefly describe the state-of-the-art GEC systems in the competition and beyond. Next, an overview of the two leading methods is presented.

2.1 CoNLL-2014 shared task and approaches

CoNLL-2014 training data (henceforth CoNLL-train) is a corpus of learner essays (1.2M words) written by students at the National University of Singapore (Dahlmeier et al., 2013), corrected and error-tagged. The CoNLL-2013 test set was included in CoNLL-2014 and is used as development. Both the development and the test sets are also from the student population studying at the same University but annotated separately. We report results on the CoNLL-2014 test.

The annotation includes specifying the relevant correction as well as the information about each error type. The tagset consists of 28 categories. Table 2 illustrates the 11 most frequent errors in the development data; errors are marked with an asterisk, and ∅ denotes a missing word. The majority of these errors are related to grammar but also include *mechanical*, *collocation*, and other errors.

An F-based scorer, named M2, was used to score the systems (Dahlmeier and Ng, 2012). The metric in CoNLL-2014 was F0.5, i.e. weighing precision twice as much as recall. Two types of annotations were used: original and revised. We follow the recommendations of the organizers and use the original data (Ng et al., 2014).

The approaches varied widely: classifiers, MT, rules, hybrid systems. Table 3 summarizes the top five systems. The top team used a hybrid system that combined rules and MT. The second system developed classifiers for common grammatical errors. The third system used MT.

As for external resources, the top 1 and top 3 teams used additional learner data to train their MT systems, the Cambridge University Press Learners' Corpus and the Lang-8 corpus (Mizumoto et al., 2011), respectively. Many teams also used native English datasets. The most common ones are the Web1T corpus (Brants and Franz, 2006), the CommonCrawl dataset, which is similar to Web1T, and the English Wikipedia. Several teams used off-the-shelf spellcheckers.

In addition, Susanto et al. (2014) made an attempt at combining MT and classifiers. They used CoNLL-train and Lang-8 as non-native data and English Wikipedia as native data. We believe that the reason this study did not yield significant improvements (Table 1) is that individual strengths of each framework have not been fully exploited. Further, each system was applied separately and decisions were combined using a general MT combination technique (Heafield et al., 2009). Finally, Mizumoto and Matsumoto (2016) attempt to improve an MT system also trained on Lang-8 with discriminative re-ranking using part-of-speech (POS) and dependency features but only obtain a small improvement. These results suggest that standard combination and re-ranking techniques are not sufficient.

2.2 Overview of the State-of-the-Art

The statistical machine translation approach is based on the noisy-channel model. The best translation for a foreign sentence f is:

$$e^* = \arg\max_e p(e)p(f|e)$$

Error type	Rel. freq. (%)	Examples
Article (*ArtOrDet*)	19.93	**∅/The* government should help encourage *the/∅ breakthroughs as well as *a/∅ complete medication system .
Wrong collocation (*Wci*)	12.51	Some people started to **think/wonder* if electronic products can replace human beings for better performances .
Noun number (*Nn*)	11.44	There are many reports around the internet and on newspaper stating that some users ' **iPhone/iPhones* exploded .
Preposition (*Prep*)	8.98	I do not agree **on/with* this argument...
Word form (*Wform*)	6.56	...the application of surveillance technology serves as a warning to the **murders/murderers* and they might not commit more murder .
Orthography/punctuation (*Mec*)	5.75	Even British Prime Minister , Gordon Brown **∅/,* has urged that all cars in **britain/Britain* to be green by 2020 .
Verb tense (*Vt*)	4.56	Through the thousands of years , most Chinese scholars **are*/{*have been*} greatly affected by Confucianism .
Linking words/phrases (*Trans*)	4.10	**However/Although* , video surveillance may be a great help .
Local redundancy (*Rloc-*)	3.70	Some solutions *{*as examples*}/∅ would be to design plants/fertilizers that give higher yield ...
Subject-verb agreement (*SVA*)	3.58	However , tracking people **are/is* different from tracking goods .
Verb form (*Vform*)	3.52	Travelers survive in desert thanks to GPS **guide/guiding* them .

Table 2: **Example errors.** In the parentheses, the error codes used in the shared task are shown. Errors exemplifying the relevant phenomena are marked; the sentences may contain other mistakes.

Rank	System name	F0.5	Approach	External training data		External error modules
				Native data	Learner data	
1	CAMB	37.33	Rules and MT	Microsoft Web LM	Cambridge Corpus, Eng. Vocab Profile	Cambridge “Write and Improve”
2	CUUI	36.79	Classif.; patterns	Web1T		
3	AMU	35.01	MT	Wikipedia, CommonCrawl	Lang-8	
4	POST	30.88	LM and rules	Web1T		PyEnchant Spell
5	NTHU	29.92	Rules, MT, classif.	Web1T, Gigaword, BNC, Google Books		Spellcheckers: Aspell, GingerIt

Table 3: **The top 5 systems in CoNLL-2014.** The last column lists external proofing tools used. LM stands for *language models*.

The model consists of two components: a *language model* assigning a probability $p(e)$ for any target sentence *e*, and a translation model that assigns a conditional probability $p(f|e)$. The language model is learned using a monolingual corpus in the target language. The parameters of the *translation model* are estimated from a parallel corpus, i.e. the set of foreign sentences and their corresponding translations into the target language. In error correction, the task is cast as translating from erroneous learner writing into corrected well-formed English. The MT approach relies on the availability of a parallel corpus for learning the translation model. In case of error correction, a set of learner sentences and their corrections functions as a parallel corpus.

State-of-the-art MT systems are phrase-based, i.e. parallel data is used to derive a phrase-based lexicon (Koehn et al., 2003). The resulting lexicon consists of a list of pairs (seq_f, seq_e) where seq_f is a sequence of one or more foreign words, seq_e is a predicted translation. Each pair comes with an associated score. At decoding time, all phrases from sentence *f* are collected with their corresponding translations observed in training. These are scored together with the language modeling scores and may include other features. The phrase-based approach by Koehn et al. (2003) uses a log-linear model (Och and Ney, 2002), and the best correction maximizes the following:

$$\begin{aligned} e^* &= \arg\max_e P(e|f) \\ &= \arg\max_e \exp(\sum_{m=1}^{M} \lambda_m h_m(e,f)) \end{aligned} \quad (1)$$

where h_m is a feature function, such as language model score and translation scores, and λ_m corresponds to a feature weight.

The classifier approach is based on the context-sensitive spelling correction methodology (Golding and Roth, 1996; Golding and Roth, 1999; Banko and Brill, 2001; Carlson et al., 2001; Carlson and Fette, 2007) and goes back to earlier approaches to article and preposition error correction (Izumi et al., 2003; Han et al., 2006; Gamon et al., 2008; Felice and Pulman, 2008; Tetreault et al., 2010; Gamon, 2010; Dahlmeier and Ng, 2011; Dahlmeier and Ng, 2012). The classifier approach to error correction has been prominent for a long time before MT, since building a classifier does not require having annotated learner data.

Property	MT	Classifier
(1a) **Error coverage**: ability to address a wide variety of error phenomena	+All errors occurring in the training data are automatically covered	-Only errors covered by the classifiers; new errors need to be added explicitly
(1b) **Error complexity**: ability to handle complex and interacting mistakes that go beyond word boundaries	+Automatically through parallel data, via phrase-based lexicons	-Need to develop via specific approaches
(2) **Generalizability**: going beyond the error confusions observed in training	-Only confusions observed in training can be corrected	+Easily generalizable via confusion sets and features
(3) **Supervision/Annotation**: role of learner data in training the system	-Required	+Not required
(4) **System flexibility**: adapting knowledge sources per error phenomena	-Not easy to integrate error-specific knowledge resources	+Flexible; phenomenon-specific knowledge sources

Table 4: **Summary of the key properties of the MT and the classifier-based approaches.** We use $+$ and $-$ to indicate a positive or a negative value with respect to each factor.

Classifiers are trained individually for a specific error type. Because an error type needs to be defined, typically only well-defined mistakes can be addressed in a straightforward way. Given an error type, a *confusion set* is specified and includes a list of confusable words. For some errors, confusion sets are constructed using a closed list (e.g. prepositions). For other error types, NLP tools are required. To identify locations where an article was likely omitted incorrectly, for example, a phrase chunker is used. Each occurrence of a confusable word in text is represented as a vector of features derived from a *context window* around the target. The problem is cast as a multi-class classification task.

In the classifier paradigm, there are various algorithms – generative (Gamon, 2010; Park and Levy, 2011), discriminative (Han et al., 2006; Gamon et al., 2008; Felice and Pulman, 2008; Tetreault et al., 2010), and joint approaches (Dahlmeier and Ng, 2012; Rozovskaya and Roth, 2013). Earlier works trained on native data (due to lack of annotation). Later approaches incorporated learner data in training in various ways (Han et al., 2010; Gamon, 2010; Rozovskaya and Roth, 2010a; Dahlmeier and Ng, 2011).

3 Error Analysis of MT and Classifiers

This section presents error analysis of the MT and classifier approaches. We begin by identifying several key properties that distinguish between MT systems and classifier systems and that we use to characterize the learning frameworks and the outputs of the systems:

(1a) Error coverage denotes the ability of a system to identify and correct a variety of error types.

(1b) Error complexity indicates the capacity of a system to address complex mistakes such as those where multiple errors interact.

(2) Generalizibility refers to the ability of a system to identify mistakes in new unseen contexts and propose corrections beyond those observed in training data.

(3) The role of supervision or having annotated learner data for training.

(4) System flexibility is a property of the system that allows it to adapt resources specially to correct various phenomena. The two paradigms are summarized in Table 4. We use $+$ and $-$ to indicate whether a learning framework has desirable (+) or undesirable characteristic with regard to each factor.

The first three properties characterize system output, while (3) and (4) arise from the system frameworks. Below we analyze the output of several state-of-the-art CoNLL-2014 systems in more detail.[1] Section 4 explores (3) and (4) that relate to the learning frameworks.

3.1 Error Coverage and Complexity

Error coverage To understand how systems differ with respect to error coverage, we consider recall of each system per error type. Error-type *recall* can be easily computed using error tags and is reported in the CoNLL overview paper.

The recall numbers show substantial variations among the systems. If we consider error categories that have non-negligible recall numbers (higher than 10%), classifier-based approaches have a much lower proportion of error types for which 10% recall was achieved. Among the 28 error types, the top classifier systems – Columbia University-University of Illinois (CUUI, top-2) and National Tsing Hua University (NTHU, top-5) – have a recall higher than 10% for 8 and 9 error types, respectively. In contrast, the two MT-based systems – Cambridge University (CAMB,

[1]Outputs are available on the CoNLL-2014 website.

(1)	It is a concern that will be with us *{*during our whole life*}/{*for our entire life*} .
(2)	The decision to inform relatives of *{*such genetic disorder*}/{*such genetic disorders*} will be dependent . . .
(3)	.. we need to respect it and we have no right *{*in saying*}/{*to say*} that he must tell his relatives about it .
(4)	...and his family might be a *{*genetically risked*}/{*genetic risk*} family .
(5)	...he was **diagnosis*/{*diagnosed with*} a kind of genetic disease which is very serious .
(6)	The situation may become **worst*/*worse* if the child has diseases like cancer or heart disease . . .

Table 5: **Complex and interacting mistakes that MT successfully addresses.** Output of the MT-based AMU system.

top-1) and the Adam Mickiewicz University system (AMU, top-3) – have 15 and 17 error types, respectively, for which the recall is at least 10%.

These recall discrepancies indicate that the MT approach has a better overall coverage, which is intuitive given that all types of confusions are automatically added through phrase-based translation tables in MT, while classifiers must explicitly model each error type. Note, however, that these numbers do not necessarily indicate good type-based performance, since high recall may correspond to low precision.

Error complexity In the MT approach, error confusions are learned automatically via the phrase translation tables extracted from the parallel training data. Thus, an MT system can easily handle interacting and complex errors where replacements involve a sequence of words. Table 5 illustrates complex and interacting mistakes that the MT approach is able to handle. Example (1) contains a phrase-level correction that includes both a preposition replacement and an adjective change. (2) is an instance of an interacting mistake where there is a dependency between the article and the noun number, and a mistake can be corrected by changing one of the properties but not both. (3), (4) and (5) require multiple simultaneous corrections on various words in a phrase. (6) is an example of an incorrect adjectival form, an error that is typically not modeled with standard classifiers.

3.2 Generalizability

Because MT systems extract error/correction pairs from phrase-translation tables, they can only identify erroneous surface forms observed in training and propose corrections that occurred with the corresponding surface forms. Crucially, in a standard MT scenario, any resulting translation consists of "matches" mined from the translation tables, so a standard MT model lacks lexical abstractions that might help generalize, thus out-of-vocabulary words is a well-known problem in MT (Daume and Jagarlamudi, 2011). While more advanced MT models can abstract by adding higher-level features such as POS, previous attempt yielded only marginal improvements (Mizumoto and Matsumoto, 2016), since one typically needs different types of abstractions depending on the error type, as we show below.

Error	AMU (MT)			CUUI (Classif.)		
type	P	R	F0.5	P	R	F0.5
Orthog./punc. (*Mec*)	61.6	16.3	**39.6**	53.3	8.7	26.4
Article (*ArtOrDet*)	38.0	10.9	25.4	31.8	47.9	**34.0**
Preposition (*Prep*)	54.9	10.4	**29.5**	31.7	8.8	20.9
Noun number (*Nn*)	49.6	43.2	**48.2**	42.5	46.2	43.2
Verb tense (*Vt*)	30.2	9.3	20.8	61.1	5.4	19.9
Subj.-verb agr. (*SVA*)	48.3	14.9	33.3	57.7	57.7	**57.7**
Verb form (*Vform*)	40.5	16.8	31.8	69.2	15.1	**40.3**
Word form (*Wform*)	59.0	36.6	**52.6**	60.0	13.5	35.6

Table 6: **Performance of MT and classifier systems from CoNLL-2014 on common errors.**

With classifiers, it is easy to generalize using higher-level information that goes beyond surface form and to adjust the abstraction to the error type. Many grammatical errors may benefit from generalizations based on POS or parse information; we can thus expect that classifiers will do better on errors that require linguistic abstractions.

To validate this hypothesis, we evaluate type-based performance of two systems: a top-3 MT-based AMU system and a top-2 classifier-based CUUI; we do not include the top-1 system, since it is a hybrid system that also uses rules.

Unlike recall, estimating type-based *precision* requires knowing the type of the correction supplied by the system, which is not specified in the output. We thus manually analyze the output of the AMU and CUUI systems for seven common error categories and assign to each correction an appropriate type to estimate precision and F0.5 (Table 6). The CUUI system addresses all of these errors, with the exception of mechanical (Mec), of which it handles a small subset. The AMU system does better on mechanical, preposition, word form, and noun number. CUUI does better on articles, verb agreement, and verb form.

We now consider examples of errors that are corrected by the classifier-based CUUI system in these three categories but are missed by the MT-based AMU system (Table 7). Examples (1) and

	Long-distance dependencies: verb agreement
(1)	As a result , in the case that when one of the members **happen/happens* to feel uncomfortable or abnormal , he or she should be aware that . . .
(2)	A study of New York University in 2010 shown that patients with family members around generally **recovers/recover* 2-4 days faster than those taken care by professional nurses .
	Confusions not found in training: verb agreement and verb form
(3)	Hence , the social media sites **serves/serve* as a platform for the connection .
(4)	After **came/coming* back from the hospital , the man told his parents that the problem was that he carried . . .
(5)	social media is the only resource they can approach to know everything **happened/happening* in their country . . .
	Superfluous words: articles
(6)	For **an/∅* example , if exercising is helpful, we can always look for more chances for the family to go exercise .
(7)	. . . as soon a person is made aware of his or her genetic profile , he or she has **a/∅* knowledge about others .
	Omissions: articles
(8)	In this case , if one of the family members or close relatives is found to carry **∅/a* genetic risk . . .

Table 7: **Generalizing beyond surface form:** Examples of mistakes that classifiers successfully address. Output of the classifier-based CUUI system.

(2) illustrate verb errors with long-distance subjects ("one" and "patients"). This is handled in the classification approach via syntactic features. An MT system misses these errors because it is limited to edits within short spans. Examples (3), (4), and (5) illustrate verb mistakes for which the correct replacements were not observed in training but that are nonetheless corrected by generalizing beyond surface form. Finally, (6) and (7) illustrate omission and insertion errors, a majority of article mistakes. The MT system is especially bad at correcting such mistakes. Notably, the classifier-based CUUI system correctly identified *twice* as many omitted articles and more than *20 times* more superfluous articles than the MT-based AMU system. This happens because an MT system is restricted to suggesting deletions and insertions in those contexts that were observed in training, whereas a classifier uses shallow parse information, which allows it to insert or delete an article in front of every eligible noun phrase. These examples demonstrate that the ability of a system to generalize beyond the surface forms is indeed beneficial for *long-distance dependencies*, for abstracting away from surface forms when *formulating confusion sets*, and for mistakes involving *omitting or inserting a word.*

4 Developing New State-of-the-Art MT and Classifier Systems

In this section, we explore the advantages of each learning approach, as identified in the previous section, within each learning framework. To this end, drawing on the strengths of each framework, we develop new state-of-the-art MT and classifier systems.[2] In the next section, we will use these MT and classifier components and show how to exploit the strengths of each framework in combination. Table 8 summarizes the data used. Results are reported with respect to all errors in the test data. This is different from performance for individual errors in Table 6.

System	Learner		Native	
	CoNLL-train 1.2M	Lang-8 48M	Eng. Wiki. 2B	Web1T 1T
MT	✓	✓	✓	-
Classif.	✓	-	-	✓

Table 8: **Data used in the experiments.** Corpora sizes are in the number of words.

4.1 Machine Translation Systems

A key advantage of the MT framework is that, unlike with classifiers, error confusions are learned from parallel data automatically, without further (linguistic) input. We build two MT systems that differ only in the use of parallel data: the CoNLL-2014 training data and Lang-8. Our MT systems are trained using Moses (Koehn et al., 2007) and follow the standard approach (Junczys-Dowmunt and Grundkiewicz, 2014; Susanto et al., 2014). Both systems use two 5-gram language models – English Wikipedia and the corrected side of CoNLL-train – trained with KenLM (Heafield et al., 2013). Table 9 reports the performance of the systems. As shown, performance increases by more than 11 points when a larger parallel corpus is used. The best MT system outperforms the top CoNLL system by 2 points.

4.2 Classifiers

We now present several classifier systems, exploring the two important properties of the classification framework – the ability to train without super-

[2]Implementation details can be found at `cogcomp.cs.illinois.edu/page/publication_view/793`

Parallel data	Performance		
	P	R	F0.5
CoNLL-train	43.34	11.81	28.25
Lang-8	66.15	15.11	39.48
CoNLL-2014 top 1	39.71	30.10	37.33

Table 9: **MT systems trained in this work**.

vision and system flexibility (see Table 4).

4.2.1 Supervision

Supervision in the form of annotated learner data plays an important role in developing an error correction system but is expensive. Native data, in contrast, is cheap and available in large quantities. Therefore, the fact that, unlike with MT, it is possible to build a classifier system without any annotated data, is a clear advantage of classifiers.

Training without supervision is possible in the classification framework, as follows. For a given mistake type, e.g. preposition, a classifier is trained on native data that is assumed to be correct; the classifier uses context words around each preposition as features. The resulting model is then applied to learner prepositions and will predict the most likely preposition in a given context. If the preposition predicted by the classifier is different from what the author used in text, this preposition is flagged as a mistake. We refer the reader to Rozovskaya and Roth (2010b) and Rozovskaya and Roth (2011) for a description of training classifiers with and without supervision for error correction tasks. Below, we address two questions related to the use of supervision:

• **Training with supervision**: When training using learner data, how does a classifier-based system compare against an MT system?

• **Training without supervision**: How well can we do by building a classifier system with native data only, compared to MT and classifier-based systems that use supervision?

Our classifier system is based on the implementation framework of the second CoNLL-2014 system (Rozovskaya et al., 2014) and consists of classifiers for 7 most common grammatical errors in CoNLL-train: article; preposition; noun number; verb agreement; verb form; verb tense; word form. All modules take as input the corpus documents pre-processed with a POS tagger[3] (Even-Zohar and Roth, 2001), a shallow parser[4] (Punyakanok and Roth, 2001), a syntactic parser (Klein and Manning, 2003) and a dependency converter (Marneffe et al., 2006).

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

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

System	Performance		
	P	R	F0.5
Classifiers (learner)	32.15	17.96	27.76
Classifiers (native)	38.41	23.05	33.89
MT	43.34	11.81	28.25
CoNLL-2014 top 1	39.71	30.10	37.33
CoNLL-2014 top 2	41.78	24.88	36.79
CoNLL-2014 top 3	41.62	21.40	35.01

Table 10: **Classifier systems trained with and without supervision.** *Learner* data refers to CoNLL-train. *Native* data refers to Web1T. The MT system uses CoNLL-train for parallel data.

Classifiers are trained either on learner data (CoNLL-train) or native data (Web1T). Classifiers built on CoNLL-train are trained discriminatively with the Averaged Perceptron algorithm (Rizzolo and Roth, 2010) and use rich POS and syntactic features tailored to specific error types that are standard for these tasks (Lee and Seneff, 2008; Han et al., 2006; Tetreault et al., 2010; Rozovskaya et al., 2011); Naïve Bayes classifiers are trained on Web1T with word n-gram features. A detailed description of the classifiers and the features used can be found in Rozovskaya and Roth (2014). We also add several novel ideas that are described below.

Table 10 shows the performance of two classifier systems, trained with supervision (on CoNLL-train) and without supervision on native data (Web1T), and compares these to an MT approach trained on CoNLL-train. The first classifier system performs comparably to the MT system (27.76 vs. 28.25), however, the native-trained classifier system outperforms both, and does not use any annotated data. The native-trained classifier system would place fourth in CoNLL-2014.

4.2.2 Flexibility

We now explore another advantage of the classifier-based approach, that of allowing for a flexible architecture where we can tailor knowledge sources for individual phenomena. In Section 4.2.1, we already took advantage of the fact that in the classifier framework it is easy to incorporate features suited to individual error types. We now show that by *adding supervision in a way tailored toward specific errors* we can further improve the classifier-based approach.

Adding Supervision in a Tailored Way There is a trade-off between training on native and learner

	Training data	Performance		
		P	R	F0.5
(1)	Learner	32.15	17.96	27.76
(2)	Native	38.41	23.05	33.89
(3)	Tailored	57.07	14.74	**36.26**

Table 11: **Classifiers: supervision in a tailored way**. Trained on (1) learner data (CoNLL-train); (2) native data (Web1T); (3) data sources tailored per error type.

System/training data	Performance		
	P	R	F0.5
Native	38.41	23.05	33.89
Native+mechanical	42.72	27.69	**38.54**
Tailored	57.07	14.74	36.26
Best (tailored+mechanical)	60.79	19.93	**43.11**
CoNLL-2014 top system	39.71	30.10	37.33
Susanto et al. (2014)	53.55	19.14	39.39
Miz. & Mats. (2016)	45.80	26.60	40.00

Table 12: **Classifier systems in this work.** Comparison to existing state-of-the-art.

System	Performance		
	P	R	F0.5
MT is trained on CoNLL-train			
MT	43.34	11.81	28.25
Spelling+MT	49.86	16.36	35.37
Article+MT	45.11	13.99	31.22
Verb agr.+MT	46.36	14.63	32.33
Art.+Verb agr.+Spell+MT	52.07	20.89	**40.10**
MT is trained on Lang-8			
MT	66.15	15.11	39.48
Spelling+MT	65.87	16.94	41.75
Article+MT	63.81	17.70	41.95
Verb. agr.+MT	66.09	18.01	43.08
Art.+Verb agr.+Spell+MT	64.13	22.15	**46.51**

Table 13: **Pipelines: select classifiers and MT.**

data. The advantage of training on native data is clearly the size, which is important for estimating context parameters. Learner data provides additional information, such as learner error patterns and the manner of non-native writing.

Instead of choosing to train on one data type, the classifier framework allows one to combine the two data sources in various ways: voting (Rozovskaya et al., 2014), alternating structure optimization (Dahlmeier and Ng, 2011), training a meta-classifier (Gamon, 2010), and extracting error patterns (Rozovskaya and Roth, 2011). We compare two approaches of adding supervision:
(1) **Learner error patterns:** Error patterns are extracted from learner data and "injected" into models trained on native data (Rozovskaya and Roth, 2011). Learner data is used to estimate mistake parameters; contextual cues are based on native data.
(2) **Learner error patterns+native predictions:** Classifiers are trained on native data. Classifier predictions are used as features in models trained on learner data. Learner data thus contributes both the specific manner of learner writing and the mistake parameters. The native data contributes contextual information.

We found that (2) is superior to (1) for article, agreement, and preposition errors; (1) works better on verb form and word form errors; and noun number errors perform best when a classifier is trained on native data. (Learner error patterns were found not to be beneficial for correcting noun number errors (Rozovskaya and Roth, 2014)). Tailored supervision yields an improvement of almost 3 points over the system trained on native data and almost 9 points over the system trained on learner data (Table 11).

Adding Mechanical Errors Finally, we add components for *mechanical* errors: punctuation, spelling, and capitalization. These are distinguished from the grammatical mistakes, as they are not specific to GEC and can be handled with existing resources or simple methods.

For capitalization and missing commas, we compile a list of patterns using CoNLL training data. We also use an off-the-shelf speller (Flor, 2012; Flor and Futagi, 2012). Results are shown in Table 12. Performance improves by almost 5 and 7 points for the native-trained system and for the best configuration of classifiers with supervision. Both systems also outperform the top CoNLL system, by 1 and 6 points, respectively. The result of 43.11 by the best classifier configuration substantially outperforms the existing state-of-the-art: a combination of two MT systems and two classifier systems, and MT with re-ranking (Susanto et al., 2014; Mizumoto and Matsumoto, 2016).

5 Combining MT and Classifier Systems

Since MT and classifiers differ with respect to the types of errors they can better handle, we combine these systems in a pipeline architecture where the MT is applied to the output of classifiers. Classifiers are applied first, since MT is better at handling complex phenomena. First, we add the speller and those classifier components that perform substantially better than MT (articles and verb agreement), due to the ability of classifiers to generalize beyond lexical information. The added classifiers are part of the best system in Table 12.

Results are shown in Table 13. Adding classifiers improves the performance, thereby demon-

System	Performance		
	P	R	F0.5
MT (CoNLL-train)	43.34	11.81	28.25
MT (Lang-8)	66.15	15.11	39.48
Best classifier (Table 12)	60.79	19.93	43.11
Best class.+MT (CoNLL-train)	51.92	25.08	42.77
Best class.+MT (Lang-8)	60.17	25.64	**47.40**

Table 14: **Pipelines: the best classifier system and MT systems.**

System	Performance		
	P	R	F0.5
Best classifier (Table 12)	60.79	19.93	**43.11**
Art.+Verb agr.+Spell+MT	64.13	22.15	**46.51**
Best classifier+MT	60.17	25.64	**47.40**
CoNLL-2014 top system	39.71	30.10	37.33
Susanto et al. (2014)	53.55	19.14	39.39
Miz. & Mats. (2016)	45.80	26.60	40.00

Table 15: **Best systems in this work.** Comparison to existing state-of-the-art.

strating that the classifiers address a complementary set of mistakes. Adding all three modules improves the results from 28.25 to 40.10 and from 39.48 to 46.51 for the MT systems trained on CoNLL-train and Lang-8, respectively. Notably, the CoNLL-train MT system especially benefits, which shows that when the parallel data is small, it is particularly worthwhile to add classifiers.

It should be stressed that even with a smaller parallel corpus, when the three modules are added, the resulting system is very competitive with previous state-of-the-art that uses a lot more supervision: Susanto et al. (2014) and Mizumoto and Matsumoto (2016) use Lang-8. These results show that *when one has an MT system, it is possible to improve by investing effort into building select classifiers for phenomena that are most challenging for MT*.

Finally, Table 14 demonstrates that combining MT with the best classifier system improves the result further when the MT system is trained on Lang-8, but not when the MT system is trained on CoNLL-train. We also note that the CoNLL-train MT system also has a much lower precision than the other systems. We conclude that when only a limited amount of data is available, the classifier approach on its own performs better.

As a summary, Table 15 lists the best systems developed in this work – a classifier system, a pipeline of select classifiers and MT, and a pipeline consisting of the best classifier and the MT systems – and compares to existing state-of-the-art. Our classifier system is a 3-point improvement over the existing state-of-the-art, while the best pipeline is a 7.4-point improvement (20% relative improvement).

6 Discussion and Conclusions

A recent surge in GEC research has produced two leading state-of-the-art approaches – machine learning classification and machine translation. Based on the analysis of the methods and an error analysis on the outputs of state-of-the-art systems that adopt these approaches, we explained the differences and the key advantages of each. With respect to error phenomena, we showed that while MT is better at handling complex mistakes, classifiers are better at correcting mistakes that require abstracting beyond lexical context. We further showed that the key strengths of the classification framework are its flexibility and the ability to train without supervision.

We built several systems that draw on the strengths of each approach individually and in a pipeline. The best classifier system and the pipelines outperform reported best results on the task, often by a large margin.

The purpose of this work is to gain a better understanding of the advantages offered by each learning method in order to make further progress on the GEC task. We showed that the values provided by each method can be exploited within each approach and in combination, depending on the resources available, such as annotated learner data (MT), and additional linguistic resources (classifiers). As a result, we built robust systems and showed substantial improvement over existing state-of-the-art.

For future work, we intend to study the problem in the context of other languages. However, it is important to realize that the problem is far from being solved even in English, and the current work makes very significant progress on it.

Acknowledgments

The authors thank Michael Flor for his help with running the spelling system on the data and John Wieting for sharing the English Wikipedia corpus. The authors are also grateful to Mark Sammons, Peter Chew, and the anonymous reviewers for the insightful comments on the paper. The work of Dan Roth on this project was supported by DARPA under agreement number FA8750-13-2-0008. 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.

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

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. J. Carlson, J. Rosen, and D. Roth. 2001. Scaling up context sensitive text correction. In *IAAI*.

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*.

H. Daume and J. Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *ACL*.

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

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. Felice, Z. Yuan, Ø. Andersen, H. Yannakoudakis, and E. Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

M. Flor and Y. Futagi. 2012. On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

M. Flor. 2012. Four types of context for automatic spelling correction. *Traitement Automatique des Langues (TAL). (Special Issue: Managing noise in the signal: error handling in natural language processing)*, 3(53):61–99.

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

A. R. Golding and D. Roth. 1996. Applying Winnow to context-sensitive spelling correction. In *ICML*.

A. R. Golding and D. Roth. 1999. A Winnow based approach to context-sensitive spelling correction. *Machine Learning*.

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*.

K. Heafield, G. Hanneman, and A. Lavie. 2009. Machine translation system combination with flexible word ordering. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Association for Computational Linguistics.

K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL*.

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*.

M. Junczys-Dowmunt and R. Grundkiewicz. 2014. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

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

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *ACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

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

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

T. Mizumoto and Y. Matsumoto. 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In *NAACL. To appear*.

T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto. 2011. Mining revision log of language learning SNS for automated japanese error correction of second language learners. In *IJCNLP*.

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*.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*.

A. Park and R. Levy. 2011. Automated whole sentence grammar correction using a noisy channel model. In *ACL*, Portland, Oregon, USA, June. Association for Computational Linguistics.

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

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. Generating confusion sets for context-sensitive error correction. In *Proceedings of EMNLP*.

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 and D. Roth. 2014. Building a State-of-the-Art Grammatical Error Correction System. In *Transactions of ACL*.

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

R. H. Susanto, P. Phandi, and H. T. Ng. 2014. System combination for grammatical error correction. In *EMNLP*.

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