Systematically Adapting Machine Translation for Grammatical Error Correction

Courtney Napoles* and **Chris Callison-Burch**†
*Center for Language and Speech Processing, Johns Hopkins University
†Computer and Information Science Department, University of Pennsylvania
napoles@cs.jhu.edu, ccb@cis.upenn.edu

Abstract

In this work we adapt machine translation (MT) to grammatical error correction, identifying how components of the statistical MT pipeline can be modified for this task and analyzing how each modification impacts system performance. We evaluate the contribution of each of these components with standard evaluation metrics and automatically characterize the morphological and lexical transformations made in system output. Our model rivals the current state of the art using a fraction of the training data.

1 Introduction

This work presents a systematic investigation for automatic grammatical error correction (GEC) inspired by machine translation (MT). The task of grammatical error correction can be viewed as a noisy channel model, and therefore a MT approach makes sense, and has been applied to the task since Brockett et al. (2006). Currently, the best GEC systems all use machine translation in some form, whether statistical MT (SMT) as a component of a larger pipeline (Rozovskaya and Roth, 2016) or neural MT (Yuan and Briscoe, 2016). These approaches make use of a great deal of resources, and in this work we propose a lighter-weight approach to GEC by methodically examining different aspects of the SMT pipeline, identifying and applying modifications tailored for GEC, introducing artificial data, and evaluating how each of these specializations contributes to the overall performance.

Specifically, we demonstrate that

- Artificially generated rules improve performance by nearly 10%.
- Custom features describing morphological and lexical changes provide a small performance gain.
- Tuning to a specialized GEC metric is slightly better than tuning to a traditional MT metric.
- Larger training data leads to better performance, but there is no conclusive difference between training on a clean corpus with minimal corrections and a noisy corpus with potential sentence rewrites.

We have developed and will release a tool to automatically characterize the types of transformations made in a corrected text, which are used as features in our model. The features identify general changes such as insertions, substitutions, and deletions, and the number of each of these operations by part of speech. Substitutions are further classified by whether the substitution contains a different inflected form of the original word, such as change in verb tense or noun number; if substitution has the same part of speech as the original; and if it is a spelling correction. We additionally use these features to analyze the outputs generated by different systems and characterize their performance with the types of transformations it makes and how they compare to manually written corrections in addition to automatic metric evaluation.

Our approach, Specialized Machine translation for Error Correction (SMEC), represents a single model that handles morphological changes, spelling corrections, and phrasal substitutions, and it rivals the performance of the state-of-the-art neural MT system (Yuan and Briscoe, 2016), which uses twice the amount of training data, most of which is not publicly available. The analysis provided in this work will help improve future efforts in GEC, and can be used to inform approaches rooted in both neural and statistical MT.

Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 345–356
Copenhagen, Denmark, September 8, 2017. ©2017 Association for Computational Linguistics

2 Related work

Earlier approaches to grammatical error correction developed rule-based systems or classifiers targeting specific error types such as prepositions or determiners, (e.g., Eeg-Olofsson and Knutsson, 2003; Tetreault and Chodorow, 2008; Rozovskaya et al., 2014), and few approaches were rooted in machine translation, though some exceptions exist (Brockett et al., 2006; Park and Levy, 2011, e.g.,). The 2012 and 2013 shared tasks in GEC both targeted only certain error types (Dale et al., 2012; Ng et al., 2013), to which classification was appropriately suited. However, the goal of the 2014 CoNLL Shared Task was correcting all 28 types of grammatical errors, encouraging several MT-based approaches to GEC, (e.g., Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014). Two of the best CoNLL 2014 systems used MT as a black box, reranking output (Felice et al., 2014), and customizing the tuning algorithm and using lexical features (Junczys-Dowmunt and Grundkiewicz, 2014). The other leading system was classification-based and only targeted certain error types (Rozovskaya et al., 2014). Performing less well, Wang et al. (2014) used factored SMT, representing words as factored units to more adeptly handle morphological changes. Shortly after the shared task, a system combining classifiers and SMT with no further customizations reported better performance than all competing systems (Susanto et al., 2014)

The current leading GEC systems all use MT in some form, including hybrid approaches that use the output of error-type classifiers as MT input (Rozovskaya and Roth, 2016) or include a neural model of learner text as a feature in SMT (Chollampatt et al., 2016); phrase-based MT with sparse features tuned to a GEC metric (Junczys-Dowmunt and Grundkiewicz, 2016); and neural MT (Yuan and Briscoe, 2016). Three of these model have been evaluated on a separate test corpus and, while the PBMT system reported the highest scores on the CoNLL-14 test set, it was outperformed by the systems with neural components on the new test set (Napoles et al., 2017).

2.1 GEC corpora

There are two broad categories of parallel data for GEC. The first is error-coded text, in which annotators have coded spans of learner text containing an error, and which includes the NUS Corpus of Learner English (NUCLE; 57k sentence pairs) (Dahlmeier et al., 2013), the Cambridge Learner Corpus (CLC; 1.9M pairs per Yuan and Briscoe (2016)) (Nicholls, 2003), and a subset of the CLC, the First Certificate in English (FCE; 34k pairs) (Yannakoudakis et al., 2011). MT systems are trained on parallel text, which can be extracted from error-coded corpora by applying the annotated corrections, resulting a clean corpus with nearly-perfect word and sentence alignments.[1] These corpora are small by MT training standards and constrained by the coding approach, leading to minimal changes that may result in ungrammatical or awkward-sounding text (Sakaguchi et al., 2016).

The second class of GEC corpora are parallel datasets, which contain the original text and a corrected version of the text, without explicitly coded error corrections. These corpora need to be aligned by sentences and tokens, and automatic alignment introduces noise. However, these datasets are cheaper to collect, significantly larger than the error-coded corpora, and may contain more extensive rewrites. Additionally, corrections of sentences made without error coding are perceived to be more grammatical. Two corpora of this type are the Automatic Evaluation of Scientific Writing corpus, with more than 1 million sentences of scientific writing corrected by professional proofreaders (Daudaravicius et al., 2016), and the Lang-8 Corpus of Learner English, which contains 1 million sentence pairs scraped from an online forum for language learners, which were corrected by other members of the `lang-8.com` online community (Tajiri et al., 2012). Twice that many English sentence pairs can be extracted from version 2 of the Lang-8 Learner Corpora (Tomoya et al., 2011).

We will include both types of corpora in our experiments in Section 4.

2.2 Evaluation

GEC systems are automatically evaluated by comparing their output on sentences that have been manually annotated corpora. The Max-Match metric (M^2) is the most widely used, and calculates the $F_{0.5}$ over phrasal edits (Dahlmeier and Ng, 2012). Napoles et al. (2015) proposed a

[1] Alignment mistakes may occur when sentences are split or joined, or when errors and corrections span multiple tokens, in which the automatic alignment within that span may err.

new metric, GLEU, which has stronger correlation with human judgments. GLEU is based on BLEU and therefore is well-suited for MT. It calculates the n-gram overlap, rewarding n-grams that systems correctly changed and penalizing n-grams that were incorrectly left unchanged. Unlike M^2, it does not require token-aligned input and therefore is able to evaluate sentential rewrites instead of minimal error spans. Since both metrics are commonly used, we will report the scores of both metrics in our results. A new test set for GEC was recently released, JFLEG (Napoles et al., 2017), Unlike the CoNLL 2014 test set, which is a part of the NUCLE corpus, JFLEG contains fluency-based edits instead of error-coded corrections. Like the Lang-8 and AESW corpora, fluency edits allow full sentence rewrites and do not constrain corrections to be error coded, and humans perceive sentences corrected with fluency edits to be more grammatical than those corrected with error-coded edits alone (Sakaguchi et al., 2016). Four leading systems were evaluated on JFLEG, and the best system by both automatic metric and human evaluation is the neural MT system of Yuan and Briscoe (2016) (henceforth referred to as *YB16*).

3 Customizing statistical machine translation

Statistical MT contains various components, including the training data, feature functions, and an optimization metric. This section describes how we customized each of these components.

3.1 Training data

A translation grammar is extracted from the training data, which is a large parallel corpus of ungrammatical and corrected sentences. Each rule is of the form

left-hand side (LHS) → *right-hand side (RHS)*

and has a feature vector, the weights of which are set to optimize an objective function, which in MT is metric like BLEU. A limiting factor on MT-based GEC is the available training data, which is small when compared to the data available for bilingual MT, which commonly uses 100s of thousands or millions of aligned sentence pairs. We hypothesize that artificially generating transformation rules may overcome the limit imposed by lack of sufficiently large training data and improve performance. Particularly, the prevalence of spelling errors is amplified in sparse data due to the potentially infinite possible misspellings and large number of OOVs. Previous work has approached this issue by including spelling correction as a step in a pipeline (Rozovskaya and Roth, 2016).

Our solution is to artificially generate grammar rules for spelling corrections and morphological changes. For each word in the input, we query the Aspell dictionary with PyEnchant[2] for spelling suggestions and create new rules for each correction, e.g.

publically → *public ally*

publically → *publicly*

Additionally, sparsity in morphological variations may arise in datasets. Wang et al. (2014) approached this issue with factored MT, which translates at the sub-word level. Instead, we also generate artificial translation rules representing morphological transformations using RASP's morphological generator, `morphg` (Minnen et al., 2001). We perform POS tagging with the Stanford POS tagger (Toutanova et al., 2003) and create rules to switch the plurality of nouns (e.g., singular ↔ plural). For verbs, we generate rules that change that verb to every other inflected form, specifically the base form, third-person singular, past tense, past participle, and progressive tense (e.g., *wake*, *wakes*, *woke*, *woken*, *waking*). Generated words that did not appear in the PyEnchant dictionary were excluded.

3.2 Features

Each grammar rule has scores assigned by several feature functions $\vec{\varphi} = \{\varphi_1 ... \varphi_N\}$ that are combined in a log-linear model as that rule's weight, with parameters $\vec{\lambda}$ set during tuning.

$$w = -\sum_{i=1}^{N} \lambda_i \log \varphi_i$$

In SMT, these features typically include a phrase penalty, lexical and phrase translation probabilities, a language model probability, binary indicators for purely lexical and monotonic rules, and counters of unaligned words and rule length. Previous work in other monolingual "translation" tasks has achieved success in using features tailored to that task, such as a measure of the relative lengths for sentence compression (Ganitkevitch et al., 2011) or lexical complexity for sentence simplification (Xu et al., 2016). For GEC,

[2] `https://pythonhosted.org/pyenchant/`

Junczys-Dowmunt and Grundkiewicz (2016) used a large number of sparse features for a phrase-based MT system that achieved state of the art performance on the CoNLL-2014 test set. Unlike that work, which uses a potentially infinite amount of sparse features, we choose to use a discrete set of feature functions that are informed by this task. Our feature extraction relies on a variety of pre-existing tools, including fast-align for word alignment (Dyer et al., 2013), trained over the parallel FCE, Lang-8, and NUCLE corpora; PyEnchant for detecting spelling changes; the Stanford POS tagger; the RASP morphological analyzer, `morpha` (Minnen et al., 2001); and the NLTK WordNet lemmatizer (Bird et al., 2009).

Given a grammatical rule and an alignment between tokens on the LHS and RHS, we tag the tokens with their part of speech and label the lemma and inflection of nouns and verbs with `morpha` and the lemma of each adjective and adverb with the WordNet lemmatizer. We then collect count-based features for individual operations and rule-level qualities. An operation is defined as a deletion, insertion, or substitution of a pair of aligned tokens (or a token aligned with ϵ). An aligned token pair is represented as (l_i, r_j), where l_i is a token on the LHS at index i, and similarly r_j for the RHS. The operation features, below, are calculated for each (un)aligned token and summed to attain the value for a given rule.

- **All operations**
  - `CLASS`-error(l_i) for deletions and substitutions
  - `CLASS`-error(r_j) for insertions

  `CLASS` refers to the broad word class of a token, such as *noun* or *verb*.
- **Deletions**
  - is-deleted(l_i)
  - `TAG`-deleted(l_i)

  `TAG` is the PTB part-of-speech tag of a token (e.g., *NN*, *NNS*, *NNP*, etc.),
- **Insertions**
  - is-inserted(r_j)
  - `TAG`-inserted(r_j)
- **Substitutions**
  - is-substituted(r_j)
  - `TAG`-substituted(r_j)
  - `TAG`-substituted-with-`TAG`(l_i, r_j)

  Morphological features:
  - inflection-change-same-lemma(l_i, r_j)
  - inflection-and-lemma-change(l_i, r_j)
  - lemma-change-same-inflection(l_i, r_j)

  Spelling features:
  - not-in-dictionary(l_i)
  - spelling-correction(l_i, r_j)

  Counts of spelling corrections are weighted by the probability of r_j in an English Gigaword language model.

We also calculate the following rule-level features:

- character Levenshtein distance(LHS, RHS)
- token Levenshtein distance(LHS, RHS)
- $\frac{\text{\# tokens(RHS)}}{\text{\# tokens(LHS)}}$
- $\frac{\text{\# characters(RHS)}}{\text{\# characters(LHS)}}$

In total, we use 24 classes and 45 tags. The total number of features 2,214 but only 266 were seen in training (due to unseen `TAG-TAG` substitutions). We additionally include 19 MT features calculated during grammar extraction.[3] Previous MT approaches to GEC, have included Levenshtein distance as a feature for tuning (Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014, 2016), and Junczys-Dowmunt and Grundkiewicz (2016) also used counts of deletions, insertions, and substitutions by word class. They additionally had sparse features with counts of each lexicalized operation, e.g. substitute(*run*, *ran*), which we avoid by abstracting away from the lemmas and instead counting the operations by part of speech and indicating if the lemmas matched or differed for substitutions. An example rule with its feature values is found in Table 1. For artificially generated rules, the MT features are all assigned a 0-value rather than estimating what that value should be, since the artificial rules are unseen in the training data.

3.3 Metric

The decoder identifies the most probable derivation of an input sentence from the translation grammar. Derivations are scored by a combination of a language model score and weighted feature functions, and the weights are optimized to a specific metric during the tuning phase. Recent work has shown that MT metrics like BLEU are not sufficient for evaluating GEC (Grundkiewicz et al., 2015; Napoles et al., 2015) or tuning MT systems for GEC (Junczys-Dowmunt and Grundkiewicz,

[3] More details about the features can be found at `https://github.com/cnap/smt-for-gec`.

Rule	
argued that → may argue that	
Alignment	
(ϵ, may), (argued, argue), (that, that)	
Feature	**Value**
Verb error	2
Substituted	1
Inserted	1
MD inserted	1
VB is substituted	1
VBD substituted with VB	1
Inflection change, same lemma	1
Token LD	2
Character LD	5

Table 1: An example rule from our grammar and the non-zero feature values from Section 3.2.

1. Generate POS tags of the cased input sentence
2. Label proper nouns in the input
3. Align the cased input tokens with the output
4. Capitalize the first alphanumeric character of the output sentence (if a letter).
5. For each pair of aligned tokens (l_i, r_j), capitalize r_j if l_i is labeled a proper noun or r_j is the token "i".

Table 2: A simple recasing algorithm, which relies on token alignments between the input and output.

2016). Fundamentally, MT metrics do not work for GEC because the output is usually very similar to the input, and therefore the input already has a high metric score. To address this issue, we tune to GLEU, which was specifically designed for evaluating GEC output. We chose GLEU instead of M^2 because the latter requires a token alignment between the input, output, and gold-standard references, and assumes only minimal, non-overlapping changes have been made. GLEU, on the other hand, measures n-gram overlap and therefore is better equipped to handle movement and changes to larger spans of text.

4 Experiments

For our experiments, we use the Joshua 6 toolkit (Post et al., 2015). Tokenization is done with Joshua and token-level alignment with fast-align (Dyer et al., 2013). All text is lowercased, and we use a simple algorithm to recase the output (Table 2). We extract a hierarchical phrase-based translation model with Thrax (Weese et al., 2011) and perform parameter tuning with pairwise ranked optimization in Joshua. Our training data is from the Lang-8 corpus (Tomoya et al., 2011), which contains 1 million parallel sentences, and grammar is extracted from the 563k sentence pairs that contain corrections. Systems are tuned to the JFLEG tuning set (751 sentences) and evaluated on the JFLEG test set (747 sentences). We use an English Gigaword 5-gram language model.

We evaluate performance with two metrics, GLEU and M^2, which have similar rankings and match human judgments on the JFLEG corpus (Napoles et al., 2017). We use two baselines: the first has misspellings corrected with Enchant (Sp. Baseline), and the second is an unmodified MT pipeline trained on the Lang-8 corpus, optimized to BLEU with no specialized features (MT Baseline), and we compare our performance to the current state of the art, YB16. While we train on about half a million sentence pairs, YB16 had nearly 2 million sentence pairs for training.[4] We additionally report metric scores for the human corrections, which we determine by evaluating each reference set against the other three and reporting the mean score.

All systems outperform both baselines, and the spelling baseline is stronger than the MT baseline. The spelling baseline also has the highest precision except for the best automatic system, YB16, demonstrating that spelling correction is an important component in this corpus. There is a disparity in the GLEU and M^2 scores for the baseline: the baseline GLEU is about 5% lower than the other systems but the M^2 is 30% lower. This can be attributed to the lesser extent of changes made by the baseline system which results in low recall for M^2 but which is not penalized by GLEU, which is a precision-based metric. The human corrections have the highest metric scores, and make changes to 77% of the sentences, which is in between the number of sentences changed by YB16 and SMEC, however the human corrections have a higher mean edit distance, because the annotators made more extensive changes when a sentence needed to be corrected than any of the models.

Our fully customized model with all modifications, Specialized Machine translation for Er-

[4] Drawn from the CLC, which is not public.

ror Correction (SMEC$^{+\mathrm{morph}}$), scores lower than YB16 according to GLEU but has the same M^2 score. SMEC$^{+\mathrm{morph}}$ has higher M^2 recall, and visual examination of the output supports this, showing many incorrect or unnecessary number of tense changes. Automatic analysis reveals that it makes significantly more inflection changes than the humans or YB16 (detected with the same method described in Section 3.2), from which we can conclude that the morphological rules errors are applied too liberally. If we remove the generated morphological rules but keep the spelling rules (SMEC), performance improves by 0.4 GLEU points and decreases by 0.1 M^2 points—but, more importantly, this system has higher precision and lower recall, and makes more conservative morphological changes. Therefore, we consider SMEC, the model without artificial morphological rules, to be our best system.

The metrics only give us a high-level overview of the changes made in the output. With error-coded text, the performance by feature type can be examined with M^2, but this is not possible with GLEU or the un-coded JFLEG corpus. To investigate the types of changes systems make on a more granular level, we apply the feature extraction method described in Section 3.2 to quantify the morphological and lexical transformations. While we developed this method for scoring translation rules, it can work on any aligned text, and is similar to the forthcoming ERRANT toolkit, which is uses a rule-based framework for automatically categorizes grammatical edits (Bryant et al., 2017). We calculate the number of each of these transformations made by to the input by each system and the human references, determining significant differences with a paired t-test ($p < 0.05$). Figure 1 contains the mean number of these transformations per sentence made by SMEC, YB16, and the human-corrected references, and Figure 2 shows the number of operations by part of speech. Even though the GLEU and M^2 scores of the two systems are nearly identical, they are significantly different in all of the transformations in Figure 1, with SMEC having a higher edit distance from the original, but YB16 making more insertions and substitutions. Overall, the human corrections have a significantly more inserted tokens than either system, while YB16 makes the most substitutions and fewer deletions than SMEC or the human corrections. The bottom plot displays the mean number of operations by part of speech (operations include deletion, insertion, and substitution). Both systems and the human corrections display similar rates of substitutions across different parts of speech, however the human references have significantly more preposition and verb operations and there are significant differences between the determiner and noun operations made by YB16 compared to SMEC and the references. This information can be further analyzed by part of speech and edit operation, and the same information is available for other word classes.

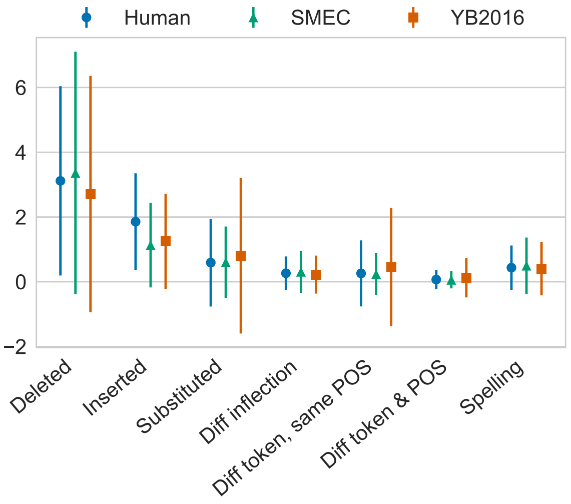

Figure 1: Mean tokens per sentence displaying certain changes from the input sentence.

5 Model analysis

We wish to understand how each component of our model contributes to its performance, and therefore train a series of variations of the model, each time removing a single customization, specifically: the optimization metric (tuning to BLEU instead of GLEU; SMEC$^{-\mathrm{GLEU}}$), the features (only using the standard MT features; SMEC$^{-\mathrm{feats}}$), and eliminating artificial rules (SMEC$^{-\mathrm{sp}}$). The impact of training data size will be investigated separately in Section 5.1. We computed the automatic metric scores of each model variation and performed the automatic edit analysis described in Section 3.2. In Table 4, we report the net metric increase or decrease compared to the full model, and the percent increase or decrease for each of the features. Changing the metric from GLEU to BLEU significantly decreases the amount of change made by the model,

System	GLEU	M² P	M² R	M² $F_{0.5}$	Sents. changed	Edit distance (tokens)
Sp. Baseline	55.5	57.7	16.6	38.4	42%	0.8
MT Baseline	54.9	56.7	14.6	36.0	39%	0.7
SMEC$^{+\text{morph}}$	57.9	54.7	**44.2**	**52.3**	**88%**	**2.8**
SMEC	58.3	55.9	41.1	52.2	85%	2.5
YB16	**58.4**	**59.4**	35.3	**52.3**	73%	1.9
Human	62.1	67.0	52.9	63.6	77%	3.1

Table 3: Results on the JFLEG test set. In addition to the GLEU and M^2 scores, we also report the percent of sentences changed from the input and the mean Levensthein distance.

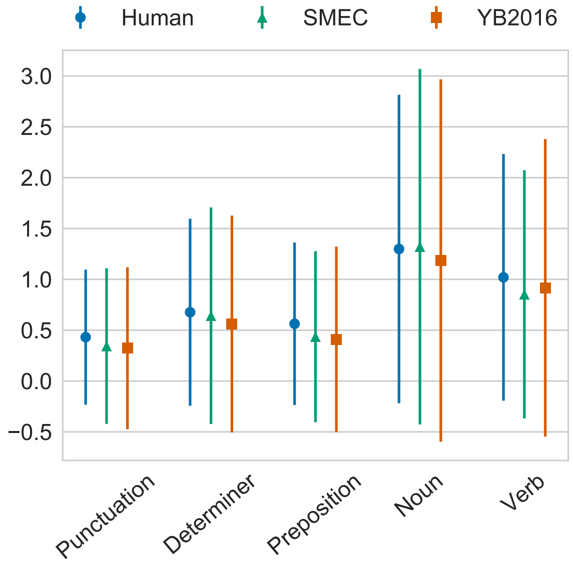

Figure 2: Mean number of operations (deletions, insertions, and substitutions) per sentence by part of speech.

SMEC$^{-\text{GLEU}}$, with a 60% lower edit distance than SMEC, and at least 50% fewer of almost all transformations. The GLEU score of this system is nearly 1 point lower, however there is almost no change in the M^2 score, indicating that the changes made were appropriate, even though they were fewer in number. Tuning to BLEU causes fewer changes because the input sentence already has a high BLEU score due to the high overlap between the input and reference sentences. GLEU encourages more changes by penalizing text that should have been changed in the output.

Removing the custom features (SMEC$^{-\text{feats}}$) makes less of a difference in the GLEU score, however there are significantly more determiners added and more tokens are substituted with words that have different lemmas and parts of speech. This suggests that the specialized features encouraged morphologically-aware substitutions, reducing changes that did not have semantic or functional overlap with the original content. Removing the artificially generated spelling rules (SMEC$^{-\text{sp}}$) had the greatest impact on performance, with a nearly 4-point decrease in GLEU score and 9.5-decrease in M^2. Without spelling rules, significantly fewer tokens were inserted in the corrections across all word classes. We also see a significantly greater number of substitutions made with words that had neither the same part of speech or lemma as the original word, which could be due to sparsity in the presence of spelling errors which is addressed with the artificial grammar.

Table 5 contains example sentences from the test set with system outputs that illustrate these observations. These ungrammatical sentences range from one that can easily be corrected using *minimal* edits; to a sentence that requires more significant changes and inference but has an obvious meaning; to a sentence that is garbled and does not have an immediately obvious correction, even to a native speaker. The reference correction contains more extensive changes than the automatic systems and makes spelling corrections not found by the decoder (*engy* → *energy*) or inferences in the instance of the garbled third sentence, changing *lrenikg* → *Ranking*. SMEC makes many spelling corrections and makes more insertions, substitutions, and deletions than the two SMEC variations. However, the artificial rules also cause some bad corrections, found in the third example changing *studens* → *stud-ens*, while the intended word, *students*, is obvious to a human reader. When optimizing to BLEU instead of the custom metric (SMEC$^{-\text{GLEU}}$), there are fewer changes and therefore output is less fluent. In the first example, SMEC$^{-\text{GLEU}}$ applies only one spelling change even though the rest of the sentence has many small errors that were all corrected in SMEC, such as missing determiner and extra auxiliary. The

Score				
	SMEC	**SMEC** −GLEU	**SMEC** −feats	**SMEC** −sp
GLEU	58.3	57.6	58.1	54.4
M^2	52.2	44.9	47.7	42.7
Transformation				
Edit dist		−60%		−11%
Deleted		−51%	−7%	−4%
Inserted		−46%		−24%
Substituted		−37%		+9%
Diff inflection		−53%		
Diff token		−18%	+9%	
Diff token&POS		−29%	+24%	+31%
Spelling		−35%	+8%	
Determiner		−51%		−7%
	del	−47%	−5%	
	ins	−70%	+39%	−30%
	sub	−56%		
Preposition		−52%		
	del	−51%		
	ins	−45%	−10%	−22%
	sub	−42%		
Noun		−40%	−6%	−10%
	del	−45%	−9%	−12%
	ins	−38%	−7%	−13%
	sub	−30%		
Verb		−57%	−5%	−11%
	del	−61%	−6%	−7%
	ins	−53%	−13%	−40%
	sub	−50%		
Punctuation		−52%		
	del	−47%	−5%	
	ins	−84%		−30%
	sub			

Table 4: Modifications of SMEC, reporting the mean occurrence of each transformation per sentence, when there is a significant difference ($p < 0.05$ by a paired t-test). We report the difference with percentages because each transformation occurs with different frequency.

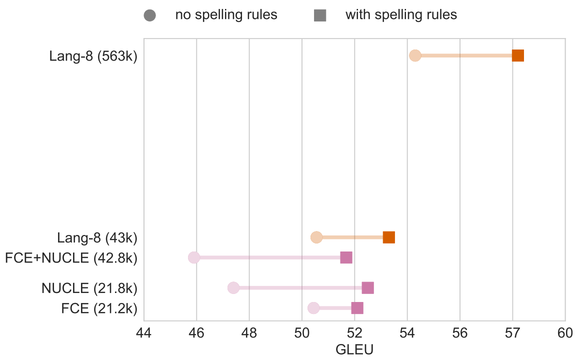

Figure 3: GLEU scores of SMEC with different training sizes, with and without artificial rules.

same pattern is visible in the other two examples. Finally, without the artificial rules, SMEC$^{-\text{sp}}$ fixes only a fraction of the spelling mistakes—however it is the only system that correctly changes *studens* → *students*. Independent from these modifications, the capitalization issues present in the input were all remedied by our recasing algorithm, which improves the metric score.

5.1 Impact of training data

Lang-8 is the largest publicly available parallel corpus for GEC, with 1 million tokens and approximately 563k corrected sentence pairs, however this corpus may contain noise due to automatic alignment and the annotators, who were users of the online Lang-8 service and may not necessarily have provided accurate or complete corrections. Two other corpora, FCE and NUCLE, contain annotations by trained English instructors and absolute alignments between sentences, however each is approximately 20-times smaller than Lang-8. We wish to isolate the effect of size and source of training data has on system performance, and therefore randomly sample the Lang-8 corpus to create a training set the same size as FCE and NUCLE (43k corrected sentence pairs), train a model following the same procedure described above. We hypothesized that including artificial rules may help address problems of sparsity in the training data, and therefore we also train additional models with and without spelling rules to determine how artificial data affects performance as the amount of training data increases. Figure 3 shows the relative GLEU scores of systems with different training data sizes and sources, before and after adding artificial spelling rules.

More data increases performance for Lang-8, however there is no clear relationship between size and performance on the FCE+NUCLE data. Models trained on FCE, NUCLE, and FCE/NUCLE all have similar performance. And, training on 43k Lang-8 sentence pairs slightly improves performance over training on just FCE/NUCLE, suggesting that more data negates the presence of noise and the sentential rewrites present in Lang-8 are better for training a GEC system. The rewrites in Lang-8 could be more similar to those found in JFLEG since both datasets allow for broader fluency changes instead of corrections to coded spans of text. In future work, we will train on version 2 of the Lang-8 Learner Corpus, which has twice

Orig	Unforturntly , almost older people can not use internet , in spite of benefit of internet .
Human	**Unfortunately** , **most** older people can not use **the** internet , in spite of **benefits** of **the** internet .
SMEC	**Unfortunately** , **most** older people can not use **the** internet , in spite of **the benefits** of **the** internet .
SMEC^{-GLEU}	**Unfortunately** , almost older people can not use internet , in spite of benefit of internet .
SMEC^{-sp}	Unforturntly , □ older people can not use **the** internet , in spite of **the benefits** of **the** internet .
Orig	becuse if i see some one did somthing to may safe me time and engy and it wok 's i will do it .
Human	**Because** if **I** see **that someone** did **something that** may **save** me time and **energy** and it **works I** will **also** do it .
SMEC	**Because** if **I** see □ one did **something** □ may **save** me time and **edgy** and □ **work** □ **, I** will do it .
SMEC^{-GLEU}	**Because** if **I** see some one did **something** to may **save** me time and **edgy** and it wok 's **I** will do it .
SMEC^{-sp}	**Because** if **I** see □ one **somthings** □ may **save** me time and engy □ **work** □ **I** will do it .
Orig	lrenikg the studens the ideas have many advantegis :
Human	**Ranking** the **students '** □ ideas **has** many **advantages** .
SMEC	**Linking** the **stud-ens** □ ideas have many **advantages** :
SMEC^{-GLEU}	**Linking** the **stud-ens** the ideas have many **advantages** :
SMEC^{-sp}	**Lrenikg** □ **students** □ ideas have □ advantegis :

Table 5: Example corrections made by a human annotator, SMEC, and two variations: trained on BLEU instead of GLEU SMEC$^{-\text{GLEU}}$ and without artificial spelling rules (SMEC$^{-\text{sp}}$). Inserted or changed text is in **bold** and deleted text is indicated with □.

as much data as the version used in this work, to determine whether performance continues to improve. For all models, adding artificial spelling rules improves performance by about 4 GLEU points (adding spelling rules to FCE training data only causes a 2-point GLEU improvement). The amount of performance does not change related to the size of the training data, however the consistent improvement supports our hypothesis that artificial rules are useful to address problems of data sparsity.

6 Conclusion

This paper has presented a systematic investigation into the components of a standard statistical MT pipeline that can be customized for GEC. The analysis performed on the contribution of each component of the system can inform the design of future GEC models. We have found that extending the translation grammar with artificially generated rules for spelling correction can increase the M^2 score by as much as 20%. The amount of training data also has a substantial impact on performance, increasing GLEU and M^2 scores by approximately 10%. Tuning to a specialized GEC metric and using custom features both help performance but yield less considerable gains. The performance of our model, SMEC, is on par with the current state-of-the-art GEC system, which is neural MT trained on twice the training data, and our analysis suggests that the performance of SMEC would continue to improve if trained on that amount of data. In future work we will test this hypothesis with the larger parallel corpus extracted from version 2 of the Lang-8 Learner Corpora. Our code will be available for automatic feature extraction and edit analysis, as well as more details about the model implementation.[5]

Acknowledgments

We are grateful to the anonymous reviewers for their detailed and thoughtful feedback.

[5] https://github.com/cnap/smt-for-gec

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, pages 249–256. https://doi.org/10.3115/1220175.1220207.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Vancouver, Canada.

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016. Adapting grammatical error correction based on the native language of writers with neural network joint models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1901–1911. https://aclweb.org/anthology/D16-1195.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Montréal, Canada, pages 568–572. http://www.aclweb.org/anthology/N12-1067.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Atlanta, Georgia, pages 22–31. http://www.aclweb.org/anthology/W13-1703.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Montréal, Canada, pages 54–62. http://www.aclweb.org/anthology/W12-2006.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, San Diego, CA, pages 53–62. http://www.aclweb.org/anthology/W16-0506.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 644–648. http://www.aclweb.org/anthology/N13-1073.

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

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 15–24. http://www.aclweb.org/anthology/W14-1702.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., pages 1168–1179. http://www.aclweb.org/anthology/D11-1108.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 461–470. http://aclweb.org/anthology/D15-1052.

Marcin Junczys-Dowmunt and Roman 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*. Association for Computational Linguistics, Baltimore, Maryland, pages 25–33. http://www.aclweb.org/anthology/W14-1703.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1546–1556. https://aclweb.org/anthology/D16-1161.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering* 7(03):207–223.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 588–593. http://www.aclweb.org/anthology/P15-2097.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Meeting of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1–12.

Diane Nicholls. 2003. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 conference*. volume 16, pages 572–581.

Y Albert Park and Roger Levy. 2011. Automated whole sentence grammar correction using a noisy channel model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 934–944.

Matt Post, Yuan Cao, and Gaurav Kumar. 2015. Joshua 6: A phrase-based and hierarchical statistical machine translation system. *The Prague Bulletin of Mathematical Linguistics* 104(1):5–16.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 34–42. http://www.aclweb.org/anthology/W14-1704.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2205–2215. http://www.aclweb.org/anthology/P16-1208.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics* 4:169–182.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 951–962. http://www.aclweb.org/anthology/D14-1102.

Toshikazu Tajiri, Mamoru Komachi, and Yuji 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)*. Association for Computational Linguistics, Jeju Island, Korea, pages 198–202. http://www.aclweb.org/anthology/P12-2039.

Joel Tetreault and Martin Chodorow. 2008. Native judgments of non-native usage: Experiments in preposition error detection. In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*. Coling 2008 Organizing Committee, Manchester, UK, pages 24–32. http://www.aclweb.org/anthology/W08-1205.

Mizumoto Tomoya, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pages 147–155. http://www.aclweb.org/anthology/I11-1017.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pages 173–180.

Yiming Wang, Longyue Wang, Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Yi Lu. 2014. Factored statistical machine translation for grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 83–90. http://www.aclweb.org/anthology/W14-1711.

Jonathan Weese, Juri Ganitkevitch, Chris Callison-Burch, Matt Post, and Adam Lopez. 2011. Joshua 3.0: Syntax-based machine translation with the Thrax grammar extractor. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pages 478–484.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification.

Transactions of the Association for Computational Linguistics 4:401–415.

Helen Yannakoudakis, Ted Briscoe, and Ben 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: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 180–189. http://www.aclweb.org/anthology/P11-1019.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 380–386. http://www.aclweb.org/anthology/N16-1042.