(Almost) Unsupervised Grammatical Error Correction using a Synthetic Comparable Corpus

Satoru Katsumata and **Mamoru Komachi**
Tokyo Metropolitan University
katsumata-satoru@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract

We introduce unsupervised techniques based on phrase-based statistical machine translation for grammatical error correction (GEC) trained on a pseudo learner corpus created by Google Translation. We verified our GEC system through experiments on a low resource track of the shared task at Building Educational Applications 2019 (BEA2019). As a result, we achieved an $F_{0.5}$ score of 28.31 points with the test data.

1 Introduction

Research on grammatical error correction (GEC) has gained considerable attention recently. Many studies treat GEC as a task that involves translation from a grammatically erroneous sentence (source-side) into a correct sentence (target-side) and thus, leverage methods based on machine translation (MT) for GEC. For instance, some GEC systems use large parallel corpora and synthetic data (Ge et al., 2018; Xie et al., 2018).

We introduce an unsupervised method based on MT for GEC that does not use parallel learner data. In particular, we use methods proposed by Marie and Fujita (2018), Artetxe et al. (2018b), and Lample et al. (2018). These methods are based on phrase-based statistical machine translation (SMT) and phrase table refinements. Forward refinement used by Marie and Fujita (2018) simply augments a learner corpus with automatic corrections. We also use forward refinement for improvement of phrase table.

Unsupervised MT techniques do not require a parallel but a comparable corpus as training data. Therefore, we use comparable translated texts using Google Translation as the source-side data. Specifically, we use News Crawl written in English as target-side data and News Crawl written in another language translated into English as source-side data.

We verified our GEC system through experiments for a low resource track of the shared task at Building Educational Applications 2019 (BEA2019). The experimental results show that our system achieved an $F_{0.5}$ score of 28.31 points.

2 Unsupervised GEC

Algorithm 1 shows the pseudocode for unsupervised GEC. This code is derived from Artetxe et al. (2018b). First, the cross-lingual phrase embeddings are acquired. Second, a phrase table is created based on these cross-lingual embeddings. Third, the phrase table is combined with a language model trained by monolingual data to initialize a phrase-based SMT system. Finally, the SMT system is updated through iterative forward-translation.

Cross-lingual embeddings First, n-gram embeddings were created on the source- and target-sides. Specifically, each monolingual embedding was created based on the source- and target-sides using a variant of skip-gram (Mikolov et al., 2013) for unigrams, bigrams, and trigrams with high frequency[1] in the monolingual data. Next, the monolingual embeddings were mapped onto a shared space to obtain cross-lingual embeddings. The self-learning method of Artetxe et al. (2018a) was used for unsupervised mapping.

Phrase table induction A phrase table was created based on the cross-lingual embeddings. In particular, this involved the creation of phrase translation models and lexical translation models.

The translation candidates were limited in the source-to-target phrase translation model $\phi(\overline{f}|\overline{e})$ for each source phrase $\overline{e}$ to its 100 nearest neighbor phrases $\overline{f}$ on the target-side. The score of

[1] We used the most frequent 200K unigrams, 400K bigrams, and 400K trigrams in the monolingual data.

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 134–138
Florence, Italy, August 2, 2019.

Algorithm 1 Unsupervised GEC

Require: language models of the target-side LM_t
Require: source training corpus C_s
Require: target training corpus C_t
Require: tuning data T
Require: iteration number N
Ensure: source-to-target phrase table $P_{s\to t}^{(N)}$
Ensure: source-to-target weights $W_{s\to t}^{(N)}$
1: $W_s^{emb} \leftarrow \text{TRAIN}(C_s)$
2: $W_t^{emb} \leftarrow \text{TRAIN}(C_t)$
3: $W_s^{cross\_emb}, W_t^{cross\_emb} \leftarrow \text{MAPPING}(W_s^{emb}, W_t^{emb})$
4: $P_{s\to t}^{(0)} \leftarrow \text{INITIALIZE}(W_s^{cross\_emb}, W_t^{cross\_emb})$
5: $W_{s\to t}^{(0)} \leftarrow \text{TUNE}(P_{s\to t}^{(0)}, LM_t, T)$
6: **for** $iter = 1, \ldots, N$ **do**
7: $\quad \text{synthetic\_data}_t$
8: $\quad\quad \leftarrow \text{DECODE}(P_{s\to t}^{(iter-1)}, LM_t, W_{s\to t}^{(iter-1)}, C_s)$
9: $\quad P_{s\to t}^{(iter)} \leftarrow \text{TRAIN}(C_s, \text{synthetic\_data}_t)$
10: $\quad W_{s\to t}^{(iter)} \leftarrow \text{TUNE}(P_{s\to t}^{(iter)}, LM_t, T)$

the phrase translation model was calculated based on the normalized cosine similarity between the source and target phrases.

$$\phi(\overline{f}|\overline{e}) = \frac{\exp(\cos(\overline{e}, \overline{f})/\tau)}{\sum_{\overline{f}'} \exp(\cos(\overline{e}, \overline{f}')/\tau)} \quad (1)$$

$\overline{f}'$ represents each phrase embedding on the target-side and τ is a temperature parameter that controls the confidence of prediction[2]. The backward phrase translation probability $\phi(\overline{e}|\overline{f})$ was determined in a similar manner.

The source-to-target lexical translation model $\text{lex}(\overline{f}|\overline{e})$ considers the word with the highest translation probability in a target phrase for each word in a source phrase. The score of the lexical translation model was calculated based on the product of respective phrase translation probabilities.

$$\text{lex}(\overline{f}|\overline{e}) = \prod_i \max\left(\epsilon, \max_j \phi\left(\overline{f}_i|\overline{e}_j\right)\right) \quad (2)$$

ϵ is a constant term for the case where no alignments are found. As in Artetxe et al. (2018b), the term was set to 0.001. The backward lexical translation probability $\text{lex}(\overline{e}|\overline{f})$ is calculated in a similar manner.

Refinement of SMT system The phrase table created is considered to include noisy phrase pairs. Therefore, we update the phrase table using an SMT system. The SMT system trained on synthetic data eliminates the noisy phrase pairs using language models trained on the target-side corpus. This process corresponds to lines 6—10 in Algorithm 1. The phrase table is refined with forward refinement (Marie and Fujita, 2018).

For forward refinement, target synthetic data were generated from the source monolingual data using the source-to-target phrase table $P_{s\to t}^{(0)}$ and target language model LM_t. A new phrase table $P_{s\to t}^{(1)}$ was then created with this target synthetic corpus. This operation was executed N times.

[2]As in Artetxe et al. (2018b), τ is estimated by maximizing the phrase translation probability between an embedding and the nearest embedding on the opposite side.

Corpus	Sent.	Learner
Fi News Crawl	1,904,880	No
En News Crawl	2,116,249	No
One-Billion	24,482,651	No
tuning data	2,191	Yes
dev data	2,193	Yes

Table 1: Data statics: train and dev data size.

Construction of a comparable corpus This unsupervised method is based on the assumption that the source and target corpora are comparable. In fact, Lample et al. (2018), Artetxe et al. (2018b) and Marie and Fujita (2018) use the News Crawl of source and target language as training data.

To make a comparable corpus for GEC, we use translated texts using Google Translation as the source-side data. Specifically, we use Finnish News Crawl translated into English as source-side. English News Crawl is used as the target-side as is. Finnish data is used because Finnish is not similar to English.

This translated data does not include misspelled words. To address these words, we use a spell checker as a preprocessing step before inference.

3 Experiment of low resource GEC

3.1 Experimental setting

Table 1 shows the training and development data size. Finnish News Crawl 2014—2015 translated into English was used as source training data and English News Crawl 2017 was used as target training data. To train the extra language model of the target-side (LM_t), we used training data of One Billion Word Benchmark (Chelba et al., 2014). We used `googletrans v2.4.0`[3] for Google Translation. This module did not work sometimes and thus, we obtained 2,122,714 trans-

[3]https://github.com/ssut/py-googletrans

Team	TP	FP	FN	P	R	$F_{0.5}$
UEDIN-MS	2,312	982	2,506	70.19	47.99	64.24
Kakao&Brain	2,412	1,413	2,797	63.06	46.30	58.80
LAIX	1,443	884	3,175	62.01	31.25	51.81
CAMB-CUED	1,814	1,450	2,956	55.58	38.03	50.88
UFAL, Charles University, Prague	1,245	1,222	2,993	50.47	29.38	44.13
Siteimprove	1,299	1,619	3,199	44.52	28.88	40.17
WebSpellChecker.com	2,363	3,719	3,031	38.85	43.81	39.75
TMU	1,638	4,314	3,486	27.52	31.97	28.31
Buffalo	446	1,243	3,556	26.41	11.14	20.73

Table 2: GEC results with test data.

lated sentences[4]. We sampled the 3,000,000 sentences from English News Crawl 2017 and excluded the sentences with more than 150 words for either source- and target-side data. Finally, the synthetic comparable corpus comprises processed News Crawl data listed in Table 1. The low resource track permitted to use W&I+LOCNESS (Bryant et al., 2019; Granger, 1998) development set, so we split it in half; tune data and dev data[5].

These data are tokenized by `spaCy v1.9.0`[6] and the `en_core_web_sm-1.2.0` model. We used moses truecaser for the training data; this truecaser model is learned from processed English News Crawl. We used byte-pair-encoding (Sennrich et al., 2016) learned from processed English News Crawl; the number of operations is 50K.

The implementation proposed by Artetxe et al. (2018b)[7] was modified to conduct the experiments. Specifically, some features were added; word-level Levenshtein distance, word-, and character-level edit operation, operation sequence model, (Durrani et al., 2013)[8] and 9-gram word class language model, similar to Grundkiewicz and Junczys-Dowmunt (2018) without sparse features. Word class language model was trained with One Billion Word Benchmark data; the number of classes is 200, and the word class was estimated with `fastText` (Bojanowski et al., 2017). The distortion feature was not used.

`Moses` (Koehn et al., 2007) was used to train the SMT system. `FastAlign` (Dyer et al., 2013) was used for word alignment and `KenLM` (Heafield, 2011) was used to train the 5-gram language model over each processed English News Crawl and One Billion Word Benchmark. MERT (Och, 2003) was used with the tuning data for `M^2 Scorer` (Dahlmeier and Ng, 2012). Synthetic sentence pairs with a [3, 80] sentence length were used at the refinement step. The number of iterations N was set to 5, and the embedding dimension was set to 300. We decided best iteration using the dev data and submitted the output of the best iteration model.

We used `pyspellchecker`[9] as a spell checker. This tool uses Levenshtein distance to obtain permutations within an edit distance of 2 over the words included in a word list. We made the word list from One Billion Word Benchmark and included words that occur more than five times.

We report precision, recall, and $F_{0.5}$ score based on the dev data and official test data. The output of dev data was evaluated using ERRANT scorer (Bryant et al., 2017) similarly to official test data.

3.2 Results

Table 2 shows the results of the GEC experiments with test data. The $F_{0.5}$ score for our system (TMU) is 28.31; this score is eighth among the nine teams. In particular, the number of false positives of our system is 4,314; this is the worst result of all.

4 Discussion

Table 3 shows the results of the dev data listed in Table 1. On the dev data, the system of iteration 1 is the best among all. According to the improvement of iteration from 0 to 1, it is confirmed that the refinement method works well. However, it is observed that the system is not improved after iteration 1. The source-side data is fixed, and target-side data is generated from the source-side for each iteration. Therefore, the quality of the

[4]Finnish News Crawl 2014—2015 have 6,360,479 sentences.

[5]Because W&I+LOCNESS data had four types of learner level, we split it so that each learner level is equal.

[6]https://github.com/explosion/spaCy

[7]https://github.com/artetxem/monoses

[8]Operation sequence model was used in refinement step.

[9]https://github.com/barrust/pyspellchecker

	iter	P	R	$F_{0.5}$
Unsupervised SMT	0	12.33	**16.13**	12.94
w/o spell check	1	**17.59**	14.63	**16.91**
	2	17.30	14.15	16.56
	3	16.04	14.17	15.63
	4	17.06	14.01	16.35
	5	15.88	13.88	15.44
spell check → SMT	1	20.58	18.04	20.01
SMT → spell check	1	19.42	16.86	18.85

Table 3: GEC results with dev data. The bold scores represent the best score without the spell checker.

	P	R	$F_{0.5}$
Top2			
SPELL	39.93	59.24	42.71
PUNCT	28.91	38.14	30.38
Bottom2			
NOUN	0.87	1.74	0.97
VERB	2.13	0.99	1.73

Table 4: Error types for which our best system corrected errors well or mostly did not correct on the dev data. Top2 denotes the top two errors, and Bottom2 denotes the lowest two errors in terms of the $F_{0.5}$[10].

source-side data is important for this refinement method. In this study, we use the automatically translated text as source-side data; thus, it is considered that the quality is not high and the refinement after iteration 1 does not work.

The results of Table 3 confirm that the spell checker works well. We also investigate the importance of the order; SMT or spell check, which is suitable for the first system for a better result? As a result, it is better to use the SMT system after using the spell checker. That is because the source-side data does not include the misspelled words as mentioned above.

Table 4 shows the error types that our system corrected well or mostly did not correct on the dev data. SPELL means the misspell errors; the correction of these errors depends only on the spell checker. PUNCT means the errors about the punctuation; e.g., 'Unfortunately when we...→ Unfortunately, when we...'. It is considered that our system can correct errors such as these owing to the n-gram co-occurrence knowledge derived from the language models.

In contrast, our system struggled to correct content word errors. For example, NOUN includes an error like this; 'way → means' and VERB includes an error like this; 'watch → see'. It is considered that our system is mostly not able to correct the errors regarding word usage based on the context because the phrase table was still noisy. Although we observed some usage error examples of 'watch' in the synthetic source data, our model was not able to replace 'watch' to 'see' based on the context.

5 Related Work

Unsupervised Machine Translation Studies on unsupervised methods have been conducted for both NMT (Lample et al., 2018; Marie and Fujita, 2018) and SMT (Artetxe et al., 2018b). In this study, we apply the USMT method of Artetxe et al. (2018b) and Marie and Fujita (2018) to GEC. The UNMT method (Lample et al., 2018) was ineffective under the GEC setting in our preliminary experiments.

GEC with NMT/SMT Several studies that introduce sequence-to-sequence models in GEC heavily rely on large amounts of training data. Ge et al. (2018), who presented state-of-the-art results in GEC, proposed a supervised NMT method trained on corpora of a total 5.4 M sentence pairs. We mainly use the monolingual corpus because the low resource track does not permit the use of the learner corpora.

Despite the success of NMT, many studies on GEC traditionally use SMT (Susanto et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014). These studies apply an off-the-shelf SMT toolkit, Moses, to GEC. Junczys-Dowmunt and Grundkiewicz (2014) claimed that the SMT system optimized for BLEU learns to not change the source sentence. Instead of BLEU, they proposed tuning an SMT system using the M^2 score with annotated development data. In this study, we also tune the weights with an $F_{0.5}$ score measured by the M^2 scorer because the official score is an $F_{0.5}$ score.

6 Conclusion

In this paper, we described our GEC system for the low resource track of the shared task at BEA2019. We introduced an unsupervised approach based on SMT for GEC. This track prohibited the use of learner data as training data, so we created a synthetic comparable corpus using Google Translation. The experimental results demonstrate that

[10]We investigate the frequent error types; the errors occur more than one hundred times in the dev data.

our system achieved an $F_{0.5}$ score of 28.31 points with the test data.

Acknowledgments

This work was partially supported by JSPS Grant-in-Aid for Scientific Research (C) Grant Number JP19K12099.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proc. of ACL*, pages 789–798.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proc. of EMNLP*, pages 3632–3642.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proc. of BEA*.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proc. of ACL*, pages 793–805.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Proc. of INTERSPEECH*, pages 2635–2639.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proc. of NAACL-HLT*, pages 568–572.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov models over minimal translation units help phrase-based smt? In *Proc. of ACL*, pages 399–405.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proc. of NAACL-HLT*, pages 644–648.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proc. of NAACL-HLT*, pages 284–290.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proc. of WMT*, pages 187–197.

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 *Proc. of CoNLL*, pages 25–33.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL Demo Sessions*, pages 177–180.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proc. of EMNLP*, pages 5039–5049.

Benjamin Marie and Atsushi Fujita. 2018. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *arXiv preprint arXiv:1810.12703*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR Workshop*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pages 1715–1725.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proc. of EMNLP*, pages 951–962.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proc. of NAACL-HLT*, pages 619–628.