The AIP-Tohoku System at the BEA-2019 Shared Task

Hiroki Asano[12]*, **Masato Mita**[21], **Tomoya Mizumoto**[21]† and **Jun Suzuki**[12]
[1] Graduate School of Information Sciences, Tohoku University
[2] RIKEN Center for Advanced Intelligence Project
asano@ecei.tohoku.ac.jp, masato.mita@riken.jp
tomoya.mizumoto@riken.jp, jun.suzuki@ecei.tohoku.ac.jp

Abstract

We introduce the AIP-Tohoku grammatical error correction (GEC) system for the BEA-2019 shared task in Track 1 (Restricted Track) and Track 2 (Unrestricted Track) using the same system architecture. Our system comprises two key components: error generation and sentence-level error detection. In particular, GEC with sentence-level grammatical error detection is a novel and versatile approach, and we experimentally demonstrate that it significantly improves the precision of the base model. Our system is ranked 9th in Track 1 and 2nd in Track 2.

1 Introduction

As part of the BEA-2019 shared task, we participated in Track 1 (Restricted Track) and Track 2 (Unrestricted Track). We utilized the Transformer (Vaswani et al., 2017) architecture as a base GEC model for machine translation systems as it has become a state-of-the-art approach for grammatical error correction (GEC).

In our system, the error correction model collaborates with a sentence-level error detection model. In GEC, $F_{0.5}$ is used for evaluation because precision is more important than recall. To improve the precision score on the test set, our system corrected the input sentences by detecting errors using a sentence-level error detection model (which we denote as SED). We applied the bidirectional encoder representations from transformers (BERT) model (Devlin et al., 2018) for sentence-level error detection. In order to improve the performance of SED, we propose an SED model taking the learner's proficiency into account. To the best of our knowledge, this is the first study that has combined GEC with SED.

Because grammatical correctness is required for output sentences in GEC, the target side of parallel training corpora should not contain noisy sentences. Our correction model is trained to correct sentence pairs, which were identified by our sentence-level grammatical error detection model. We call this data cleaning process BERT-Cleaning.

For Track 1, similar to back-translation (Sennrich et al., 2016b; Edunov et al., 2018), we augmented the parallel training corpus with errors generated from monolingual data. After addition of the generated data and SED process, the $F_{0.5}$ score on the base model improved.

For Track 2, we used the EF-Cambridge Open Language Database (EFCAMDAT) (Geertzen et al., 2013) and non-public Lang-8 as the external language learner corpus.

2 Related Work

2.1 Error Detection

The field of grammatical error detection (GED) has a long history. Many previous studies have treated GED as a token-level binary classification task (Tetreault and Chodorow, 2008; Han et al., 2006; Chodorow et al., 2012; Rei and Yannakoudakis, 2016; Rei et al., 2016; Rei, 2017). Kaneko et al. (2017) improved grammatical error detection by learning word embeddings that consider grammaticality and error patterns. Yannakoudakis et al. (2017) propose an approach to N-best list re-ranking using neural sequence-labelling models.

While many studies in GED focus on token-level error detection, there are studies that perform sentence-level binary classification of sentences that need some editing (Han et al., 2006; Tetreault and Chodorow, 2008; Chodorow et al., 2012;

[*] Current affiliation: Yahoo Japan Corporation, hiroasan@yahoo-corp.jp

[†] Current affiliation: Future Corporation, mizumoto.tomoya.mh7@is.naist.jp

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 176–182
Florence, Italy, August 2, 2019.

Schmaltz et al., 2016). Compared with token-level grammatical error correction, sentence-level grammatical error correction is a simple problem setting because there is no need to identify the location of errors.

2.2 Error Generation

In the field of machine translation, back-translation is an effective method for neural machine translation systems (Sennrich et al., 2016b; Imamura et al., 2018). Edunov et al. (2018) reported that back-translation obtained via sampling or noised beam outputs is effective for neural machine translation systems.

Recently, back-translation has been applied to grammatical error detection and correction. Rei et al. (2017) proposed artificial error generation with statistical machine translation and syntactic patterns for error detection. Kasewa et al. (2018) constructed synthetic samples using a seq2seq neural model with greedy search and temperature sampling for error detection. Xie et al. (2018) proposed certain noising methods for error generation, and Ge et al. (2018) proposed back-boost learning using fluency scores.

3 System Architecture

3.1 Base Correction Model

We used Transformer, the self-attention-based translation model, as a base GEC system (Vaswani et al., 2017). Some previous studies used Transformer to achieve high performance (Junczys-Dowmunt et al., 2018; Zhao et al., 2019).

3.2 Sentence-level Error Detection

3.2.1 Motivation

The sentence-level error detection (SED) module is one of the key components of our system, with the goal of detecting sentences with grammatical errors. The aim of introducing SED is to reduce false positive by passing only sentences that contain errors to the GEC model. We calculated the rate of a sentence that changes in the W&I+LOCNESS development set and found it to be 64.34%, i.e., almost 35% of the sentences did not require corrections.

3.2.2 Base Model

We built a base SED model using BERT (Devlin et al., 2018), which is a straightforward extension of sequence classification tasks such as CoLA (Warstadt et al., 2018) and SST-2 (Socher et al., 2013). For setting up a training set for the base SED model, we preprocessed it to obtain binary labeled data (e.g., 0 for correct and 1 for incorrect, respectively).

3.2.3 Proposed Model

Figure 1 shows the architecture of our proposed SED model. The key ideas of our proposed model are as follows:

- There is a correlation between the error rate and the learner's level of proficiency.
- The performance of SED can be improved by fine-tuning the model according to the learners proficiency.

The first idea is based on the following observation on the W&I+LOCNESS development set: Looking at the word error rate (WER) across three different CEFR levels: A (beginner), B (intermediate), C (advanced), we can confirm that 19.49% for level A, 13.18% for level B, and 6.04% for level C. The second idea comes from previous studies on GEC (Junczys-Dowmunt and Grundkiewicz, 2016; Junczys-Dowmunt et al., 2018). They showed that better results can be achieved if the error rate of the training data is adapted to the error rate of the development set, which is called error adaptation.

Let N and M denote the total number of source words and sentences in a corpus, respectively. WER is defined as follows:

$$\text{WER} = \frac{\sum_{m=1}^{M} d(X^m, Y^m)}{\sum_{m=1}^{M} N^m}$$

where X^m denotes each source sentence, Y^m denotes each corrected sentence, and $d(X^m, Y^m)$ denotes the edit distance between X^m and Y^m.

Based on the above ideas, our SED model is developed in two steps:

1. Building Proficiency Prediction Module (PPM): The PPM predicts the proficiency of the learner who wrote a given sentence. Based on the above key ideas, we employed a multi-task learning approach in which the model estimates the learner's proficiency and performs sentence-level error detection simultaneously (PP&SED in Figure1), trained on labelled data obtained by simply conjoining the SED label with PP label (e.g., 1_A).

We confirmed that the PP&SED outperforms the vanilla PP by a large margin of up to 7.8 points at accuracy (from 42.2 to 50.0).

2. Fine-tuning SED model: After dividing the given text by proficiency based on the label estimated by the PPM, the SED model is fine-tuned for each level of proficiency.

Then, the SED module performs sentence-level binary classification of sentences that need editing. Table 1 shows the performance of SED on our dev set. Here, we split the official development set into test/dev set for our experiments. Our proposed SED model achieved a significant improvement both in precision and recall, by considering learner proficiency.

	Prec.	Rec.	F
Base Model	88.5	79.8	83.9
Proposed Model	**91.3**	**95.6**	**93.4**

Table 1: Performance of sent-level error detection (SED).

3.3 Error Generation

Our error generation system follows the system developed by Edunov et al. (2018). A target-to-source model is trained, and back-translation is applied to monolingual data to generate pseudo-parallel data via sampling from the distribution of the target-to-source model.

4 Experiment

4.1 Experimental Setting

We will now describe the training data and tools used to train our model.

4.1.1 Tools

We used the Transformer implemented in Fairseq[1] (Ott et al., 2019) as our GEC model. For the Transformer, we used a token embedding size of dimension 512. The hidden size is set to 512, and the filter size is set to 2048. The multi-head attention has eight individual attention heads, whereas the encoder and decoder have six layers. We use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We use inverse squared root decay. We set the dropout to 0.3. Rather than using words directly, we used byte pair encoding (BPE) (Sennrich et al., 2016a), and each of the source and target vocabularies comprises 30K elements, which are the most frequent BPE tokens.

For building the sentence-level error detection model, we employed the model based on BERT, especially for the sequence-level tasks as described in Section 3.2. Thus, we used the PyTorch implementations for Googles BERT model [2].

For building the error generation model, we used a 7-layer convolutional seq2seq model implemented in Fairseq (Gehring et al., 2017; Chollampatt and Ng, 2018). As Chollampatt and Ng (2018), both source and target embeddings are of 500 dimensions. Each of the source and target vocabularies comprises the 30K most frequent BPE tokens. The hidden size of encoders and decoders is 1,024 with a convolution window width of 3. The output of each encoder and decoder layer is 1,024 dimensions. We set the dropout rate to 0.3. The parameters are optimized using the Nesterov Accelerated Gradient (Sutskever et al., 2013) optimizer with a momentum value of 0.99. We set the initial learning rate to 0.25, using early stopping.

For evaluating the system outputs, the ERRANT (Bryant et al., 2017) is used as a scorer. In this study, all the results shown are "span-based correction F0.5".

4.1.2 Dataset for Track-1

For training our transformer-based GEC system, we used the BEA-2019 workshop official data: the First Certificate in English corpus (FCE) (Yannakoudakis et al., 2011), the Lang-8 Corpus of Learner English (Lang-8) (Mizumoto et al., 2011; Tajiri et al., 2012), the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), and W&I+LOCNESS (Bryant et al., 2019; Granger, 1998). Our pre-processing for training data is the same as that reported previously (Chollampatt and Ng, 2018). As the result, we obtained 564,565 sentence pairs.

In generating erroneous sentences, we used Simple Wikipedia and essay scoring data sets (i.e., International Corpus of Learner English (Granger et al., 2009), and International Corpus Network of Asian Learners of English (Ishikawa, 2013), the Automated Student Assessment Prize dataset[3], ETS Corpus of Non-Native English (TOEFL

[1] https://github.com/pytorch/fairseq

[2] https://github.com/huggingface/pytorch-pretrained-BERT

[3] https://www.kaggle.com/c/asap-sas

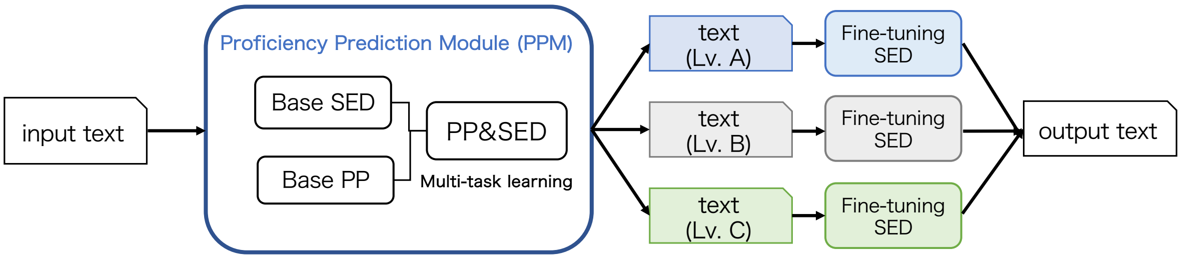

Figure 1: Architecture of proposed sentence-level error detection (SED) model. Native-level (N) is combined with C-level data.

	Prec.	Rec.	F0.5
Base	61.97	42.11	56.63
Base+SED	65.45	38.04	57.20
Base+GenData	64.57	**46.40**	59.88
Base+SED+GenData	**68.62**	42.16	**60.97**

Table 2: Track1 results

11) (Blanchard et al., 2013). With respect to Simple Wikipedia, we ignored sentences that were longer than 60 tokens. To remove erroneous sentences, we applied BERT-Cleaning to the essay scoring data sets. After BERT-Cleaning and preprocessing (Chollampatt and Ng, 2018), we obtained 1,426,354 sentence pairs by error generation.

4.1.3 External Dataset for Track-2

We used EFCAMDAT (Geertzen et al., 2013) and non-public Lang-8 as the external language learner corpus. The EFCAMDAT is constructed by the Department of Theoretical and Applied Linguistics at the University of Cambridge. Lo et al. (2018) were the first the researchers to use the EFCAMDAT for the GEC task. However, the system trained with the EFCAMDAT gave lower performance than the system trained with the Lang-8 Corpus. One of the causes of the lower performance is that many errors are found in the EFCAMDAT corrected sentences. Thus, we applied BERT-Cleaning to the EFCAMDAT to remove the erroneous sentences. Consequently, the number of sentence pairs of EFCAMDAT was reduced from 1,157,339 to 760,393. Finally, we used 7,739,577 sentence pairs (non-public Lang-8 + Cleand EFCAMDAT) by using pre-processing (Chollampatt and Ng, 2018) as the additional training data.

	Prec.	Rec.	F0.5
Track1	68.62	42.16	60.97
Track1 + AddData	**70.60**	**51.03**	**65.57**

Table 3: Track2 results

4.2 Results on Track-1

Table 2 shows the results of our systems, ensemble decoding of five independently trained models. We compared the following four systems: (1) Base (Transformer-based GEC system), (2) Base plus sentence error detection (Base+SED) described in section 3.2, (3) Base plus generated data (Base+GenData), and (4) Base plus sentence error detection and generated data (Base+SED+GenData).

Note that our system, which was composed of both SED and GenData, achieved a 60.97 $F_{0.5}$ score. Our proposed methods, the SED, and the GenData were effective for improving GEC performance. Especially, the SED is effective for a precision score, which improved from 61.97 to 65.45 (+3.48). However, the recall dropped from 42.11 to 38.04 (4.07). Nevertheless, the GenData improved both recall (from 42.11 to 46.40) and precision (from 61.97 to 64.57).

4.3 Results on Track-2

Table 3 shows the results of the model trained with additional data (Track1+AddData). The additional data improve precision and recall, and notably give a large increase in recall (improved from 42.16 to 51.03).

5 Conclusion

We described our system for the BEA-2019 Shared Task. Our system has two key components: error generation and sentence-level error detection. We input grammatically incorrect sentences predicted by the sentence-level error detection model into our correction model. Sentence-level grammatical error detection is a novel approach to grammatical error correction, and we have shown that it can significantly improve performance. Our system ranked 9th in Track-1 and 2nd in Track-2.

References

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A Corpus of Non-Native English. Technical report.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*.

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*, pages 793–805.

Martin Chodorow, Markus Dickinson, Ross Israel, and Joel Tetreault. 2012. Problems in Evaluating Grammatical Error Detection Systems. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 611–628.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5755–5762.

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 8th Workshop on Building Educational Applications Using NLP*, pages 22–31.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency Boost Learning and Inference for Neural Grammatical Error Correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1055–1065.

Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. 2013. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In *Proceedings of the 31st Second Language Research Forum. Somerville, MA: Cascadilla Proceedings Project*, pages 240–254.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Sylviane Granger. 1998. The Computer Learner Corpus: A Versatile New Source of Data for SLA Research. In *Learner English on Computer*, pages 3–18.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *International Corpus of Learner English v2*. Presses universitaires de Louvain, Louvain-la-Neuve.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering*, 12(2):115–129.

Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. 2018. Enhancement of Encoder and Attention Using Target Monolingual Corpora in Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 55–63.

Shin'ichro Ishikawa. 2013. The ICNALE and Sophisticated Contrastive Interlanguage Analysis of Asian learners of English. *Learner Corpus Studies in Asia and the World*, 1:91–118.

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*, pages 1546–1556.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 595–606.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical Error Detection Using Error- and Grammaticality-Specific Word Embeddings. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 40–48.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection.

In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983.

Yu-Chun Lo, Jhih-Jie Chen, Ching-Yu Yang, and Jason S. Chang. 2018. Cool English: A Grammatical Error Correction System Based on Large Learner Corpora. In *Proceedings of the 26th International Conference on Computational Linguistics: Demo*, pages 82–85.

Tomoya Mizumoto, 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 the 5th International Joint Conference on Natural Language Processing*, pages 147–155.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demo*, pages 48–53.

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 2121–2130.

Marek Rei, Gamal Crichton, and Sampo Pyysalo. 2016. Attending to Characters in Neural Sequence Labeling Models. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 309–318.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial Error Generation with Machine Translation and Syntactic Patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292.

Marek Rei and Helen Yannakoudakis. 2016. Compositional Sequence Labeling Models for Error Detection in Learner Writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1181–1191.

Allen Schmaltz, Yoon Kim, Alexander M. Rush, and Stuart Shieber. 2016. Sentence-Level Grammatical Error Identification as Sequence-to-Sequence Correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 242–251.

Rico Sennrich, Barry Haddow, and Birc Alexandra. 2016a. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147.

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*, pages 198–202.

Joel R Tetreault and Martin Chodorow. 2008. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 865–872.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural Network Acceptability Judgments. *CoRR*, abs/1805.12471.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 619–628.

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*, pages 180–189.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural Sequence-Labelling Models for Grammatical Error Correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 156–165.