A Simple Recipe for Multilingual Grammatical Error Correction

Sascha Rothe
Google
rothe@google.com

Jonathan Mallinson
Google
jonmall@google.com

Eric Malmi
Google
emalmi@google.com

Sebastian Krause
Google
bastik@google.com

Aliaksei Severyn
Google
severyn@google.com

Abstract

This paper presents a simple recipe to train state-of-the-art multilingual Grammatical Error Correction (GEC) models. We achieve this by first proposing a language-agnostic method to generate a large number of synthetic examples. The second ingredient is to use large-scale multilingual language models (up to 11B parameters). Once fine-tuned on language-specific supervised sets we surpass the previous state-of-the-art results on GEC benchmarks in four languages: English, Czech, German and Russian. Having established a new set of baselines for GEC, we make our results easily reproducible and accessible by releasing a CLANG-8 dataset.[1] It is produced by using our best model, which we call gT5, to clean the targets of a widely used yet noisy LANG-8 dataset. CLANG-8 greatly simplifies typical GEC training pipelines composed of multiple fine-tuning stages – we demonstrate that performing a single fine-tuning step on CLANG-8 with the off-the-shelf language models yields further accuracy improvements over an already top-performing gT5 model for English.

1 Introduction

Grammatical Error Correction (GEC) is the task of correcting grammatical and other related errors in text. It has been the subject of several modeling efforts in recent years due to its ability to improve grammaticality and readability of user generated texts. This is of particular importance to non-native speakers, children, and individuals with language impairments, who may be more prone to producing texts with grammatical errors.

Modern approaches often view the GEC task as monolingual text-to-text rewriting (Náplava and Straka, 2019; Katsumata and Komachi, 2020; Grundkiewicz et al., 2019) and employ encoder-decoder neural architectures (Sutskever et al., 2014; Bahdanau et al., 2015). These methods typically require large training sets to work well (Malmi et al., 2019) which are scarce especially for languages other than English. One of the largest and most widely used datasets for GEC is the LANG-8 Learner Corpus, which covers 80 languages and has been created by language learners correcting each other's texts.[2] However, the distribution of languages is very skewed, with Japanese and English being the most prevalent languages with over a million ungrammatical-grammatical sentence pairs each, while only ten languages have more than 10,000 sentence pairs each. Additionally, given the uncontrolled nature of the data collection, many of the examples contain unnecessary paraphrasing and erroneous or incomplete corrections.

Limited amounts of suitable training data has led to multiple approaches that propose to generate synthetic training data for GEC (Madnani et al., 2012; Grundkiewicz and Junczys-Dowmunt, 2014; Grundkiewicz et al., 2019; Lichtarge et al., 2019; Awasthi et al., 2019). Although using synthetic data as the first fine-tuning step has been shown to improve model accuracy, it introduces practical challenges that make the development and fair comparison of GEC models challenging: (i) the synthetic methods often require language-specific tuning (e.g. language-specific hyperparameters and spelling dictionaries (Náplava and Straka, 2019)), and; (ii) due to the inability of synthetic data to capture the complete error distribution of the target eval sets, the final model is obtained by following a multi-stage fine-tuning process (Lichtarge et al., 2019, 2020; Omelianchuk et al., 2020). Because of this, carefully picking the learning rates and number of training steps for each of the fine-tuning

[1] CLANG-8 can be found at https://github.com/google-research-datasets/clang8

[2] Corpus collected from https://lang-8.com/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 702–707
August 1–6, 2021.

stages is required, making it difficult to replicate and build on top of previous best reported models.

The ideas of leveraging self-supervised pre-training and increasing the model size have yielded significant improvements on numerous seq2seq tasks in recent years (Raffel et al., 2019; Xue et al., 2020; Lewis et al., 2020; Song et al., 2019; Chan et al., 2019; Rothe et al., 2020), but these approaches have been applied to GEC to only a limited extent.

In this paper we adopt the mT5 (Xue et al., 2020) as our base model which has already been pre-trained on a corpus covering 101 languages. To adapt the model to the GEC task, we design a fully unsupervised language-agnostic pre-training objective that mimics corrections typically contained in labeled data. We generate synthetic training data by automatically corrupting grammatical sentences, but in contrast to the previous state-of-the-art by Náplava and Straka (2019) for low-resources languages, we use our synthetic pre-training to train a single model on all 101 languages, employing no language-specific priors to remain fully language-agnostic. After pre-training we further fine-tune our model on supervised GEC data for available languages (with data conditions ranging from millions to tens of thousands). Additionally, we explore the effect of scaling up the model size from 60M to 11B parameters. We surpass the previous state-of-the-art results on four evaluated languages: English, Czech, German and Russian.

Fine-tuning and running inference with our largest and most accurate models require multi-GPU/TPU infrastructure. To make the results of our research widely accessible we release a CLANG-8 dataset obtained by using our largest gT5 model to clean up the targets of the frequently used yet noisy LANG-8 dataset. We show that off-the-shelf variants of T5 (Raffel et al., 2019) when fine-tuned only on CLANG-8, outperform those models trained on the original LANG-8 data with and w/o additional fine-tuning data, thus simplifying the complex multi-stage process of training GEC models. Thus CLANG-8 not only allows others to easily train highly competitive GEC models, but it also greatly simplifies GEC training pipeline, basically reducing a multi-step fine-tuning processs to a single fine-tuning step.

Our contributions in this paper are three-fold: (1) We show that a simple language-agnostic pre-training objective can achieve state-of-the-art GEC results when models are scaled up in size; (2) We show the effect model size has on GEC, and; (3) We release a large multilingual GEC dataset based on Lang-8, which allows for state-of-the-art results without additional fine-tuning steps, thus significantly simplifying the training setup.

2 Model

Our model builds on top of mT5 (Xue et al., 2020) a multilingual version of T5 (Raffel et al., 2019) – a Transformer encoder-decoder model which has been shown to achieve state-of-the-art results on a wide range of NLG tasks. mT5 comes in different sizes, however for this work we use *base* (600M parameters) and *xxl* (13B parameters).

2.1 mT5 Pre-training

mT5 has been pre-trained on mC4 corpus, a subset of Common Crawl, covering 101 languages and composed of about 50 billion documents. For details on mC4, we refer the reader to the original paper (Xue et al., 2020). The pre-training objective is based on a span-prediction task, an adaptation of masked-language objective for autoregressive seq2seq models. An example of span prediction:

Input: `A Simple` [x] `Multilingual Grammatical Error` [y]
Target: [x] `Recipe for` [y] `Correction`

All mT5 models were trained for 1M steps on batches of 1024 input sequences with a maximum sequence length of 1024, corresponding to roughly 1T seen tokens. For all of our experiments we use the publicly available mT5 and T5 checkpoints (Section 4 only).

2.2 GEC Pre-training

The span-prediction objective of mT5 does not enable the model to perform GEC without further fine-tuning, as the span-prediction task uses special tokens to indicate where text should be inserted. Another limiting constraint is that mT5 has been trained on paragraphs, not sentences. We therefore split all paragraphs in mC4 corpus into sentences. We corrupt each sentence using a combination of the following operations: a) drop spans of tokens b) swap tokens c) drop spans of characters d) swap characters e) insert characters[3] f) lower-case a word g) upper-case the first

[3]We insert characters from the same passage, thus avoiding to insert character from a different alphabet.

character of a word. An example pair of an original sentence and its corrupted version looks as follows:

Input: `Simple recipe for Multingual Grammatical Correction Error`
Target: `A Simple Recipe for Multilingual Grammatical Error Correction`

We leave about 2% of examples uncorrupted, so the model learns that inputs can also be grammatical. We refrain from using more sophisticated text corruption methods, as these methods would be hard to apply to all 101 languages. For example, Náplava and Straka (2019) perform word substitutions with the entries from ASpell[4] which in turn makes the generation of synthetic data language-specific. Pre-training with this unsupervised objective is done on all languages in the mC4 corpus and not limited to the languages evaluated in this paper.

3 gT5: Large Multilingual GEC Model

Fine-tuning datasets. For English, we fine-tune our pre-trained models on the FCE (Yannakoudakis et al., 2011) and W&I (Bryant et al., 2019a) corpora. For Czech, German, and Russian, we use the AKCES-GEC (Náplava and Straka, 2019), Falko-MERLIN (Boyd, 2018), and RULEC-GEC (Rozovskaya and Roth, 2019) datasets, respectively. Table 1 reports statistics of datasets available for different languages.

lang	Corpus	Train	Dev	Test
EN	FCE, W&I	59,941		
EN	CoNLL-13/-14		1,379	1,312
EN	BEA			4,477
CS	AKCES-GEC	42,210	2,485	2,676
DE	Falko-MERLIN	19,237	2,503	2,337
RU	RULEC-GEC	4,980	2,500	5,000

Table 1: The size of the datasets used to fine-tune gT5.

Training Regime. We experimented with several training setups. All of them build on the mT5 pre-trained models (Section 2.1). We experimented with a) mixing GEC pre-training data (Section 2.2) with fine-tuning data (Section 3), b) mixing pre-training and finetuning examples but annotating them with different prefixes, and c) first using GEC pre-training until convergence and then fine-tuning. While c) is the most computationally expensive approach, it also gave us the best results. GEC pre-training as well as finetuning uses a constant learning rate of 0.001. Pre-training is done until convergence and fine-tuning until exact match accuracy on the development set degrades, which happens after 200 steps or 800k seen examples or 7 epochs.

[4] http://aspell.net

Models	CoNLL-14	BEA test	Czech	German	Russian
Omelianchuk et al.*	66.5	**73.6**	-	-	-
Lichtarge et al.*	**66.8**	73.0	-	-	-
Náplava and Straka	63.40	69.00	80.17	73.71	50.20
Katsumata and Komachi*	63.00	66.10	73.52	68.86	44.36
gT5 base	54.10	60.2	71.88	69.21	26.24
gT5 xxl	65.65	69.83	**83.15**	**75.96**	**51.62**

Table 2: $F_{0.5}$ Scores. Models denoted with * are ensemble models. We used the M^2 scorer for CoNLL-14, Russian, Czech and German, and the ERRANT scorer (Bryant et al., 2019b) for BEA test.

Results. For English, we evaluate on standard benchmarks from CoNLL-14 and the BEA test (Bryant et al., 2019a), while we use CoNLL-13 as the development set (Table 1). For other languages we use the test and development sets associated with their training data. Table 2 shows the results for all languages. We first see that the base model size is inferior to the current state-of-the-art models. This is expected as the model capacity is not enough to cover all 101 languages. We therefore use a larger xxl (11B) model, which produces new state-of-the-art results on all languages except for English. When looking at the development set performance for English, we observed that it had a high variance and the training was over-fitting very quickly. This suggests that train and dev/test set domains are not well aligned for English. In the following Section 4 we further refine our approach, also achieving state-of-the-art results for English.

4 cLang-8: Cleaned Lang-8 Corpus

To be able to distill the knowledge learned by gT5 xxl into smaller, more practical models, we create and release cLang-8, a cleaned version of the popular Lang-8 corpus. As discussed earlier, Lang-8 is a large corpus of texts written by language learners and user-annotated corrections to these texts. However, corrected texts frequently contain unnecessary paraphrasing and erroneous or incomplete corrections – phenomena that hurt the performance of a GEC model trained on this data. For instance, the following source–target pair

	LR	WER	Sub	Del	Ins
LANG-8	98%	15.46	8.85	2.41	4.19
CLANG-8	98%	10.11	5.85	1.35	2.92
CLANG-8-S	99%	01.22	0.64	0.00	0.58

Table 3: Dataset statistics of English LANG-8 and CLANG-8, including sequence **L**ength **R**atio between the source and the target, **W**ord **E**rror **R**ate, which is comprised of **Sub**stitutions, **Del**etions, and **Ins**ertions.

is taken from LANG-8: *"It is cloudy or rainy recently ."* → *"It is It 's been cloudy or and rainy recently ."*

We experiment with two approaches for cleaning the data. First, to create CLANG-8, we generate new targets for LANG-8, disregarding the original targets. We tried using both the unsupervised model, which was trained using the GEC pretraining objective (Section 2.2) and the supervised model (gT5 xxl) (Section 3), but the former did not yield comparable results, so all reported numbers use the supervised model. Second, to create CLANG-8-S, we used the unsupervised and the supervised models to score the original targets, disregarding the lowest scoring 20%, 50%, 70%, or 90% targets. Disregarding 50% was the best performing setup and there was not a significant difference between the supervised and unsupervised model. We therefore report numbers using the unsupervised model disregarding the worst 50% of the targets. Table 3 shows that CLANG-8 moderately reduces the Word Error Rate (WER) between the source and target, with deletions receiving the largest relative reduction, which may suggest that less information from the source sentence is removed. In contrast CLANG-8-S has a significantly lower WER, indicating that the unsupervised model has only kept corrections which are close to the source sentence.

Experiments. To evaluate the effect cleaning LANG-8 has for English, we train two distinct models on this data: T5 (Raffel et al., 2019), a monolingual sequence-to-sequence model, and FELIX (Mallinson et al., 2020), a non-auto-regressive text-editing model.[5] We also tried fine-tuning these models on BEA (i.e. FCE and W&I) after fine-tuning them on CLANG-8, but this did not further improve the scores but slightly decreased them, e.g. 0.43 absolute decrease for BEA test when using T5 base. This can be explained by the fact that the model used to clean the target texts has already been trained on BEA. This suggests that the typical GEC training pipeline where a model is first fine-tuned on LANG-8 and then on BEA can be both simplified and made more accurate by only fine-tuning on CLANG-8.

Finally, we train mT5 models on the German and Russian portions of the CLANG-8 dataset and evaluate these models on the test sets from Table 1.

[5]The FELIXINSERT variant which we use does not employ re-ordering.

Model	#params	Training Data	CoNLL-14	BEA test
SOTA			66.8	73.6
gT5 xxl			65.65	69.83
FELIX	220M	LANG-8	41.63	30.54
FELIX	220M	LANG-8 + BEA	48.75	48.80
FELIX	220M	CLANG-8	**58.21**	**59.05**
T5 base	220M	LANG-8	52.77	59.14
T5 base	220M	LANG-8 + BEA	60.61	67.12
T5 base	220M	CLANG-8	**65.13**	**69.38**
T5 base	220M	CLANG-8-S	58.70	59.95
T5 small	60M	CLANG-8	60.70	65.01
T5 base	220M	CLANG-8	65.13	69.38
T5 large	770M	CLANG-8	66.10	72.06
T5 xl	3B	CLANG-8	67.75	73.92
T5 xxl	11B	CLANG-8	**68.87**	**75.88**

Table 4: $F_{0.5}$ scores on CoNLL-14 and BEA test. Block two and three compare different training data. The last block compares different model sizes for T5.

	LANG-8		CLANG-8	
	base	xxl	base	xxl
PUNCT	68.27	**78.75**	75.51	76.31
DET	63.84	77.31	79.04	**83.88**
PREP	57.09	72.54	74.67	**79.79**
ORTH	72.77	**76.86**	69.23	71.39
SPELL	74.38	84.64	85.83	**88.29**

Table 5: BEA test scores for the top five error types. Bold scores represent the best score for each error type.

Results & Analysis. The results for CoNLL-14 and BEA test benchmarks can be seen in Table 4. For both models and both test datasets, CLANG-8 improves the $F_{0.5}$ score compared to using the original LANG-8 corpus. While CLANG-8-S performs significantly worse than CLANG-8, it still improves over LANG-8. In terms of model size, larger models are consistently better then their smaller siblings. This is even true when comparing xl and xxl, suggesting that there might still be headroom by using models larger than xxl.

In Table 5 we compare error types made on BEA

Model	#params	Training Data	German	Russian
SOTA			73.71	50.20
gT5 xxl			**75.96**	**51.62**
mT5 small	300M	CLANG-8	61.78	17.80
mT5 base	580M	CLANG-8	67.19	25.20
mT5 large	1.2B	CLANG-8	70.14	27.55
mT5 xl	3.7B	CLANG-8	72.59	39.44
mT5 xxl	13B	CLANG-8	74.83	43.52

Table 6: $F_{0.5}$ scores on German and Russian.

test for T5 base and T5 xxl, trained on either LANG-8 or CLANG-8. We see that for both data conditions increasing the model size leads to an increase in performance. Comparing CLANG-8 and LANG-8, shows that CLANG-8 improves on all error types apart from orthographic (ORTH) and punctuation (PUNCT).

In Table 6, we evaluate mT5 trained on the German and Russian portions of the CLANG-8 dataset, which contain 114K and 45K training examples, respectively. We see that for both languages performance increases with the model size, with no indication of slowing, suggesting further headroom for improvement. For German, the xxl model achieves a better score than the previous state-of-the-art, however, it is worse than gT5 xxl. Whereas for Russian, mT5 trained on CLANG-8 does not match state-of-the-art performance. We believe this is in part due to the small size of CLANG-8 in Russian. Additionally, the training data for Russian and German comes from the same dataset as the test data which is not the case for English, making the training data of significantly greater relevance. For German and Russian GEC tasks, where in-domain training data is unavailable, CLANG-8 could have a greater impact.

We release the re-labeled CLANG-8 dataset, which contains 2.4M training examples for English, 114k examples for German, and 45k examples for Russian. The Czech portion of Lang-8 would have resulted in only 2k examples, and as such is excluded.

5 Conclusion

In this paper we report new state-of-the-art results on GEC benchmarks in four languages we studied. Our simple setup relies on a language-agnostic approach to pretrain large multi-lingual language models. To enable the distillation of our largest model into smaller, more efficient models, we released a cleaned version of the LANG-8 dataset, enabling easier and even more accurate training of GEC models.

Acknowledgements

We would like to thank Costanza Conforti, Shankar Kumar, Felix Stahlberg and Samer Hassan for useful discussions as well as their help with training and evaluating the models.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019a. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019b. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. 2019. Kermit: Generative insertion-based modeling for sequences.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *International Conference*

on Natural Language Processing, pages 478–490. Springer.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *arXiv preprint arXiv:2005.11849*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Exploring grammatical error correction with not-so-crummy machine translation. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

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, Portland, Oregon, USA. Association for Computational Linguistics.