On the Impact of Data Augmentation on Downstream Performance in Natural Language Processing

Itsuki Okimura, Machel Reid, Makoto Kawano, Yutaka Matsuo
The University of Tokyo
{okimura, machelreid, kawano, matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

With in the broader scope of machine learning, data augmentation is a common strategy to improve generalization and robustness of machine learning models. While data augmentation has been widely used within computer vision, its use in the NLP has been comparably rather limited. The reason for this is that within NLP, the impact of proposed data augmentation methods on performance has not been evaluated in a unified manner, and effective data augmentation methods are unclear. In this paper, we look to tackle this by evaluating the impact of 12 data augmentation methods on multiple datasets when finetuning pre-trained language models. We find minimal improvements when data sizes are constrained to a few thousand, with performance degradation when data size is increased. We also use various methods to quantify the strength of data augmentations, and find that these values, though weakly correlate with downstream performance, correlate negatively or positively depending on the task. Furthermore, we find a glaring lack of consistently performant data augmentations. This all alludes to the difficulty of data augmentations for NLP tasks and we are inclined to believe that static data augmentations are not broadly applicable given these properties.

1 Introduction

Data augmentation may be useful in situations where the data size is insufficient for the number of parameters in the model, resulting in overtraining (Perez and Wang, 2017). It has been pointed out that data augmentation does not degrade the expressive power of the model and achieves an improvement in the generalization performance of the model without adjusting the hyperparameters (Hernández-García and König, 2018). While data augmentation is standard in the field of computer vision, it is not fully used in natural language processing. Two factors can be cited for this. The first reason is that there has been insufficient unified validation of data augmentation methods for a wide range of datasets and data sizes. Another reason is that it is still unclear what kind of data augmentation is effective for learning. In natural language processing, it is difficult to judge whether a data augmentation method is good or bad without relying on experiments, and it is necessary to search for effective data augmentations by trial and error (Feng et al., 2021). If it is possible to predict whether a data augmentation is effective for learning before training, it would be possible to search for data augmentations more efficiently.

This paper examines the performance impact of data augmentation methods that have been proposed for natural language processing on various datasets. Through this experiment, we will verify whether the data augmentation method can contribute to the improvement of performance on multiple datasets and problem settings. We also use various measures of the strength of a given data augmentation, and investigate its relationship with performance after learning. We find that although data augmentation strength (i.e. how significantly it perturbs the input) is correlated with the change in downstream performance to a given degree, its sign and degree often varies significantly. Based on this, we believe that static data augmentations are not a wise choice for NLP tasks with a reasonable amount of data, and may need to be combined with data-dependent modeling innovations to be broadly applicable to future work.

2 Related Work

Data Augmentation for NLP Data augmentation has been explored in NLP recently with EDA (Wei and Zou, 2019), as well as NL-augmenter (Dhole et al., 2021). Masked language modeling can be considered to be data augmentation (Devlin et al., 2019), while dictionary-derived augmentation methods have been employed recently for aug-

Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 88 - 93
May 26, 2022 ©2022 Association for Computational Linguistics

menting multilingual language models with large improvements (Chaudhary et al., 2020; Reid et al., 2021; Reid and Artetxe, 2022). However, Longpre et al. (2020) showed that two data augmentation methods in natural language processing had small effects on pre-trained language models. We further expand the scope of this study to examine the performance impact of 12 different data augmentation methods.

Evaluating Data Augmentation In the field of computer vision, researchers have been studying what kind of data augmentation contributes to the performance (Taylor and Nitschke, 2018; Perez and Wang, 2017). And some studies have been done to create metrics on data augmentation and evaluate the relationship with the performance of the model after training. Gontijo-Lopes et al. (2020) proposed two indices, affinity and diversity, to quantify how data augmentation improves the generalization of the model, and pointed out that data augmentation methods that are evaluated as having high affinity and diversity will lead to better performance in computer vision. Meanwhile, it is still unclear what characteristics of data augmentation methods are effective in the field of natural language processing.

3 Evaluation Metrics and Training Strategies

In this section, we briefly go over metrics we use to evaluate the strength of our data augmentations of a given task as well as strategies for training using data augmentations.

3.1 Training Strategy

In this subsection, we briefly discuss our two training strategies for incorporating data augmentation. Given an i.i.d. dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ containing N examples where each x_i represents an input, and y_i represents the assigned label corresponding to x_i.

Oftentimes, we simply fit a given model on this dataset. However, given a data augmentation function $f(x_i) = \hat{x}_i$, where $\hat{x}_i$ represents an augmented input, we can also augment this dataset to improve the diversity of inputs which should hopefully lead to better model generalization and robustness. That is, we now have augmented dataset $\hat{D} = \{(\hat{x}_1, y_1), \ldots, (\hat{x}_N, \hat{y}_N)\}$.

We now explain the following finetuning methods:

Normal training Finetuning our models on D

1-step training Finetuning our models jointly on augmented dataset $\hat{D}$ and original dataset D—this method is commonly employed in computer vision.

2-step training To mitigate the distribution shift introduced by the augmentation, but still allowing the model to learn from the augmented dataset, we look at two-step finetuning where we first finetune on $\hat{D}$ and then finetune on D.

3.2 Data Augmentation Strength

We also look to analyse whether there are certain trends among the strength of augmentation methods and their impact on downstream performance. To do this, we measure the strength of augmentation methods using the following metrics:

Semantic Similarity We use semantic similarity (Cer et al., 2017) as a measure of strength of data augmentation. For example, if a given example is perturbed in a more significant manner, we assume that it's semantic similarity will decrease, therefore indicating a "stronger" data augmentation. We use SentenceBERT (Reimers and Gurevych, 2019) to measure the cosine similarity between sentence representation of the original example x_i and sentence representation of augmented example $\hat{x}_i$.

BLEU We use BLEU (Papineni et al., 2002; Post, 2018) as a metric that works on discrete tokens (therefore more sensitive to exact token matches), that is not model dependent as our semantic similarity measure is. That is, a lower BLEU score represents a stronger data augmentation.

BERTScore We also use text generation metric BERTScore (Zhang* et al., 2020), which measures cosine-similarity at a token-level, rather than on a sequence-level like our semantic similarity measure.

In our analyses (Sec. 5), we measure the correlation between these measures and the ± change in performance.

4 Experimental Setup

4.1 Data Augmentation Methods

In our experiments, we compared the performance of the model when trained with 12 typical data augmentation methods with that of the model trained without data augmentation. Our data augmentations methods are sourced from NL-Augmenter[1]

[1] https://github.com/GEM-benchmark/NL-Augmenter

(Dhole et al., 2021) and nlpaug[2] (Ma, 2019). We provide additional details in Appendix B.

4.2 Datasets

In experiments, we use three datasets for different language tasks, MRPC (Dolan and Brockett, 2005), SICK (Marelli et al., 2014), and SST-2 (Socher et al., 2013). MRPC is a dataset in which the task is to predict whether a sentence-pair is semantically equivalent. SICK is a dataset that contains a task to infer the connotation between a given premise and an explanation. In this experiment, it is a binary classification problem whether the meaning of the explanatory sentence is contained in the meaning of the premise sentence or not. SST-2 is a binary classification problem in which a dataset for sentiment analysis of sentences is created from movie reviews, are classified as positive or negative. For MRPC and SICK, we extended the data to the second sentence in the experiment, and the combination of the first sentence, the extended second sentence pair, and the original label was used as the augmented data set. For SST-2, the combination of the augmented sentence and the original label was used as the augmented data set.

4.3 Models

In this experiment, we used the GPT-2 (345M) (Radford et al., 2019) and BERT-large (Devlin et al., 2018) as pre-trained language models. We train models on a single NVIDIA V100 16GB GPU. We measured the performance of training on the original dataset as a *baseline*, and compared the performance of fine-tuning on the training dataset with the augmented data. We train models until convergence, and perform early stopping where we use a patience of 3 epochs for all models.

5 Results

Performance Changes Due to Data Augmentation Table 1 shows the scores for single-step and 2-step training on the data set with data augmentation (see Appendix D for per-task results). For both training strategies, we also measure the impact of data size, experimenting with various data sizes (10%, 50%, and 100% of the full dataset). When all data was used for training, we found that no data augmentation that improved scores on average for both the language model and the masked language model, except for the 2-step training with BERT with synonym substitution. This indicates that although data-augmentation has the tendency to help at a smaller scale, perhaps mitigating effects of (lack of) data diversity, as the data scale grows we notice that performance degrades where the augmentations most likely add more noise to the dataset.

Relationship between Data Augmentation Intensity and Post-training Performance The correlation coefficients measured by the difference in F1 scores between the data augmentation intensity obtained by the language model and the masked language model and the baseline for each model and learning method are shown in Table2. A positive value indicates that a weaker (i.e. more similar) data augmentation results in better performance. When we use 1-step training, this correlation is generally positive — this indicates that when using naive data combination, then a more similar (i.e. weaker augmentation) is generally more effective. This supports our hypothesis about distribution shift negatively impact augmentation. However, this finding varies significantly when switching to 2-step training depending on model and dataset. Given the relatively strong performance of 2-step training, this indicates that strength of data augmentation can have varying effects when using various training schedules/models.

6 Discussion

When all the original training data was used for training in the three datasets tested in this study, the effect of data augmentation on performance improvement was small, and the performance on the test data deteriorated in many cases. There are two possible reasons for this. The first is that the augmented data may have become noise. It is almost inevitable that data augmentation will result in the augmentation of sentences whose labels cannot be preserved. If some of the augmented sentences are incorrectly labeled, the quality of the dataset will deteriorate to some extent. Therefore, in a setting where a relatively large number of data can be prepared, such as using all the training data, the negative impact of the decrease in data quality is stronger than the positive impact of the increase in the number of data. The second reason is that the knowledge that can be obtained by data augmentation may have already been acquired through prior learning. This is also pointed out by Longpre et al. (2020). Therefore, for data

[2] https://github.com/makcedward/nlpaug

	1- step GPT2			1-step BERT			2-step GPT-2			2-step BERT		
	100%	50%	10%	100%	50%	10%	100%	50%	10%	100%	50%	10%
baseline	0.8997	0.8795	0.8567	0.9028	0.8866	0.8461	0.8997	0.8795	0.8567	0.9028	0.8866	0.8461
character substitution	0.8929	**0.8836**	0.8356	0.8982	0.8735	**0.8517**	0.8959	0.8768	0.8483	0.8954	0.8846	**0.8494**
W2V substitution	0.8902	0.8765	0.8311	0.8939	0.8595	0.8457	0.8886	0.8779	0.8501	0.9027	0.8862	0.8406
BERT-based substitution	0.8804	0.8728	0.8452	0.8780	0.8592	**0.8462**	0.8906	0.8790	0.8304	0.8957	0.8825	0.8292
synonym substitution	0.8946	**0.8846**	0.8338	0.8971	0.8710	**0.8535**	0.8920	0.8772	**0.8585**	**0.9032**	0.8826	**0.8462**
word paraphrase	0.8916	**0.8799**	0.8509	0.8980	0.8799	**0.8518**	0.8981	**0.8820**	0.8475	0.8972	**0.8871**	0.8424
LM-based substitution	0.8910	0.8745	0.8416	0.8928	0.8654	0.8368	0.8918	**0.8740**	0.8564	0.8941	0.8858	0.8420
subject-object switching	0.8958	**0.8875**	0.8362	0.8963	0.8780	0.8451	0.8932	**0.8875**	**0.8615**	0.8968	**0.8889**	**0.8476**
random word deletion	0.8889	**0.8857**	0.8544	0.8924	0.8782	**0.8568**	0.8910	**0.8765**	**0.8606**	0.8891	**0.8873**	0.8443
stammering insertion	0.8899	**0.8799**	0.8401	0.8990	0.8795	**0.8557**	0.8920	**0.8836**	**0.8604**	0.8974	**0.8902**	**0.8496**
EDA	0.8958	0.8774	0.8226	0.8995	0.8770	**0.8529**	0.8927	**0.8860**	**0.8571**	0.8967	0.8835	**0.8514**
back translation	0.8965	**0.8809**	0.8318	0.9003	0.8786	**0.8557**	0.8968	**0.8795**	**0.8607**	0.8924	**0.8867**	**0.8483**
summarization	0.8905	0.8770	0.8490	0.8901	0.8599	**0.8501**	0.8917	**0.8864**	**0.8600**	0.8896	0.8789	**0.8471**

Table 1: Table of average F1 scores in 1-step and 2-step training for each percentage of data used for training when data augmentation is used for MRPC, SICK and SST-2.

	Sentence similarity				BLEU				BERTScore			
	GPT-2		BERT		GPT-2		BERT		GPT-2		BERT	
	1-step	2-step	1-step	2-step	1-step	2-step	1-step	2-step	1-step	2-step	1-step	2-step
MRPC	0.2478	0.5813	0.5540	0.2011	0.3782	0.5150	0.6574	0.2712	0.2406	0.6119	0.4879	0.2064
SICK	0.5138	-0.1941	0.4790	-0.5216	0.2424	0.4192	0.0392	-0.5645	0.1085	0.1314	0.0483	-0.2592
SST-2	0.3251	0.3216	0.5897	-0.4152	0.2226	0.4876	0.2015	-0.2712	0.1686	0.3699	0.4524	-0.4342

Table 2: Correlation coefficient between data augmentation strength and difference in F1 score from baseline.

augmentation in a specific domain, it is possible that data augmentation based on knowledge about the domain, such as substitution based on a list of words that can be substituted in the domain, which cannot be obtained by pre-training with a general corpus, may be effective. On the other hand, when the number of data used for training was limited, we observed some cases where the performance improved even when using a pre-training model. Therefore, in domains where only a few hundred examples are available, performance improvement can be expected by augmenting the existing data.

In addition, in 1-step learning, the weaker the data augmentation, the better the performance. However, in 2-step learning, the relationship between the strength of consistent data augmentation and performance depended on the type of data set. This suggests that in 2-step learning, the effective strength of data augmentation may differ depending on the characteristics of the data set. For example, in MRPC, the difference between the data augmentation intensity and the F1 score of the baseline was negatively correlated because even trivial changes are likely to produce data that become noise in learning. In SICK and SST-2, even if some of the content changes, the labels of the sentences are retained as long as the words indicating relevance and emotion remain the same. In this case, the various sentences created by strong data reinforcement in two-stage learning contribute to the learning process, allowing clean data to be learned in the second half. This may be why the difference between the strength of the data reinforcement and the F1 score from the baseline may have been positively correlated in some cases. Therefore, by comparing the augmentation intensity determined by the proposed index, it may be possible to efficiently search for promising data augmentation methods before actual training. However, more work needs to be done to effectively use these methods in a practical setting.

7 Conclusion

In this paper, we observed that most of the data augmentation methods did not improve performance when training on datasets with thousands of examples, but some of them improved performance when training on datasets with hundreds of examples. This suggests that, depending on the task and the data size, data augmentation may be effective even when a pre-trained language model is used for training. We also defined data augmentation intensity, a measure to evaluate whether data augmentation produces sentences that are different from the original sentences, and evaluated the relationship between this measure and the performance after training. As a result, the data augmentation intensity showed different correlations with the change in performance after training depending on the target dataset. For tasks with enough data, this indicates the limited applicability and predictability of static data augmentations. In future work, we believe the NLP community should look at modeling or adaptive learning methods (Dery et al., 2022) to account for these differences in data.

References

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, and Jiecao Chen. 2020. Dict-mlm: Improved multilingual pre-training using bilingual dictionaries.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Lucio M. Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2022. Should we be pre-training? an argument for end-task aware training as an alternative.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. Nl-augmenter: A framework for task-sensitive natural language augmentation.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. 2020. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*.

Alex Hernández-García and Peter König. 2018. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*.

Shayne Longpre, Yu Wang, and Christopher DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? *arXiv preprint arXiv:2010.01764*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proc. of WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Machel Reid and Mikel Artetxe. 2022. PARADISE: Exploiting parallel data for multilingual sequence-to-sequence pretraining. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Machel Reid, Junjie Hu, Graham Neubig, and Yutaka Matsuo. 2021. AfroMT: Pretraining strategies and reproducible benchmarks for translation of 8 African languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1306–1320, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Luke Taylor and Geoff Nitschke. 2018. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547. IEEE.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.