Domain Specific Augmentations as Low Cost Teachers for Large Students

Po-Wei Huang
National University of Singapore
huangpowei@comp.nus.edu.sg

Abstract

Current neural network solutions in scientific document processing employ models pretrained on domain-specific corpi, which are usually limited in model size, as pretraining can be costly and limited by training resources. We introduce a framework that uses data augmentation from such domain-specific pretrained models to transfer domain-specific knowledge to larger general pretrained models and improve performance on downstream tasks. Our method improves the performance of Named Entity Recognition in the astrophysical domain by more than 20% compared to domain-specific pretrained models finetuned to the target dataset.

1 Introduction

Scientific Document Processing (SDP) is an emerging field in Natural Language Processing (NLP) that proves to have more obstacles than everyday text due to the extensive scientific jargon and long text spans. Recent work in SDP on transformer architectures (Vaswani et al., 2017) has placed emphasis on constructing pretrained models in scientific corpi, such as BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019), and astroBERT (Grezes et al., 2021). However, such models are usually trained on the base size of its corresponding architectures, limiting the potential inference performances due to the smaller number of trainable parameters compared to the large-size models usually used in state-of-the-art (SOTA) performance for benchmarks in everyday text. *Are we able to achieve similar or better results with finetuning models larger in size whilst transferring knowledge from such pretrained scientific models to increase robustness?*

In this paper, we propose a training method inspired by the Unsupervised Data Augmentation (Xie et al., 2020a) and the Noisy Student (Xie et al., 2020b) framework. We first augment the training data with model that is trained on a corpus that is more closely aligned with the context domain of the target dataset. We then train a larger model on both the original training data and the augmented training data, combining the computational availability of the larger model with the domain-specific trained knowledge of the smaller domain-pretrained model.

We describe the shared task DEAL (Grezes et al., 2022) and its dataset in Section 2 and briefly review the previous work we used in Section 3. We detail our model architecture and methodology in Section 4, and go through our experimental setup and results in Section 5. Finally, we go through an in depth discussion of our results in Section 6 and conclude our findings in Section 7.

2 Task Description and Dataset

Named Entity Recognition (NER) refers to the identification and recognition of entities from a string of text. Although this task is well explored in everyday text in benchmarks such as CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and WNUT2017 (Derczynski et al., 2017), the focus of scientific text is not prominently showcased in such work. Even in benchmarks that focus on scientific document processing (SDP), the corpi in question often lie in the domain of biology and chemistry, such as the NCBI-Disease (Doğan et al., 2014) and BioCreative V CDR (Li et al., 2016) corpi, with a lack of evaluation and state-of-the-art models in the astrophysics domain.

The shared task DEAL (Detecting Entities in the Astrophysics Literature; Grezes et al. 2022) is a sequence labeling task that aims to increase the accuracy in Named Entity Recognition in the domain of astrophysics. Given the overlapping usage of historical names and acronyms in different types of astrophysical entities, it may be difficult to extract named entities in astrophysics purely by carefully constructed systematic rules. For exam-

Proceedings of the 1st Workshop on Information Extraction from Scientific Publications, pages 84–90

ple, `Maxwell` may refer to either the physician James Clerk Maxwell, a crater on the far side of the moon, or a series of equations. DEAL aims not only to discern such entities, but also to discern between such different types of entities.

The training dataset consists of 1,753 samples of text fragments from the text and acknowledgments of astrophysics papers provided by the NASA Astrophysical Data System (NASA ADS; Kurtz et al. 1993). For evaluation purposes, the labeled development dataset consists of 20 samples, while the unlabeled validation and test dataset consists of 1,366 and 2,505 samples, respectively. We evaluate the performances based on the `seqeval` (Nakayama, 2018) F1 score at the entity level and Matthew's correlation coefficient (Matthews, 1975) at the token level in the validation and test dataset.

3 Literature Review

We briefly review some previous work that are utilized in our proposed system.

3.1 Pretrained Transformer Models

With the introduction of BERT (Devlin et al., 2019), the usage of pretraining as a self-supervised technique to optimize model weights in a particular text domain for transformer architectures has been widely used in scientific document processing and other domain-specific language tasks such as biomedical text (Lee et al., 2019) and clinical notes (Alsentzer et al., 2019). We now discuss key transformer models we use in our work.

- RoBERTa (Liu et al., 2019), which is more optimally pretrained on a larger corpus compared to BERT, and has a larger vocabulary.
- SciBERT (Beltagy et al., 2019), which is pretrained on a scientific corpus with a mixture of biology and computer science papers. SciBERT's vocabulary is also constructed separately, consisting of more scientific jargon than BERT, with a token overlap of 42%.
- SpaceTransformers (Berquand et al., 2021), a series of models including SpaceRoBERTa and SpaceSciBERT, which are further trained on astronomical text based on the base model of RoBERTa and the uncased version SciBERT on its scientific vocabulary, respectively. SpaceTransformers do not construct a new vocabulary and instead reuse the vocabularies constructed in the original models.

3.2 Adapter Architecture

Adapters (Houlsby et al., 2019) are introduced as a parameter-efficient alternative to finetune transformer models (Vaswani et al., 2017) for downstream tasks. Unlike finetuning, which modifies the top layer of the transformer, adapters inject layers of parameters into the architecture itself, training only on these injected parameters while freezing the parameters of the original network. Adapter training consumes much less computational cost when compared to direct finetuning, making it a more cost-efficient architecture to adopt while training large sized models.

3.3 Data Augmentation and Semi-Supervised Methods

Data augmentation is a commonly used technique in semi-supervised training in conjunction with unlabeled data to increase the robustness of the model. Xie et al. (2020a) noted that such augmentations should have both diversity and validity compared to the original data. They proposed using backtranslation (Sennrich et al., 2016; Edunov et al., 2018) as an augmentation method to produces paraphrases of the original text that can be utilized for sequence classification tasks.

In the same paper, the authors introduced a semi-supervised learning technique named Unsupervised Data Augmentation (UDA; Xie et al. 2020a) which compares unlabeled data with its augmented version by introducing a consistency loss term, reasoning that a robust enough model should yield similar predictions. For sequence labeling tasks, Lowell et al. (2021) proposed to augment the data by randomly masking parts of the test and filling in the masked tokens with BERT (Devlin et al., 2019), similar to a cloze test, as known as the MaskLM task. Furthermore, Lowell et al. (2021) also showed that even without the inclusion of unlabeled data, adding a consistency loss term by comparing training data and its augmented version can also increase the robustness of the inference model.

Another semi-supervised learning framework, the Noisy Student, proposed by (Xie et al., 2020b), utilizes self-training and pseudo-labeling to iteratively train a series of student-teacher models that increase in performance level. A normal teacher model is first trained on labeled images. The teacher model is then used to generate pseudo labels for the unlabeled data. The labeled and now pseudo-labeled data would then be used to train an

equal-or-larger student model with noise injected via data augmentation and model dropout. The process can then be iterated using the student model as the new teacher model and training a new student model.

4 Architecture and Methodology

We propose a system that uses data augmentation as a low-cost method of teacher-student training to transfer domain-specific knowledge to a larger adapter-based model.

4.1 Preprocessing

The DEAL training dataset contains samples that far exceed the size of the token number of 512 that transformer models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are pretrained on. Although transformer architectures for long text such as Big Bird (Zaheer et al., 2020) can be used train the entire text in less than quadratic time, we reason that the recognition of named entities may not require the contextual information of text in sentences in which the named entity itself does not reside. We instead partition the sample text into multiple input cases, separating the text by sentence via regex.

We first identify end-of-sentence characters, namely periods, question marks, and exclamation marks. We then partition the text unless the end-of-sentence character is followed by another punctuation or whitespace followed by punctuation, in which case we partition after the punctuation. Using the `nltk` library (Bird et al., 2009), we avoid tokenizing common abbreviations such as "Mr." and "Dr.", as well as other abbreviations found in the training data and scientific text in general such as "fig.", "tab.", "et al.", etc. Due to capitalization being important in the identification of named entities, we retain capitalization after tokenization.

The training dataset is partitioned into 25596 samples after our preprocessing, with an average of 22.39 words and a standard deviation of 15.64 words. Furthermore, the number of named entities in a sample has an average of 1.6, and a standard deviation of 2.6, with 41159 named entities in the training dataset in total.

4.2 Augmentation

For our data augmentation step, we borrowed the consistency loss term from UDA (Xie et al., 2020a) on a supervised basis and augment our text by BERT based MaskLM as suggested by Lowell et al. (2021). We take this a step further and view the MaskLM data augmentation technique as a low-cost teacher model that we can use to further train a larger student model while finetuning the training dataset. Replacing the simple BERT for data augmentation domain-specific pretrained models such as SciBERT (Beltagy et al., 2019), we aim to transfer the domain-specific knowledge of such models to the main backbone model. We randomly mask 30% of the total tokens as suggested by Lowell et al. (2021), and, following Devlin et al. (2019), replacing 80% of such tokens with the `[MASK]` token, 10% of such tokens with a random token, and keep 10% unchanged. However, as our task requires the augmented text to have the same amount of words as the original, since our labels are provided on a word-level basis, we revert the tokens to the original if the replaced token causes a reduction or increase of words in the augmented sentence.

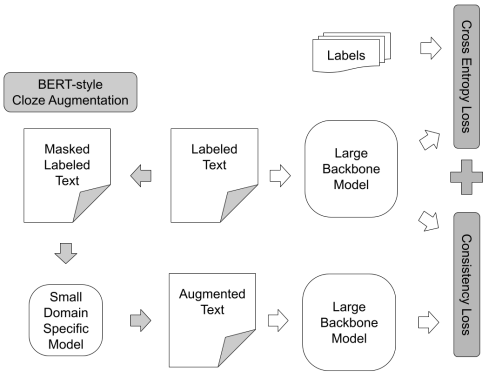

Figure 1: Our Proposed Architecture for Low Cost Domain Specific Teachers

4.3 Backbone Model Architecture

Instead of training smaller student models to perform knowledge distillation, we take inspiration from the Noisy Student framework (Xie et al., 2020b) and train a student model larger than the teacher model to act as our backbone model for training. Due to its various SOTA performances in GLUE (Wang et al., 2018), we select DeBERTaV3-large (He et al., 2021a,b) as our backbone model.

As opposed to finetuning the backbone model directly, we use the adapter (Houlsby et al., 2019) version of the model to decrease computational costs, while obtaining similar results to finetuning the full model itself.

Original:	This research made use of NASA's Astrophysics Data System Bibliographic Services; the SIMBAD data base (Wenger et al. 2000) and VizieR catalogue access tool (Ochsenbein, Bauer Marcout 2000), both operated at CDS, Strasbourg, France; and the Jean-Marie Mariotti Center Aspro2 service 1 .
Augmented:	**The project** made use of NASA's Astrophysics Data System Bibliographic **database**; the SIMBAD data base (Wenger et al. 2000) and VizieR **data** access tool (**Sch,ouin, and** Marcout 2000), which operated at **CNR**, Strasbourg, France; and the Jean-Marie Mariotti Center **Asprox** service 1 .

* Bold text indicates augmented text.
† ulined text indicates named entities.

Table 1: Sample Augmentations by CosmicRoBERTa

4.4 Loss Function Engineering

Incorporating the augmented data created from the MaskLM task, we add an additional consistency loss between the original data and the augmented data during training, as shown in Figure 1.

We now write the full loss term that we use for training. Let $\mathcal{X} = \{(x_b, y_b) : b \in 1, 2, \cdots, b\}$ be a batch of B labeled data samples with x_b being the input sample and y_b being the ground-truth label. We denote $\hat{y}(x)$ as the predicted class distribution of sample x made by the model. Further, we also denote $H(q, p)$ the standard cross-entropy loss of predicted distribution p and target distribution q, and $D(q||p)$ as the Kullback–Leibler divergence (Kullback and Leibler, 1951) between distributions p and q. Denoting the augmentation via MaskLM as $\mathcal{A}(\cdot)$, we get the loss term that we use for training:

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} H(y_b, \hat{y}(x_b)) + D(\hat{y}(\mathcal{A}(x_b))||\hat{y}(x_b)) \tag{1}$$

For validation and testing purposes, we compute the loss term based on the cross entropy loss alone.

5 Experiments

We describe the experimental setup and the results in this section.

5.1 Experimental Setup

We implement our model using PyTorch (Paszke et al., 2019) and Lightning[1], importing pretrained model weights from Huggingface (Vaswani et al., 2017). We set the learning rate of 3×10^{-4} on the AdamW optimizer (Loshchilov and Hutter, 2019). Training was conducted on a single core 12GB NVIDIA K80 kernel.

5.2 Results

We present an abridged comparison of our results and established baselines provided in the DEAL task in Table 2. Our best model on the DEAL *testing dataset* uses Pfeiffer et al. (2020)'s adapter architecture of the DeBERTaV3-large model as the backbone model and uses CosmicRoBERTa[2], a further pretrained version of SpaceRoBERTa (Berquand et al., 2021), as the augmentation teacher model. Our model has a +20 improvement on the F1 score, while having a +8 improvement on the MCC score, indicating an increase in performance both on the token-level and the entity-level recognition of entities.

6 Analysis

We now present a more detailed analysis of the performance of different variants of our model and some considerations between experimental setups.

6.1 Large Parameter Efficient Models

Our first idea to increase performance is simple: Use a larger model to boost performance, as the increased number of hyperparameters to tune and the larger architecture indicates a larger capacity to generalize to the training dataset. In order to train a large sized model on limited training resources to increase accuracy, we adopt the usage of adapter architecture due to the reduction of tunable parameters by two orders without affecting training convergence (Houlsby et al., 2019), which also reduces memory usage as less gradient computations need to be computed and stored. According to the

[1] https://github.com/Lightning-AI/lightning

[2] https://huggingface.co/icelab/cosmicroberta

	F1(entity)	MCC(word)
Random	0.0166	0.1089
BERT (Devlin et al., 2019)	0.4738	0.7405
SciBERT (Beltagy et al., 2019)	0.5595	0.8016
astroBERT (Grezes et al., 2021)	0.5781	0.8104
(*Ours*) DeBERTaV3$_{\text{adapter}}$ (He et al., 2021a,b; Houlsby et al., 2019)		
+ SciBERT (Beltagy et al., 2019)	0.7751	0.8898
+ CosmicRoBERTa (Berquand et al., 2021)	*0.7799*	*0.8928*

Table 2: Evaluation Results on Testing Dataset

	F1(entity)	MCC(word)	Accuracy(entity)
astroBERT	0.5781	0.8104	0.9389
DeBERTaV3$_{\text{adapter}}$ (He et al., 2021a,b; Houlsby et al., 2019)	0.7896	0.8987	0.9667
+ SciBERT$_{\text{cased}}$ (Beltagy et al., 2019)	**0.7988**	**0.9063**	**0.9692**
+ RoBERTa (Liu et al., 2019)	0.7970	0.9057	0.9690
+ CosmicRoBERTa (Berquand et al., 2021)	0.7972	0.9050	0.9687
+ SpaceSciBERT$_{\text{uncased}}$ (Berquand et al., 2021)	0.7859	0.9030	0.9680

Table 3: Augmentation Model Comparison on Validation Dataset

empirical results of Rücklé et al. (2021), the use of adapters speeds up training approximately 1.35 times. With such settings, we are able to construct the baseline model directly by using DeBERTV3-large in an adapter setting, achieving a +21 improvement on the entity-level F1 metric and a +8 improvement on the word-level MCC metric without further augmentations. (See Tab. 3)

6.2 Augmentation as Teacher Models

Using the results of direct finetuning of the DeBERTaV3 model as our baseline, we explore the effects of using different pretrained "teacher models" to augment training data. We present the training results in Table 3, evaluated in the validation dataset.

We find that augmentation via SciBERT seems to provide the best performance on the validation dataset, while augmentation via CosmicRoBERTa provides the best performance on the test dataset.

As we are using the MaskLM task to augment sentences, the model would only fill the masked tokens with tokens in its vocabulary, which would rely on both the vocabulary itself and the model's ability to fill in the correct token. While CosmicRoBERTa is pretrained on an astronomical corpus, the vocabulary itself is based on RoBERTa, thus producing a more valid augmentation, but not diverse enough. On the other hand, SciBERT has a self-constructed vocabulary, thus such an augmentation would produce a more diverse augmentation, or at least an augmentation containing more scientifically oriented text, but not valid enough. On the other hand, while SpaceSciBERT seems to fit the above two criteria of diversity and validity, the model itself is uncased, hence the produced augmented words are uncased, leading to a poor augmentation, the model would underfit on the augmented data and overfit on the training data, leading to poorer performance during inference.

For further work, we expect the usage of astroBERT as an augmentation teacher model to be more beneficial than previous attempts, as the model is both pretrained on astrophysical text, and contains a vocabulary with more jargon, achieving both diversity and validity in augmentation.

7 Conclusion

In this paper, we show that we are able to surpass models pretrained on domain-specific knowledge by utilizing general corpus pretrained adapter models of larger sizes. Furthermore, such a method can by used in conjunction to the aforementioned domain-specific pretrained models via data augmentation to transfer such knowledge to the backbone model. Further work may explore other methods of augmentation to act as teacher models or combining multiple augmentations in training.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In Proceedings of the 2nd Clinical Natural Language Processing Workshop, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Audrey Berquand, Paul Darm, and Annalisa Riccardi. 2021. SpaceTransformers: Language modeling for space systems. IEEE Access, 9:133111–133122.

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

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. volume 47, pages 1–10.

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, Brussels, Belgium. Association for Computational Linguistics.

Felix Grezes, Thomas Allen, Tirthankar Ghosal, and Sergi Blanco-Cuaresma. 2022. Overview of the first shared task on detecting entities in the astrophysics literature (deal). In Proceedings of the 1st Workshop on Information Extraction from Scientific Publications, Taipei, Taiwan. Association for Computational Linguistics.

Felix Grezes, Sergi Blanco-Cuaresma, Alberto Accomazzi, Michael J. Kurtz, Golnaz Shapurian, Edwin Henneken, Carolyn S. Grant, Donna M. Thompson, Roman Chyla, Stephen McDonald, Timothy W. Hostetler, Matthew R. Templeton, Kelly E. Lockhart, Nemanja Martinovic, Shinyi Chen, Chris Tanner, and Pavlos Protopapas. 2021. Building astroBERT, a language model for astronomy & astrophysics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with disentangled attention. In International Conference on Learning Representations.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799. PMLR.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. The Annals of Mathematical Statistics, 22(1):79 – 86.

M. J. Kurtz, T. Karakashian, C. S. Grant, G. Eichhorn, S. S. Murray, J. M. Watson, P. G. Ossorio, and J. L. Stoner. 1993. Intelligent Text Retrieval in the NASA Astrophysics Data System. In Astronomical Data Analysis Software and Systems II, volume 52 of Astronomical Society of the Pacific Conference Series, page 132.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. Database, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In International Conference on Learning Representations.

David Lowell, Brian Howard, Zachary C. Lipton, and Byron Wallace. 2021. Unsupervised data augmentation with naive augmentation and without unlabeled data. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4992–5001, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure, 405(2):442–451.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 46–54, Online. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pages 142–147.

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 Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. In Advances in Neural Information Processing Systems, volume 33, pages 6256–6268. Curran Associates, Inc.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020b. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In Advances in Neural Information Processing Systems, volume 33, pages 17283–17297. Curran Associates, Inc.