CLULab-UofA at SemEval-2024 Task 8: Detecting Machine-Generated Text Using Triplet-Loss-Trained Text Similarity and Text Classification

MohammadHossein Rezaei **Yeaeun Kwon** **Reza Sanayei**
Abhyuday Singh **Steven Bethard**
University of Arizona
mhrezaei,yeaeunkwon,rsanayei,abhyudaysingh,bethard@arizona.edu

Abstract

Detecting machine-generated text is a critical task in the era of large language models. In this paper, we present our systems for SemEval-2024 Task 8, which focuses on multi-class classification to discern between human-written and maching-generated texts by five state-of-the-art large language models. We propose three different systems: unsupervised text similarity, triplet-loss-trained text similarity, and text classification. We show that the triplet-loss-trained text similarity system outperforms the other systems, achieving 80% accuracy on the test set and surpassing the baseline model for this subtask. Additionally, our text classification system, which takes into account sentence paraphrases generated by the candidate models, also outperforms the unsupervised text similarity system, achieving 74% accuracy.

1 Introduction

The rapid evolution of large language models (LLMs) has significantly impacted the dynamics of information exchange, blurring the lines between human and machine-generated text. State-of-the-art LLMs are available to the public on a large scale, allowing users to generate human-like text with minimal effort. This advancement poses a dual-edged sword: while offering unprecedented capabilities in generating human-like text, it also raises critical concerns about privacy(Huang et al., 2022), ethics (Smiley et al., 2017; Kamocki and Witt, 2022), and misinformation (Pan et al., 2023; Goldstein et al., 2023; Stiff and Johansson, 2022) —especially given the LLMs' tendency to produce plausible yet factually baseless content, known as hallucinations (Dziri et al., 2022; Das et al., 2022). Distinguishing between human and machine authorship has thus emerged as a major challenge, bearing implications for content credibility and ethical standards in digital communication. As a response to the need for effective detection methods that can discern the origin of text in this new landscape, the SemEval-2024 Task 8 (Wang et al., 2024) presents an exciting challenge of AI-generated text detection over three different subtasks: Subtask A: Binary Human-Written vs. Machine-Generated Text Classification, Subtask B: Multi-Way Machine-Generated Text Classification, and Subtask C: Human-Machine Mixed Text Detection.

In this paper, we work on Subtask B, which focuses on multi-class classification to distinguish between human-written and machine-generated text by five state-of-the-art LLMs. These models are ChatGPT, `text-davinci-003`, LLaMa (Touvron et al., 2023), Cohere, Dolly-v2 (Conover et al., 2023), and BLOOMz (Muennighoff et al., 2023).

We propose three different systems to address this task: unsupervised text similarity, triplet-loss-trained text similarity, and text classification.

We show that the triplet-loss-trained text similarity system outperforms the other systems, achieving 80% accuracy on the test set and surpassing the baseline model for this subtask. Additionally, our text classification system, which takes into account sentence paraphrases generated by the candidate models, also outperforms the unsupervised text similarity system, achieving 74% accuracy. However, the unsupervised text similarity system performs poorly, achieving only 29% accuracy on the test set. We note that the latter is the only system that we submitted to the task, and the other systems are post-evaluation improvements. The main contributions of this paper are:

- An unsupervised text similarity system that computes cosine similarity to measure text similarity, which assesses the angle between vector representations of texts.
- A sentence transformer trained with triplet loss to learn the distinctions between the given texts.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1498–1504

- A RoBERTa classifier that makes decisions based on the given paragraph.
- A RoBERTa classifier which takes into account sentence paraphrases generated by the candidate models.

2 Background and Related Work

Recent research has resulted in significant advancements in Natural language generation (NLG) models (Vaswani et al., 2017) and generative pre-trained transformer (GPT) models (Devlin et al., 2019; Qiu et al., 2020) However, with potential threats posed by these models, research on identifying machine-generated text has also surged (Jawahar et al., 2020; Valiaiev, 2024). Initially, methods employing traditional machine learning models such as logistic regression were proposed (Ippolito et al., 2020). Nevertheless, the limitation of the machine learning model, which requires extensive re-training (Valiaiev, 2024), and the rise of the pre-trained transformer models, have prompted researchers to adopt the large models. Relatively smaller language models such as RoBERTa (Liu et al., 2020) have achieved state-of-the-art performance across various domains including social media, news articles, and online reviews (Uchendu et al., 2020; Adelani et al., 2020; Fagni et al., 2021). In addition to this, other approaches based on contrastive learning and similarity metrics (Boenninghoff et al., 2019) have also emerged. Such research efforts continue with the ongoing evolution and adoption of text-generative models.

3 Dataset

We work with M4 (Wang et al., 2023), a dataset for SemEval-2024 Task 8, which consists of 71,027 data samples for the training set, 3,000 data samples for the development set, and 18,000 data samples for the test set. Each sample is labeled with one of the six labels: Human, ChatGPT, Davinci, Cohere, BLOOMz, or Dolly. Figure 1 shows an example of the given dataset, which consists of id, text, model, label, and source.

Wang et al. (2023) prompted these models to write a passage given some information from the source. The sources of the texts are diverse, including Wikipedia, WikiHow (Koupaee and Wang, 2018), Reddit, arXiv, and Peer-Read (Kang et al., 2018).

4 System Overview

In this section, we provide a comprehensive overview of the three approaches we explored: unsupervised text similarity, triplet-loss-trained text similarity, and text classification.

4.1 Approach 1: Unsupervised Text Similarity

The first strategy we submitted is based on computing cosine similarity to measure text similarity, which assesses the angle between vector representations of texts. A label for multi-class classification is assigned based on the highest cosine similarity score.

4.1.1 Model Architecture

The text data is encoded using a pre-trained sentence transformer model (Reimers and Gurevych, 2019) without any additional training, followed by computing the averaged pooling embedding across all the training instances of each class. Subsequently, we calculated the cosine similarity between the text and the average-pooled embedding for each class, assigning the text to the class with the highest cosine similarity. This approach effectively categorizes the semantic similarity of texts based on topics and classifies texts with divergent writing styles (Ibrahim et al., 2023).

4.2 Approach 2: Triplet-Loss-Trained Text Similarity

Text similarity models can also be trained on the provided training data.[1] For this approach, we train a sentence transformer model with a triplet loss, which requires three inputs during training: anchor, positive, and negative samples $(x_i, {x_i}^+, {x_j}^-)$. This loss function aims to minimize the distance between the anchor and positive data $(x_i, {x_i}^+)$ while simultaneously maximizing the distance between the anchor and negative data$(x_i, {x_j}^-)$ (Ren and Xue, 2020). We conduct this training to enhance the vector representations of texts for multi-class classification.

4.2.1 Constructing Triplets

To construct the dataset with three inputs, we adopt the concept of hard positive ${x_i}^+$ and hard negative ${x_j}^-$ sampling. Hard positive involves selecting a text with the lowest similarity within the same class i, whereas hard negative involves choosing a text with the highest similarity from different

[1]This approach is a post-evaluation improvement and was not submitted to the task.

Id	Text	Model	Label	Source
557	Have you ever wanted to surprise someone with a unique and personalized cake? Look no further than an iPhone cake! With a few simple steps and some creativity, you can make a one-of-a-kind dessert that will impress anyone who sees it. Follow these steps to make your own iPhone cake: 1. Prepare 2 rectangular package cakes that can be easily form-fitted to fit with round corners. If you can't find rectangular cakes, you can simply cut and shape the cakes after baking to create the desired size and shape. . . . 10. Use several colors of fondant to create some of the apps all devices have. Work with these small pieces of colored fondants. You can use a toothpick to stick these apps into the cake, or use water and a brush to brush them onto the cake. In conclusion, making an iPhone cake is not as difficult as it may seem. This cake will be a hit with anyone, from your kids to your coworkers, and will impress them with your creativity. Just follow these simple steps and enjoy the final result!	ChatGPT	1	wikihow

Figure 1: An example of the given dataset consists of id, text, model, label, and source. Note that some part of the text from the middle is truncated with . . . for brevity.

Type	Text	Label
Anchor	How to Play Forza Motorsport This wikiHow teaches you how to play Forza ...	Human
Positive	Perfumes are a blend of different levels of scent, also called "notes". When you spray a ...	Human
Negative	Forza Motorsport is a popular racing game that provides players with the ability ...	ChatGPT

Table 1: An example of the triplet dataset which consists of anchor, positive, and negative. These pairs are chosen in a mini-batch for training. Anchor and positive data have the lowest similarity within the same class, and negative data shows the highest similarity to anchor within different classes.

classes. This concept maximizes the distinction between various classes (Robinson et al., 2021; Xu and Bethard, 2021). As the metric for similarity, we employ cosine similarity to select hard positive x_i^+ and hard negative samples x_j^- within a mini-batch. An example of the triple dataset is shown in Table 1.

4.2.2 Model Architecture

We first fine-tuned the same pre-trained sentence transformer model as in approach 1 (Reimers and Gurevych, 2019) using the triplet data and TripletMarginLoss. Then we attached a six-way classification head to the transformer using a linear layer and CrossEntropy loss. Figure 2 illustrates the overall framework, including triplet learning and classification.

4.3 Approach 3: Text Classification

We also explored a simple text classification approach where a classifier takes the given passage as the input and predicts one of the six possible labels (human or one of five LLMs) as output.[2]

We explored a variant of this text classification approach where we augment the input by asking each of the five LLMs to generate a short text. We mask a random sentence in the input paragraph and

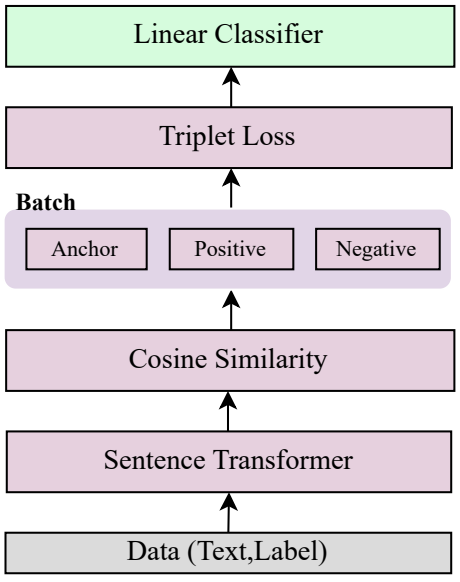

Figure 2: The overall framework of our triplet learning system proposed for Semeval-2024 Task 8.

[2]This approach is a post-evaluation improvement and was not submitted to the task.

then prompt the models to fill in the mask with a sentence that has a meaning similar to the original sentence in their own style. Due to computational limitations, we were unable to run Dolly due to its memory requirements and did not have enough resources to generate sentence paraphrases for all the over 70,000 instances in the training set. Therefore, we randomly chose 4,000 instances of each class for training, and generated paraphrases for all models other than Dolly.

4.3.1 Model Architecture

For both text classification models, we train a transformer that takes text as input and produces one of the six possible labels (human or one of five LLMs) as output. Due to the limitation of the number of input tokens for the transformer model we use (RoBERTa), we had to truncate the inputs to keep 512 tokens of the given paragraph and, for the input-augmented model, 128 tokens from each sentence paraphrase.

5 Experiments

5.1 Experimental Setup

For the unsupervised text similarity approach, we used the `paraphrase-distilroberta-base-v1` sentence transformers model from the HuggingFace library (Wolf et al., 2020). This model is based on the DistilRoBERTa architecture for clustering or semantic search. We computed the cosine similarity between the text embeddings using the PyNNDescent library[3] to facilitate an approximate nearest neighbor search in a huge dataset.

For the triplet-trained text similarity approach, we used the same sentence transformers model but trained it on the training data. We explored different hyper-parameter combinations, varying two learning rates (1-e5 and 3e-5) and two batch sizes (16 and 32) across 5 epochs and 10 epochs. Our final embedding model was trained using a learning rate of 1e-5 and a batch size of 16 for 10 epochs. For the six-way multi-class classification learning, we experimented with several classification head formulations: ReLU activation functions, dropout layers, and linear layers. The final classifier was trained with a linear layer and CrossEntropy loss. For this multi-class classification, we utilized a learning rate of 1e-5 and a batch size of 32 for 10 epochs.

[3] https://github.com/lmcinnes/pynndescent

Split	Metrics	UnSim	TripSim	TextCls	ParaCls
Test	A	0.29	0.80	0.72	0.74
Test	P	0.37	0.82	0.79	0.81
Test	R	0.29	0.80	0.72	0.74
Test	F1	0.24	0.79	0.71	0.73

Table 2: Accuracy (A), precision (P), recall (R), and F1 score of unsupervised text similarity (UnSim), triplet-trained text similarity (TripSim), text classification (TextCls), and paraphrase-augmented text classification (ParaCls).

For the text classification approach, we used the `roberta-large` model (Liu et al., 2020) from the HuggingFace library (Wolf et al., 2020). We used a learning rate of 1e-6 and 5e-7 respectively with a batch size of 8, and early stopping set to 3.

5.2 Results

Table 2 shows the performance of our different approaches – unsupervised text similarity (UnSim), triplet-trained text similarity (TripSim), text classification (TextCls), and paraphrase-augmented text classification (ParaCls) – in terms of accuracy (A), Precision (P), Recall (R), and F1 score. The submitted approach, UnSim, shows low metrics scores: 29% accuracy for the test dataset. Both the simple text classifier and the paraphrase-augmented text classifier performed better, achieving 72% and 74% accuracy on the test set, respectively. The paraphrase augmentation provided some additional information to the model, with a statistically significant improvement (McNemar's test (McNemar, 1947), $p < 0.05$) over not using sentence paraphrases. The best model was the text similarity model trained with triplet loss, which achieved 80% accuracy and 82% precision on the test dataset, surpassing the baseline model for this subtask. This improved performance underscores that the embedding obtained from triplet loss effectively learned the text distinctions by maximizing the differences between positive and negative samples.

We provide a breakdown by label for the text classification models, as shown in Table 3.

6 Conclusion

In this paper, we presented several different systems for SemEval 2024 Task 8's text classification between human and five distinct machines. Our submitted model, which relied on unsupervised embeddings coupled with cosine similarity, was poor at handling the diverse writing styles over the

Model	Label	P	R	F-1
TextCls	Human	1.00	0.44	0.61
TextCls	ChatGPT	0.52	1.00	0.68
TextCls	Cohere	0.99	0.61	0.75
TextCls	Davinvi	0.70	0.51	0.59
TextCls	BLOOMz	0.76	1.00	0.86
TextCls	Dolly	0.80	0.77	0.78
ParaCls	Human	0.99	0.39	0.56
ParaCls	ChatGPT	0.52	0.95	0.67
ParaCls	Cohere	0.97	0.66	0.78
ParaCls	Davinvi	0.77	0.64	0.70
ParaCls	BLOOMz	0.93	0.99	0.96
ParaCls	Dolly	0.67	0.80	0.73

Table 3: Detailed breakdown of results on the test set for the text classification models.

same topics that were present in the data, resulting in low classification scores. Our text classification approaches and our triplet-trained text similarity approach all outperformed the simple unsupervised model. The triplet loss learning especially improved performance over the submitted model, with its pretraining allowing it to better maximize the distinctions between texts.

For future work, we plan to adapt our systems to other classification tasks. We also plan to explore other methods for training the triplet loss model, such as using a larger model or using a different loss function. Additionally, using a larger dataset for the text classification models could improve their performance.

7 Limitations

We note that our systems are not perfect and have several limitations. For instance, we did not have enough resources to generate sentence paraphrases for all instances in the training set. We also did not have enough resources to run Dolly due to its memory requirements. Additionally, we did not explore other methods for training the triplet loss model, such as using a larger model or using a different loss function. Finally, we acknowledge that we did not try using LLMs for the classification of machine-generated text, which could potentially improve the performance of our systems.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Advanced Information Networking and Applications*, pages 1341–1354, Cham. Springer International Publishing.

Benedikt Boenninghoff, Robert M. Nickel, Steffen Zeiler, and Dorothea Kolossa. 2019. Similarity learning for authorship verification in social media. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2457–2461.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Souvik Das, Sougata Saha, and Rohini Srihari. 2022. Diving deep into modes of fact hallucinations in dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 684–699, Abu Dhabi, United Arab Emirates. 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.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Josh A. Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Momen Ibrahim, Ahmed Akram, Mohammed Radwan, Rana Ayman, Mustafa Abd-El-Hameed, Nagwa El-Makky, and Marwan Torki. 2023. Enhancing authorship verification using sentence-transformers. *Working Notes of CLEF*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pawel Kamocki and Andreas Witt. 2022. Ethical issues in language resources and language technology – tentative categorisation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 559–563, Marseille, France. European Language Resources Association.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Fuji Ren and Siyuan Xue. 2020. Intention detection based on siamese neural network with triplet loss. *IEEE Access*, 8:82242–82254.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

Charese Smiley, Frank Schilder, Vassilis Plachouras, and Jochen L. Leidner. 2017. Say the right thing right: Ethics issues in natural language generation systems. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 103–108, Valencia, Spain. Association for Computational Linguistics.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Dmytro Valiaiev. 2024. Detection of machine-generated text: Literature survey.

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.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th*

International Workshop on Semantic Evaluation, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

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.

Dongfang Xu and Steven Bethard. 2021. Triplet-trained vector space and sieve-based search improve biomedical concept normalization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 11–22, Online. Association for Computational Linguistics.