Paraphrasing with Large Language Models

Sam Witteveen
Red Dragon AI
sam@reddragon.ai

Martin Andrews
Red Dragon AI
martin@reddragon.ai

Abstract

Recently, large language models such as GPT-2 have shown themselves to be extremely adept at text generation and have also been able to achieve high-quality results in many downstream NLP tasks such as text classification, sentiment analysis and question answering with the aid of fine-tuning. We present a useful technique for using a large language model to perform the task of paraphrasing on a variety of texts and subjects. Our approach is demonstrated to be capable of generating paraphrases not only at a sentence level but also for longer spans of text such as paragraphs without needing to break the text into smaller chunks.

1 Introduction

Paraphrase generation is an NLP task that has multiple uses in content creation, question answering, translation, and data augmentation. It is a task that has been attempted for many decades using statistical and rules-based approaches (McKeown, 1979; Meteer and Shaked, 1988).

We propose a system that generates paraphrased examples in an autoregressive fashion using a neural network, without the need for techniques such as top-k word selection or beam search.

We demonstrate that by using large language models we are able to produce not only paraphrases that are longer and of a higher quality than previous work, but can also paraphrase text beyond the individual sentence-level (i.e. full paragraphs at a time).

The large language models we use implement the encoder-decoder structure of the transformer architecture (Vaswani et al., 2017) which has been shown to learn different representations of language at each level of its encoding (Devlin et al., 2019). The power of language models like GPT-2 (Radford et al., 2019) and BERT allows them to develop useful representations of language which can be used far beyond just generation of the next word (Rothe et al., 2019). In our experiments, we have observed that the models have representations of syntax and grammar, allowing them to be fine-tuned for the task of paraphrase generation.

2 Related Work

Paraphrase generation has attracted a number of different NLP approaches. These have included rule-based approaches (McKeown, 1979; Meteer and Shaked, 1988) and data-driven methods (Madnani and Dorr, 2010), with recently the most common approach being that the task is treated as a language translation task (Bannard and Callison-Burch, 2005; Barzilay and McKeown, 2001; Pang et al., 2003) - often performed using a bilingual corpus pivoting back and forth (Madnani and Dorr, 2010; Prakash et al., 2016; Mallinson et al., 2017). Other methods proposed include more recently the use of Deep Reinforcement Learning (Li et al., 2018) , supervised learning using sequence-to-sequence models (Gupta et al., 2018; Prakash et al., 2016) and unsupervised approaches (Bowman et al., 2016; Roy and Grangier, 2019).

3 Methodology

In this section, we define the task, present our implementation, describe our evaluation protocol and the paraphrase datasets used.

Paraphrase generation can be described as generating a phrase which is semantically as close as possible to the original while being rewritten in new words and phrases.

3.1 Large Language Model training

Training language models has become a popular way of creating models suited for Transfer Learning in the field of NLP. (Howard and Ruder, 2018;

		USE	ROUGE-L	BLEU
In:	A prisoner can asphyxiate himself in 90 seconds and, after eight minutes or so, he will be brain dead.			
Out:	In 90 seconds, a prisoner can asphyxiate himself and be brain dead after eight minutes or so.	0.9326	0.4706	0.4730
In:	The restaurant is a carved-off space up a couple of stairs to one side, dominated by faux bare-brick columns, faux-wood floors and an air of foetid despondency.			
Out:	It is a carved-off space, up a couple of flights of stairs, to the other side of the restaurant, dominated by fake bare-brick columns, fake wood floors and an air of foetid despondency.	0.8954	0.5000	0.5348
In:	I signed a bill that made the problem worse, and I want to admit it, he said.			
Out:	He signed a bill that made the problem worse and he wants to admit it.	0.8363	0.4667	0.5299
In:	It said the damage to the wing provided a pathway for hot gasses to penetrate the ship's thermal armor during Columbia's ill-fated reentry.			
Out:	The document says the damage to the wing provided a pathway for hot gases to penetrate Columbia's thermal armour during its fatal re-entry.	0.9190	0.4545	0.5445

Table 1: Examples of GPT-2 generated paraphrased sentences with scores for each pair

Peters et al., 2018; Radford et al., 2018; Dong et al., 2019). While these models are initially trained in a semi-supervised manner to predict the next word or words in a sequence, they can be fine-tuned and used for a variety of downstream NLP tasks such as text classification, sentiment analysis, tagging, and entity extraction.

More recently, large language models using transformer architectures are achieving state of the art results for many of these tasks while using less supervised data than previously needed.

One example of these large language models that has proven to be very good at text generation is GPT-2. It makes use of a transformer architecture and comes in various sizes up to 1.5 billion parameters. In these experiments, we have taken a pre-trained version of the GPT-2 model trained in a semi-supervised fashion on the WebText dataset (Radford et al., 2019) of over 8 million documents with 40 GB of text in total.

3.2 Fine-tuning for Task

We take the GPT-2 model and fine-tune it on a supervised dataset of pre-made paraphrase examples. These examples are fed into the model as original phrase / paraphrase pairs, separated by a specific identifying sequence (such as ">>>>").

This training is done for a small number of epochs to give the model just enough examples of what the task is asking from the model : The goal being to avoid overfitting the model on the new data, while giving it sufficient exposure to the task to enable it to learn the general pattern expected.

While we experimented with TPUs for the fine-tuning, in the end we were able to reproduce the same results on a single K-80 GPU with around 90 minutes of training.

Once the model is fine-tuned, we find that it can also produce similar paraphrase training examples if sampled from with no conditional input. To give an indication of training progress, these 'naive' paraphrases are sampled on a periodic basis during the training.

After fine-tuning on this dataset, we are then able to feed in any original phrase followed by the unique token and have the model generate paraphrases on demand.

3.3 Candidate Generation and Selection

After the model is trained, we then sample from the model using previously unseen sentences as conditional input. This conditional input allows us to generate multiple candidate sentences for the single original sentence.

While the quality of the paraphrases is somewhat variable, by generating multiple outputs and then scoring them, we can select just the best quality paraphrases based on a number of criteria that serve to filter our output down to a set of satisfactory results.

First, we obtain a similarity score between the generated paraphrase and the original sentence by using the Universal Sentence Encoder (USE) (Cer et al., 2018) to make a 512 dimensional sentence embedding for each output sentence and then compare them to the embedding of the original sentence via the cosine similarity measure.

As a second step, we measure the ROUGE-L (Lin, 2004) score of the candidate paraphrases against the original sentence and eliminate candidates with a ROUGE-L score of above 0.7 . This prevents candidates that are too close to the original sentence being chosen. After testing both cutoff scores for ROUGE-L and BLEU (Papineni et al., 2002), ROUGE-L has shown to be more useful at finding candidates that are more unique in comparison to the original sentence.

By choosing samples with sufficiently low ROUGE-L scores but as high a similarity as possible, we end up with an output that is semantically similar to the original phrase but has a unique word order when compared to the original phrase.

3.4 Datasets

We fine-tuned multiple versions of the model on several different datasets : 2 datasets of sentences and their matching paraphrases; and 1 dataset of paragraphs with matching paraphrases :

1. The MSR Paraphrase Identification dataset (Dolan et al., 2004) which consists of just over 4,000 examples of original sentences with a matching paraphrased sentence in its train set.

2. An original dataset of 10,000 sentences from online news articles along with matching paraphrases that were human-generated.

3. A further original dataset of paragraphs with corresponding paraphrased paragraphs from various entertainment, news, and food articles found online, where the paraphrases were human-generated.

We fine-tuned 3 versions of the GPT-2 model, one corresponding to each dataset, and then made predictions using the same system outlined above.

By calculating USE, ROUGE-L and BLEU scores for each dataset we are able to quantify the quality of human-generated paraphrases and then use that as a comparison for the models generated sentences (see Table 2).

Dataset	USE	R-L	BLEU
MSR_train	0.8462	0.4315	0.4593
MSR_test	0.8415	0.4202	0.4966
News dataset	0.8948	0.4686	0.5648
Paragraphs dataset	0.9208	0.4966	0.5762

Table 2: Average USE, ROUGE-L, BLEU Scores of the datasets

4 Experiments

We implemented the system described above using GPT-2 and trained it on the different datasets for various lengths of training.

To evaluate the output of the model, we randomly selected sentences from sources such as Wikipedia, news sites and entertainment sites with no matching paraphrase to use as the conditional input to the model.

5 Results and Scoring

When comparing our generated sentences with the average scores of the original datasets, we can see that that they compare favorably.

As discussed earlier, we assessed the semantic similarity of the sentence meanings using Universal Sentence Encoder (Cer et al., 2018) and compared them to the average USE score from the datasets that were trained on. This showed that the system can generate paraphrases which are semantically on par with the human-generated ones in each of the datasets.

We also compared the ROUGE-L (Lin, 2004) scores of the generated samples with the average values for the datasets which were human-generated. This again shows that our phrases are coherent and on par with human-generated paraphrases.

When we further compared the results of unfiltered examples generated by the model (Table 3) we observe that when the USE score is below 0.85 we see clear deterioration in the semantic similarity quality of the paraphrased versions.

We also observe that if the USE score is too close to 1.0 then the ROUGE-L score also rises and the generated examples are too similar in word

		USE	R-L
In:	A prisoner can asphyxiate himself in 90 seconds and, after eight minutes or so, he will be brain dead.		
Out 1:	After 8 minutes, a brain fart will subdue the sufferer.	0.524	0.0
Out 2:	After 8 minutes, he will be brain-dead and his heart will stop.	0.565	0.138
Out 3:	A brain aneurysm can asphyxiate itself in 90 seconds and, after eight minutes, it will be dead.	0.721	0.412
Out 4:	After eight minutes, a brain anesthetist can asphyxiate a prisoner in 90 seconds and for several minutes after that.	0.758	0.167
Out 5:	A brain-dead prisoner canasphyxiate himself in 90 seconds and then out loud after eight minutes.	0.809	0.312
Out 6:	At asphyxiation, the prisoner canasphyxiate himself in 90 seconds and, after 8 minutes, he will be brain dead.	0.884	0.514
Out 7:	After eight minutes, a prisoner can asphyxiate himself in 90 seconds and, after that, he will be brain dead.	0.884	0.514
Out 8*:	**In 90 seconds, a prisoner can asphyxiate himself and be brain dead after eight minutes or so**	**0.932**	**0.473**
Out 9:	A prisoner can asphyxiate himself in 90 seconds and, after eight minutes, he will be brain dead.	0.972	0.824

Table 3: Showing Candidates Selection and Scoring - *Selected Sentence

and phrase selection to the original sentence to be useful paraphrases.

This technique can be performed not only at sentence-level but also to generate paragraph-level paraphrases. Comparing USE and ROUGE-L scores of the generated paragraphs we see they are again on par with the human generated examples from our paragraph dataset (samples are given in the Supplemental Materials).

Due to the pre-training of the Language Model, the model is able to generalize to and generate paraphrases for types of content it has never seen during the fine-tuning phase.

6 Discussion

The technique outlined in this paper shows the applicability of large language models to the paraphrasing task. It also highlights that there is still much to be learnt about further applications of large language models, and also the approaches used to fine-tune and use them for applications.

Most of the results from models such as GPT-2 have focused on the quality of text generation rather than quantitative methods for measuring and improving the quality of text created, to make it more consistent and usable. We propose the scoring and filtering of candidates using techniques such as we have shown with USE and ROUGE-L, may be a useful technique not just for paraphrasing but other text generation tasks.

The ability of our technique to work with long spans of text also gives it an advantage over prior work which used rule-based and other statistical approaches which performed best on shorter spans of text.

Our experiments show that pre-training of GPT-2 on such a large amount of data in the WebText dataset allows it to 'understand' the syntax and to a degree the grammar of English allowing it to be able to quickly learn the task of paraphrasing through fine-tuning training on a small set of paraphrasing examples.

7 Future Work

Extrapolating from the paraphrasing results into more generalizable ideas, we hope to investigate the extent by which the representations learned in the different layers of the transformer network correspond to different parts of the linguistic hierarchy. One possible approach to doing this would be to trace a set of 'markers' through the transformer networks existing attention mechanism, in parallel to the text which gives rise to that structure.

In addition, the ability of the networks to learn tasks within the span of a single context frame indicates the possibility of an inherent bias towards meta-(or one-shot) learning. These will be the subject of further work.

Acknowledgments

We would like to thank Google for access to the TFRC TPU program which was used in training and fine-tuning models for this paper.

References

Colin J. Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *ArXiv*, abs/1803.11175.

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.

William B. Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *CoRR*, abs/1905.03197.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36:341–387.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893.

Kathleen McKeown. 1979. Paraphrasing using given and new information in a question-answer system. In *17th Annual Meeting of the Association for Computational Linguistics*, pages 67–72.

Marie Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–188.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. *CoRR*, abs/1610.03098.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2019. Leveraging pre-trained checkpoints for sequence generation tasks. *CoRR*, abs/1907.12461.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. *ArXiv*, abs/1905.12752.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.