Improving Paraphrase Generation models with machine translation generated pre-training

Anonymous ACL submission

Abstract

Paraphrase generation is a fundamental and longstanding problem in the Natural Language Processing field. With the huge success of pre-trained transformers, the pre-train–fine-tune approach has become a standard choice. At the same time, popular task-agnostic pre-trainings usually require terabyte datasets and hundreds of GPUs, while available pre-trained models are limited to architecture and size. We propose a simple and efficient pre-training approach specifically for paraphrase generation, which noticeably boosts model quality and doesn't require significant computing power. We also investigate how this procedure influences the scores across different architectures and show that it helps them all.

1 Introduction

Paraphrase Generation is one of the most popular and challenging tasks in the field of Natural Language Processing. There are several good reasons for this. First, this task is a special case of text generation. And there are many models for text generation to apply to the paraphrase generation task. Secondly, the task of paraphrase generation is essentially an analog of machine translation, with the only difference being that the sentence must be translated into the same language, but in other words. Therefore, not only machine translation models are directly applicable to this task, but machine translation quality metrics are entirely suitable for paraphrase systems estimation.

The peculiarity of paraphrase generation in comparison with other tasks of Natural Language Processing is a large number of works that don't use labeled data but operate only with the usual text corpora. The fact is that the input and output for this task are interchangeable: if from the sentence $x_1, x_2, \ldots, x_m$ we can get the sentence $y_1, y_2, \ldots, y_k$ with a high probability, then it is logical that at the input $y_1, y_2, \ldots, y_k$ the output $x_1, x_2, \ldots, x_m$ must have a high probability. Moreover, each sentence should not have strictly 1 paraphrase and could be rewritten in different ways, which emphasizes the probabilistic nature of the problem. There are a relatively large number of data sources of different quality and different levels for the Paraphrase Generation task.

In this article, we present a description of the approach for improving the quality of neural networks for Paraphrase Generation. We propose a simple and efficient pre-training procedure, which is task-specific. It consistently boosts the performance across different evaluation sets and model architectures.

2 Approach

Nowadays, the pre-train–fine-tune paradigm prevails. Especially in Natural Language Processing, pre-trainings have led to significant performance gains. It was shown that this technique adds robustness, enriches the model with better contextual representation and additional knowledge. Usually, the models are pre-trained on a large unlabeled text corpus. Training objectives could be both general, like Masked Language Modelling, or task-specific. For instance, synthetic data generation (denoising task) is widely known to boost the accuracy of neural Grammatical Error Correction systems (Zhao et al., 2019; Omelianchuk et al., 2020).

Ideally, we need a dataset, which would be huge in terms of the number of examples and related to the task. For Paraphrase Generation, ParaNMT-50M (Wieting and Gimpel, 2017) fits well for this purpose. It contains more than 50 million English-English sentential paraphrase pairs. It's generated automatically by using neural machine translation to translate the non-English side of a large parallel corpus. Thus, we can first train the model on this data and then fine-tune it on specific Paraphrase Generation datasets.

System	QQP (test)			MSCOCO (dev)		
	BLEU↑	TER↓	METEOR↑	BLEU↑	TER↓	METEOR↑
residual LSTM	28.4	59.1	30.2	26.9	63.3	24.2
transformer base	29.1	59.5	30.5	26.9	63.3	24.2
CGMH	22.5	65.0	27.0	17.3	72.6	21.9
MCPG	24.1	64.5	31.8	16.5	73.5	23.2
PTS	25.6	58.7	31.4	17.0	69.9	22.8
pre-trained transformer base	**30.6**	**57.4**	**33.2**	27.4	58.5	**26.0**
pre-trained LSTM + Luong attn	29.2	58.1	32.6	26.7	59.0	25.5
pre-trained fully convolutional	29.5	57.8	32.6	**27.7**	**57.8**	25.7

Table 1: Comparison of ParaNMT pre-trained models against other reported systems

3 Experimental Setup

3.1 Datasets

Following the majority of works for supervised Paraphrase generation, we use the MSCOCO (Lin et al., 2014) dataset and the Quora Question Pairs[1] (QQP) dataset in our experiments. The MSCOCO dataset was initially built for the image captioning task. Each image corresponds to 5 different annotations, which describe the most noticeable object or action. These captions can be treated as paraphrases, as they're generally close to each other. There're two versions of the dataset: 2014 and 2017. We use the 2017 version. For each set of paraphrases, we use all possible pairs during training, which helps to increase the number of training examples significantly. For the evaluation stage, we use the first description as a source and the rest as references.

The QQP dataset is a paraphrase identification corpus. Questions from the Quora website were marked as either duplicate or not by moderators. In the experiments, we use those pairs, which are labeled as duplicates. As there are no train/dev/test splits in the original dataset, we follow the partition in Wang et al. (2017). Similarly, for each pair, we use both questions as the paraphrase of each other. During the evaluation, we have only 1 reference.

3.2 Metrics

As the evaluation of text generation is usually challenging, we rely on a combination of metrics in our experiments. We report surface metrics BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) and semantic metric METEOR (Lavie and Agarwal, 2007). As studied by Wubben et al. (2010), human judgments on generated paraphrases correlate well with these metrics.

To ensure the evaluation is robust, we use the SacreBLEU (Post, 2018) library for BLEU and TER calculation. For METEOR, we use the original Java scorer. Both libraries accept detokenized (raw) data and thus eliminate tokenization influence.

3.3 Training parameters

In our experiments, we use 3 different neural network architectures: fully convolutional (Gehring et al., 2017), LSTM (Hochreiter and Schmidhuber, 1997), and transformer (Vaswani et al., 2017). All neural networks have a comparable number of parameters. We use shared embeddings both for encoder, decoder input, and decoder output (softmax), as paraphrase generation is a monolingual task.

For the transformer model, we simply use the base setup. For the LSTM model, we use 3-layer (both encoder and decoder) LSTM with Luong attention and hidden size 512. The fully convolutional model has the following structure: 4 layers of convolutions with kernel size 512 and width 3; 2 layers of convolutions with kernel size 1024 and width 3; 1 layer of convolutions with kernel size 2048 and width 1. During training, we use an inverse square root schedule with a warm-up. We first train models on the ParaNMT-50M dataset and then fine-tune them on QQP and MSCOCO separately.

4 Results

We report the results of our experiments in Table 1. There are multiple issues with Paraphrase Generation evaluation methodology, like different dataset versions or splits, sentence length shrinking, tok-

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Architecture	QQP (test)			MSCOCO (dev)		
	BLEU↑	TER↓	METEOR↑	BLEU↑	TER↓	METEOR↑
No pre-training						
transformer base	**28.7**	**58.6**	**31.5**	25.2	**60.6**	24.5
LSTM with Luong attn	27.5	59.7	30.1	25.0	61.1	24.6
fully convolutional	27.9	59.9	30.7	**25.3**	61.5	**24.9**
With pre-training						
transformer base	**30.6**	**57.4**	**33.2**	27.4	58.5	**26.0**
LSTM with Luong attn	29.2	58.1	32.6	26.7	59.0	25.5
fully convolutional	29.5	57.8	32.6	**27.7**	**57.8**	25.7
Gain from pre-training						
transformer base	**1.9**	1.2	1.7	**2.2**	2.1	**1.5**
LSTM with Luong attn	1.7	1.3	**2.5**	1.7	2.1	0.9
fully convolutional	1.6	**2.1**	1.9	**2.2**	**3.7**	0.8
No fine-tuning						
transformer base	23.7	66.7	31.4	17.4	69.9	23.6
LSTM with Luong attn	24.1	64.1	31.5	18.1	68.7	24.1
fully convolutional	24.2	65.8	31.4	17.9	69.1	24.2

Table 2: Comparison of the models initialized randomly, pre-trained on ParaNMT, and trained solely on ParaNMT for Paraphrase Generation task regarding the model architecture. The models evaluated on QQP test set and MSCOCO dev set

enization. Thus, to be able to compare our models, we use an evaluation strategy similar to Fabre et al. (2021) and compare our results with them.

In their work, the authors train the neural networks from previous works on Paraphrase Generation with fixed train and evaluation strategies. Among them enocoder-decoder models, like residual LSTM (Prakash et al., 2016) and transformer base, and weakly-supervised method CGMH (Miao et al., 2019). Additionally, they present Monte-Carlo Tree Search For Paraphrase Generation (MCPG) and Pareto Tree Search (PTS) methods, where paraphrase generation task is treated as a multicriteria search problem by using PPDB 2.0 large-scale database (Pavlick et al., 2015).

The models pre-trained on ParaNMT consistently show better results across both evaluation sets and all metrics. While the difference on the BLEU metric is not that big, transformer base shows significant improvement on METEOR and TER. Moreover, pre-trained LSTM is on par (or better) with the best encoder-decoder models.

5 Ablation study

In the era of pre-trained language models, transformer architecture is the default choice for Natural Language Processing. At the same time, some of the recent studies (Tay et al., 2021) show that not only transformers can incorporate knowledge gained during the pre-training stage. In this study, we investigate the influence of Paraphrase Generation pre-training on model quality regarding the architecture.

In Table 2, we observe that the pre-training boosts the model performance regardless of the architecture. In some cases fully convolutional and LSTM models outperforms transformer in terms of the score gain from pre-training. Surprisingly, the gain is bigger on average on the MSCOCO dataset (in terms of BLEU and TER), despite the fact that its training set is bigger than QQP.

We also explore the quality of the neural networks trained solely on the ParaNMT-50M dataset, without further fine-tuning, in Table 2 (lower block). For such models, METEOR score is higher on the QQP test set and similar on the MSCOCO dev set, while the models trained on the actual datasets expectedly prevail in terms of BLEU and TER.

Another observation is that the ParaNMT-only transformer shows consistently worse results on both datasets compared to LSTM and full convolutional nets. One of the possible reasons is that thanks to better inductive bias, the transformer better tunes to the ParaNMT dataset and, thus, generalizes worse on other datasets.

6 Related Work

Based on the idea of variational autoencoders with discrete latent structures, in Fu et al. (2020a) authors propose a latent bag of words (BOW) model for paraphrase generation. The semantics of a discrete latent variable is modeled by the BOW from the target sentences. This latent variable is used to build a fully differentiable content planning and surface realization model. Source words are used to predict their neighbors and model the target BOW with a mixture of softmax. Gumbel top-k reparameterization is employed to perform differentiable subset sampling from the predicted BOW distribution. The retrieved sampled word embeddings are used to augment the decoder and guide its generation search space.

In paper Krishna et al. (2020), authors reformulate unsupervised style transfer as a paraphrase generation problem, and present a simple methodology based on fine-tuning pretrained language models on automatically generated paraphrase data. Despite its simplicity, the described method significantly outperforms state-of-the-art style transfer systems on both human and automatic evaluations.

Work Goyal and Durrett (2020) proposes to use syntactic transformations to softly "reorder" the source sentence and guide neural paraphrasing model. First, given an input sentence, the method derives a set of feasible syntactic rearrangements using an encoder-decoder model. This model operates over a partially lexical, partially syntactic view of the sentence and can reorder big chunks. Next, the method uses each proposed rearrangement to produce a sequence of position embeddings, which encourages the final encoder-decoder paraphrase model to attend to the source words in a particular order.

A method for generating paraphrases of English questions that retain the original intent but use a different surface form was proposed in Hosking and Lapata (2021). An encoder-decoder model was trained to reconstruct a question from a paraphrase with the same meaning and an exemplar with the same surface form, leading to separated encoding spaces. A Vector-Quantized Variational Autoencoder was used to represent the surface form as a set of discrete latent variables that allows the application of a classifier to select a different surface form at test time. It was experimentally proved that the proposed model is able to generate paraphrases with a better tradeoff between semantic preservation and syntactic novelty compared to previous methods.

In paper Fu et al. (2020b), authors explore the use of structured variational autoencoders to infer latent templates for sentence generation using a soft, continuous relaxation in order to utilize reparameterization for training. Specifically, they propose a Gumbel-CRF, a continuous relaxation of the CRF sampling algorithm using a relaxed Forward Filtering Backward-Sampling (FFBS) approach. As a reparameterized gradient estimator, the Gumbel-CRF gives more stable gradients than score-function based estimators. As a structured inference network, it was shown that it learns interpretable templates during training, which allows it to control the decoder during testing. The effectiveness of methods was demonstrated with experiments on unsupervised paraphrase generation.

7 Conclusion

In this paper, we studied the effect of ParaNMT pre-training for Paraphrase Generation. We propose a simple and efficient approach for improving the quality of neural models for the task. We show that ParaNMT pre-training significantly benefits neural networks regardless of the architecture. Moreover, models trained solely on the ParaNMT already perform well on both evaluation sets.

Relevant pre-training enhances neural networks' quality at no cost in terms of model size or inference time. Task-agnostic pre-training procedures require substantial computational resources, and available models are limited to architectures. At the same time, task-specific pre-training significantly improves model performance while being easier to reach.

References

Betty Fabre, Tanguy Urvoy, Jonathan Chevelu, and Damien Lolive. 2021. Neural-driven search-based paraphrase generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2100–2111.

Yao Fu, Yansong Feng, and John P Cunningham. 2020a. Paraphrase generation with latent bag of words. *arXiv preprint arXiv:2001.01941*.

Yao Fu, Chuanqi Tan, Bin Bi, Mosha Chen, Yansong Feng, and Alexander M Rush. 2020b. Latent template induction with gumbel-crfs. *arXiv preprint arXiv:2011.14244*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. *arXiv preprint arXiv:2005.02013*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Tom Hosking and Mirella Lapata. 2021. Factorising meaning and form for intent-preserving paraphrasing. *arXiv preprint arXiv:2105.15053*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: Tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

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.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Yi Tay, Mostafa Dehghani, Jai Gupta, Dara Bahri, Vamsi Aribandi, Zhen Qin, and Donald Metzler. 2021. Are pre-trained convolutions better than pre-trained transformers? *arXiv preprint arXiv:2105.03322*.

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.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. *arXiv preprint arXiv:1903.00138*.