Transformer and seq2seq model for Paraphrase Generation

Elozino Egonmwan and **Yllias Chali**
University of Lethbridge
Lethbridge, AB, Canada
{elozino.egonmwan, yllias.chali}@uleth.ca

Abstract

Paraphrase generation aims to improve the clarity of a sentence by using different wording that convey similar meaning. For better quality of generated paraphrases, we propose a framework that combines the effectiveness of two models – transformer and sequence-to-sequence (seq2seq). We design a two-layer stack of encoders. The first layer is a transformer model containing 6 stacked identical layers with multi-head self-attention, while the second-layer is a seq2seq model with gated recurrent units (GRU-RNN). The transformer encoder layer learns to capture long-term dependencies, together with syntactic and semantic properties of the input sentence. This rich vector representation learned by the transformer serves as input to the GRU-RNN encoder responsible for producing the state vector for decoding. Experimental results on two datasets-QUORA and MSCOCO using our framework, produces a new benchmark for paraphrase generation.

1 Introduction

Paraphrasing is a key abstraction technique used in Natural Language Processing (NLP). While capable of generating novel words, it also learns to compress or remove unnecessary words along the way. Thus, gainfully lending itself to abstractive summarization (Chen and Bansal, 2018; Gehrmann et al., 2018) and question generation (Song et al., 2018) for machine reading comprehension (MRC) (Dong et al., 2017). Paraphrases can also be used as simpler alternatives to input sentences for machine translation (MT) (Callison-Burch et al., 2006) as well as evaluation of natural language generation (NLG) texts (Apidianaki et al., 2018).

Existing methods for generating paraphrases, fall in one of these broad categories – rule-based (McKeown, 1983), seq2seq (Prakash et al., 2016), reinforcement learning (Li et al., 2018), deep generative models (Iyyer et al., 2018) and a varied combination (Gupta et al., 2018; Mallinson et al., 2017) of the later three.

In this paper, we propose a novel framework for paraphrase generation that utilizes the transformer model of Vaswani et al. (2017) and seq2seq model of Sutskever et al. (2014) specifically GRU (Cho et al., 2014). The multi-head self attention of the transformer complements the seq2seq model with its ability to learn long-range dependencies in the input sequence. Also the individual attention heads in the transformer model mimics behavior related to the syntactic and semantic structure of the sentence (Vaswani et al., 2017, 2018) which is key in paraphrase generation. Furthermore, we use GRU to obtain a fixed-size state vector for decoding into variable length sequences, given the more qualitative learned vector representations from the transformer.

The main contributions of this work are:

- We propose a novel framework for the task of paraphrase generation that produces quality paraphrases of its source sentence.
- For in-depth analysis of our results, in addition to using BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) which are word-overlap based, we further evaluate our model using qualitative metrics such as Embedding Average Cosine Similarity (EACS), Greedy Matching Score (GMS) from Sharma et al. (2017) and METEOR (Banerjee and Lavie, 2005), with stronger correlation with human reference.

2 Task Definition

Given an input sentence $S = (s_1, ..., s_n)$ with n words, the task is to generate an alternative output

Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019), pages 249–255
Hong Kong, China, November 4, 2019.
www.aclweb.org/anthology/D19-56%2d

S: What are the dumbest questions ever asked on Quora? **G:** what is the stupidest question on quora? **R:** What is the most stupid question asked on Quora?
S: How can I lose fat without doing any aerobic physical activity **G:** how can i lose weight without exercise? **R:** How can I lose weight in a month without doing exercise?
S: How did Donald Trump won the 2016 USA presidential election? **G:** how did donald trump win the 2016 presidential **R:** How did Donald Trump become president?

Table 1: Examples of our generated paraphrases on the **QUORA** sampled test set, where **S**, **G**, **R** represents Source, Generated and Reference sentences respectively.

S: Three dimensional rendering of a kitchen area with various appliances. **G:** a series of photographs of a kitchen **R:** A series of photographs of a tiny model kitchen
S: a young boy in a soccer uniform kicking a ball **G:** a young boy kicking a soccer ball **R:** A young boy kicking a soccer ball on a green field.
S: The dog is wearing a Santa Claus hat. **G:** a dog poses with santa hat **R:** A dog poses while wearing a santa hat.
S: the people are sampling wine at a wine tasting. **G:** a group of people wine tasting. **R:** Group of people tasting wine next to some barrels.

Table 2: Examples of our generated paraphrases on the **MSCOCO** sampled test set, where **S**, **G**, **R** represents Source, Generated and Reference sentences respectively.

sentence $Y = (y_1, ..., y_m) \mid \exists y_m \notin S$ with m words that conveys similar semantics as S, where preferably, $m < n$ but not necessarily.

3 Method

In this section, we present our framework for paraphrase generation. It follows the popular encode-decode paradigm, but with two stacked layers of encoders. The first encoding layer is a transformer encoder, while the second encoding layer is a GRU-RNN encoder. The paraphrase of a given sentence is generated by a GRU-RNN decoder.

3.1 Stacked Encoders

3.1.1 Encoder – TRANSFORMER

We use the transformer-encoder as sort of a pre-training module of our input sentence. The goal is to learn richer representation of the input vector that better handles long-term dependencies as well as captures syntactic and semantic properties before obtaining a fixed-state representation for decoding into the desired output sentence. The transformer contains 6 stacked identical layers mainly driven by self-attention implemented by Vaswani et al. (2017, 2018).

3.1.2 Encoder – GRU-RNN

Our architecture uses a single layer uni-directional GRU-RNN whose input is the output of the transformer. The GRU-RNN encoder (Chung et al., 2014; Cho et al., 2014) produces fixed-state vector representation of the transformed input sequence using the following equations:

$$z = \sigma(x_t U^z + s_{t-1} W^z) \quad (1)$$

$$r = \sigma(x_t U^r + s_{t-1} W^r) \quad (2)$$

$$h = tanh(x_t U^h + (s_{t-1} \odot r) W^h) \quad (3)$$

$$s_t = (1 - z) \odot h + z \odot s_{t-1} \quad (4)$$

where r and z are the reset and update gates respectively, W and U are the network's parameters, s_t is the hidden state vector at timestep t, x_t is the input vector and $\odot$ represents the Hadamard product.

3.2 Decoder – GRU-RNN

The fixed-state vector representation produced by the GRU-RNN encoder is used as initial state for the decoder. At each time step, the decoder receives the previously generated word, y_{t-1} and hidden state s_{t-1} at time step t_{-1}. The output word, y_t at each time step, is a softmax probability of the vector in equation 3 over the set of vocabulary words, V.

50K					
MODEL	BLEU	METEOR	R-L	EACS	GMS
VAE-SVG-EQ (Gupta et al., 2018)	17.4	22.2	-	-	-
RbM-SL (Li et al., 2018)	35.81	28.12	-	-	-
TRANS (ours)	35.56	**33.89**	27.53	79.72	62.91
SEQ (ours)	34.88	32.10	29.91	78.66	61.45
TRANSEQ (ours)	37.06	33.73	**30.89**	80.81	**63.63**
TRANSEQ + beam (size=6) (ours)	**37.12**	33.68	30.72	**81.03**	63.50
100K					
MODEL	BLEU	METEOR	R-L	EACS	GMS
VAE-SVG-EQ (Gupta et al., 2018)	22.90	25.50	-	-	-
RbM-SL (Li et al., 2018)	**43.54**	32.84	-	-	-
TRANS (ours)	37.46	**36.04**	29.73	80.61	64.81
SEQ (ours)	36.98	34.71	32.06	79.65	63.49
TRANSEQ (ours)	38.75	35.84	**33.23**	81.50	**65.52**
TRANSEQ + beam (size=6) (ours)	38.77	35.86	33.07	**81.64**	65.42
150K					
MODEL	BLEU	METEOR	R-L	EACS	GMS
VAE-SVG-EQ (Gupta et al., 2018)	38.30	33.60	-	-	-
TRANS (ours)	39.00	**38.68**	32.05	81.90	65.27
SEQ (ours)	38.50	36.89	34.35	80.95	64.13
TRANSEQ (ours)	**40.36**	38.49	**35.84**	**82.84**	**65.99**
TRANSEQ + beam (size=6) (ours)	39.82	38.48	35.40	82.48	65.54

Table 3: Performance of our model against various models on the **QUORA** dataset with **50k,100k,150k** training examples. R-L refers to the ROUGE-L F1 score with 95% confidence interval

4 Experiments

We describe baselines, our implementation settings, datasets and evaluation of our proposed model.

4.1 Baselines

We compare our model with very recent models (Gupta et al., 2018; Li et al., 2018; Prakash et al., 2016) including the current state-of-the-art (Gupta et al., 2018) in the field. To further highlight the gain of stacking 2 encoders we use each component – Transformer (TRANS) and seq2seq (SEQ) as baselines.

- VAE-SVG-EQ (Gupta et al., 2018): This is the current state-of-the-art in the field, with a variational autoencoder as its main component.

- RbM-SL (Li et al., 2018): Different from the encoder-decoder framework, this is a generator-evaluator framework, with the evaluator trained by reinforcement learning.

- Residual LSTM (Prakash et al., 2016): This implements stacked residual long short term memory networks (LSTM).

- TRANS: Encoder-decoder framework as described in Section 3 but with a single transformer encoder layer.

- SEQ: Encoder-decoder framework as described in Section 3 but with a single GRU-RNN encoder layer.

4.2 Implementation

We used pre-trained 300-dimensional $gloVe$[1] word-embeddings (Pennington et al., 2014) as the distributed representation of our input sentences. We set the maximum sentence length to 15 and 10 respectively for our input and target sentences following the statistics of our dataset.

For the transformer encoder, we used the $transformer\_base$ hyperparameter setting from

[1] https://nlp.stanford.edu/projects/glove/

MODEL	BLEU	METEOR	R-L	EACS	GMS
Residual LSTM (Prakash et al., 2016)	37.0	27.0	-	-	-
VAE-SVG-EQ (Gupta et al., 2018)	41.7	31.0	-	-	-
TRANS (ours)	41.8	38.5	33.4	79.6	70.3
SEQ (ours)	40.7	36.9	35.8	78.9	70.0
TRANSEQ (ours)	43.4	38.3	37.4	80.5	71.1
TRANSEQ + beam (size=10) (ours)	**44.5**	**40.0**	**38.4**	**81.9**	**71.3**

Table 4: Performance of our model against various models on the **MSCOCO** dataset. R-L refers to the ROUGE-L F1 score with 95% confidence interval

the tensor2tensor library (Vaswani et al., 2018)[2], but set the hidden size to 300. We set dropout to 0.0 and 0.7 for MSCOCO and QUORA datasets respectively. We used a large dropout for QUORA because the model tends to over-fit to the training set. Both the GRU-RNN encoder and decoder contain 300 hidden units.

We pre-process our datasets, and do not use the pre-processed/tokenized versions of the datasets from tensor2tensor library. Our target vocabulary is a set of approximately 15,000 words. It contains words in our target training and test sets that occur at least twice. Using this subset of vocabulary words as opposed to over 320,000 vocabulary words contained in $gloVe$ improves both training time and performance of the model.

We train and evaluate our model after each epoch with a fixed learning rate of 0.0005, and stop training when the validation loss does not decrease after 5 epochs. The model learns to minimize the seq2seq loss implemented in `tensorflow API`[3] with `AdamOptimizer`. We use greedy-decoding during training and validation and set the maximum number of iterations to 5 times the target sentence length. For testing/inference we use beam-search decoding.

4.3 Datasets

We evaluate our model on two standard datasets for paraphrase generation – **QUORA**[4] and **MSCOCO** (Lin et al., 2014) as described in Gupta et al. (2018) and used similar settings. The **QUORA** dataset contains over 120k examples with a 80k and 40k split on the training and test sets respectively. As seen in Tables 1 and 2, while the QUORA dataset contains question pairs, MSCOCO contains free form texts which are human annotations of images. Subjective observation of the MSCOCO dataset reveals that most of its paraphrase pairs contain more novel words as well as syntactic manipulations than the QUORA pairs making it a more interesting paraphrase generation corpora. We split the QUORA dataset to 50k, 100k and 150k training samples and 4k testing samples in order to align with baseline models for comparative purposes.

4.4 Evaluation

For quantitative analysis of our model, we use popular automatic metrics such as BLEU, ROUGE, METEOR. Since BLEU and ROUGE both measure $n-gram$ word-overlap with difference in brevity penalty, we report just the ROUGE-L value. We also use 2 additional recent metrics – GMS and EACS by (Sharma et al., 2017)[5] that measure the similarity between the reference and generated paraphrases based on the cosine similarity of their embeddings on word and sentence levels respectively.

4.5 Result Analysis

Tables 3 and 4 report scores of our model on both datasets. Our model pushes the benchmark on all evaluation metrics compared against current published top models evaluated on the same datasets. Since several words could connote similar meaning, it is more logical to evaluate with metrics that match with embedding vectors capable of measuring this similarity. Hence we also report GMS and EACS scores as a basis of comparison for future work in this direction.

Besides quantitative values, Tables 1 and 2 show that our paraphrases are well formed, abstractive (*e.g dumbest – stupidest, dog is wearing*

[2] https://github.com/tensorflow/tensor2tensor

[3] https://www.tensorflow.org/api_docs/python/tf/contrib/seq2seq/sequence_loss

[4] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[5] https://github.com/Maluuba/nlg-eval

– *dog poses*), capable of performing syntactic manipulations (*e.g in a soccer uniform kicking a ball – kicking a soccer ball*) and compression. Some of our paraphrased sentences even have more brevity than the reference, and still remain very meaningful.

5 Related Work

Our baseline models – VAE-SVG-EQ (Gupta et al., 2018) and RbM-SL (Li et al., 2018) are both deep learning models. While the former uses a variational-autoencoder and is capable of generating multiple paraphrases of a given sentence, the later uses deep reinforcement learning. In tune, with part of our approach, ie, seq2seq, there exists ample models with interesting variants – residual LSTM (Prakash et al., 2016), bi-directional GRU with attention and special decoding tweaks (Cao et al., 2017), attention from the perspective of semantic parsing (Su and Yan, 2017).

MT has been greatly used to generate paraphrases (Quirk et al., 2004; Zhao et al., 2008) due to the availability of large corpora. While much earlier works have explored the use of manually drafted rules (Hassan et al., 2007; Kozlowski et al., 2003).

Similar to our model architecture, Chen et al. (2018) combined transformers and RNN-based encoders for MT. Zhao et al. (2018) recently used the transformer model for paraphrasing on different datasets. We experimented using solely a transformer but got better results with TRANSEQ. To the best of our knowledge, our work is the first to cross-breed the transformer and seq2seq for the task of paraphrase generation.

6 Conclusions

We proposed a novel framework, TRANSEQ that combines the efficiency of a transformer and seq2seq model and improves the current state-of-the-art on the QUORA and MSCOCO paraphrasing datasets. Besides quantitative results, we presented examples that highlight the syntactic and semantic quality of our generated paraphrases.

In the future, it will be interesting to apply this framework for the task of abstractive text summarization and other NLG-related problems.

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. The research reported in this paper was conducted at the University of Lethbridge and supported by Alberta Innovates and Alberta Education.

References

Marianna Apidianaki, Guillaume Wisniewski, Anne Cocos, and Chris Callison-Burch. 2018. Automated paraphrase lattice creation for hyter machine translation evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 480–485.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 675–686.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question

answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

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*.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 410–413.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885.

Raymond Kozlowski, Kathleen F McCoy, and K Vijay-Shanker. 2003. Generation of single-sentence paraphrases from predicate/argument structure using lexico-grammatical resources. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 1–8. 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. *Text Summarization Branches Out*.

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.

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 R McKeown. 1983. Paraphrasing questions using given and new information. *Computational Linguistics*, 9(1):1–10.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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*.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 142–149.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, volume 1, pages 193–199.

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.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Shiqi Zhao, Cheng Niu, Ming Zhou, Ting Liu, and Sheng Li. 2008. Combining multiple resources to improve smt-based paraphrasing model. In *Proceedings of ACL-08: HLT*, pages 1021–1029.