A COMPARISON OF NEURAL NETWORK ARCHITECTURES FOR UNSUPERVISED PARAPHRASE GENERATION

A PREPRINT

Jacob N. Gursky, B.A.
Saint Olaf College
Northfield, MN 55057
gurskyjacob@gmail.com

Matthew Richey, Ph.D
Department of Math/Stat/Comp Sci
Saint Olaf College
Northfield, MN 55057
richeym@stolaf.edu

May 20, 2019

ABSTRACT

Neural paraphrasing is a challenging task in machine learning that requires semantically accurate vector representations of a text in order to reconstruct the target sentence with expressionally diversity. Using a Sequence-to-Sequence architecture, sentences are first encoded as a thought vector, then reconstructed to back to the source using a decoder. We compare the performance of two popular sentence embedding techniques, sequential autoencoding and skip-thought vectors in semantic representation, diverse paraphrase generation tasks, and the structure of the underlying thought vector distributions. We also explore an efficient framework for training networks, preserving the property of proximal semantic-relatedness for paraphrase generation.

1 Introduction

Paraphrase generation is a challenging task within the domain of natural language processing that describes the generation of semantically similar texts with variation in expression. Given a source sentence, a set of paraphrases must be generated that are similar in meaning to the source but display variation in structure and language. Advancement in paraphrase generation is an important milestone in neural text generation, as it is a proxy task for semantic-understanding and potentially long-sequence texts which are intelligible. Historically this problem has been approached using a variety rule-based and statistical translation approaches [21, 3], but recently recurrent neural networks have become the ideal tool for paraphrase generation [14]. Work has been done in both supervised and unsupervised paraphrase generation [3, 6, 9, 11, 10, 17], though we contend that an unsupervised approach is ideal, as paraphrase generation is a non deterministic process and there is a vastly larger body of unlabelled text than labelled paraphrase pairs. Evaluating the quality of a set of paraphrases is another difficult problem associated with paraphrase generation. The current set of metrics such as BLEU and METEOR [16, 2], are useful in determining the quality of translations in relation to a human translator, but evaluating the semantic-relatedness of a paraphrase set to its source sentence is a more challenging problem. Some proposed methods include evaluating the similarity of sentence representations [19], but this does not address the trade-off between semantic-relatedness and expression diversity.

2 Literature Review

There exists a wide body of literature currently on supervised paraphrase generation, however the corresponding body the unsupervised counterpart is relatively sparse. Much of the current literature involves the usage of a few large annotated paraphrase datasets such as Microsoft's Paraphrase dataset [5]. We believe this approach is limiting because it requires expensive data collection and the resulting paraphrase generation becomes a deterministic process rather than the unsupervised paraphrase generation approach we propose. However, insights into optimal network architectures and training procedures can still be gained based on prior literature.

2.1 Sequential Autoencoders

Following the Sequence-to-Sequence architecture proposed by [20], one can express any arbitrarily long text as a fixed-length 'thought vector', using an encoder-decoder architecture. One widespread application of this architecture is that of Neural Machine Translation [1]. The choice of recurrent unit in the encoder and decoder networks has an impact on the performance and training efficiency of the networks. Chung et al. showed that Gated Recurrent Units [4] are a viable alternative to LSTM (Long Short-Term Memory) cells, with the advantage of being much more computationally efficient. Using an unlabelled corpus of text, we can train a sequential autoencoder to project a given sentence as a thought vector using the encoder network, then reconstruct the source sentence using the decoder. In the context of paraphrase generation, the goal is to be able to have a thought vector representation that allows for an accurate reconstruction of the source sentence, but also to generate diverse paraphrases when noise is applied. However, this requires that the encoder learn the semantic meaning of sentences and demonstrate the property of proximal semantic-relatedness, rather than learning an individual encoding for each sentence with no spatial relation. In the absence of proximal semantic-relatedness, it is difficult to see how a set of paraphrases with a tunable tradeoff between semantic-relatedness and expressionally diversity can be achieved.

$$\sum_{t} logP(w_i^t | w_i^{<i}, h_i) \tag{1}$$

Equation 1: Objective function for sequential autoencoders

2.2 Skip-Thought Vectors

While it is likely that sequential autoencoders are able to translate the semantic relatedness of similar sentences into the thought vector space, [9] propose an encoder-decoder architecture called skip-thoughts that retain this property. Rather than reconstructing the source sentence from the thought vector, Kiros et al. draw inspiration from the Word2Vec methodology of [13]. The skip-thought architecture is comprised of an encoder and two decoders, one that attempts to reconstruct the sentence preceding the source sentence and another that attempts to reconstruct the sentence following the source sentence. The loss of the skip-thought network is defined as the sum of the average cross-entropy loss of both decoders. After training, the decoders are removed and one is left with an encoder that is able to accurately represent the semantic meaning sentences. Kiros et al. showed that these sentences embeddings outperformed all alternative state-of-the-art methods other than extremely costly networks, such as dependency-tree LSTMs.

An essential step in paraphrase generation is being able to represent the semantic meaning of a sentence as a thought vector, which is often done using a neural network with an encoder-decoder architecture [20]. First an encoder is defined that learns to express the source sequence as a fixed-length thought vector, while a decoder is defined that learns to construct a sequence conditioned on the thought vector from the encoder. After training the decoder network is removed and we use the encoder portion for sentence representations. There has been much success in using networks with LSTM and GRU units [17, 4], as well as a fine-tuning approach using reinforcement learning [10].

$$\sum_{t} logP(w_{i+1}^t | w_{i+1}^{<i}, h_i) + \sum_{t} logP(w_{i-1}^t | w_{i-1}^{<i}, h_i) \tag{2}$$

Equation 2: Objective function for skip-thoughts

3 Unsupervised Paraphrase Generation with Proximal Semantic-Relatedness

We propose a slight reformulation of the task of paraphrasing. Rather than generating a set of paraphrases from a source sentence that are nearly identical in semantic similarity, we propose an expansion on this property. A set of paraphrases can be drawn from a continuous space surrounding the thought embedding of a source sentence, which represents the meaning of a sentence as a fixed-length vector. We propose that a key property of thought vectors for paraphrase generation is that thought vectors that occupy a similar point in space must encode semantically-similar sentences. We call this property "proximal semantic-relatedness".

A paraphrase is generated by applying gaussian noise to the thought vector, and a paraphrase is generated from the resulting vector. A trade-off is incurred in the amount of noise applied to the original thought vector: the more noise that is applied, the less semantically-similar the resulting paraphrases will be to the source sentence, but the more expressionally diverse the resulting set will be. For example, given a source sentence "I saw a dog in the park today," a paraphrase generated with a small amount of noise may be "Today, I saw a dog in the park.", which is nearly identical to the source sentence but with some diversity. However, if we noise the source thought vector further, a resulting paraphrase may look something like "Today there was a dog in park," which is less semantically-related.

We compare the performance of two encoding architecture in generating paraphrases in an unsupervised fashion: skip-thoughts and sequential autoencoding. We contend that a skip-thought architecture best preserves proximal semantic-relatedness, whereas a sequential autoencoding approach is not able to capture sentence meaning and will solely be useful in exact reconstruction of the source sequence. Furthermore, we posit that the underlying distribution of skip-thought vectors holds a useful structure in clustering sentence meaning in an unsupervised fashion. Also key to training sentence representations is efficiently training such networks, as generative networks often take a relatively long time to learn.

4 About the Data

The corpus for our study comes from Project Gutenberg, comprising over 3,500 classic works of literature [18]. When tokenized the corpus contains 14 million sentences of contiguous text, a necessity for skip-thought modeling. We preserved the 50,000 most frequent tokens, though future studies should explore expanding this, with a maximum sequence length of 30 tokens. Note also that all punctuation was padded with whitespaces to create distinct tokens and all tokens were converted to lowercase. Due to resource constraints, we subsampled from the data, with 240K sentences composing our training set and 60K sentences composing our validation set.

5 Methods

5.1 Networks Examined

In order to determine an efficient framework for thought vectorization with the goal of unsupervised paraphrase generation, we examine two encoder-decoder architectures: sequential autoencoders and skip-thoughts. Both networks are composed of an encoder with three layers of 256 GRU units, project to a thought vector of length 256, and decoders with two layers of 256 GRU units. Logeswaran et al. argue the need for a deeper encoder than decoder to encourage higher-quality thought vectorizations. Tokens are embedded using an embedding layer of 256-dimensions. Models were trained using the Adam optimization algorithm proposed by Kingma et al., a popular choice in training recurrent networks. We trained the network until the validation loss did not improve for five epochs. All networks were trained on a single NVIDIA 980ti graphics card using Tensorflow 2.0. Note that our network and training set sizes were hampered quite a bit by memory limitations.

5.2 Paraphrasing and Embeddings

In order to generate a set of candidate paraphrases, noise is applied to the thought vectors of the contending networks from a Gaussian distribution of mean zero and variance sigma. Note that as sigma increases, we expect the resulting paraphrases to be more diverse but less semantically-related to the source sentence. We explore different values of this hyperparameter and how its effect differs between architectures.

To generate paraphrases after training the skip-thoughts encoder using the process outlined by Kiros et al., a decoder must be trained that attempts to reconstruct the sources sentence using the skip-thought representation. Note that when training the final decoder only the decoder weights are defined as trainable, as the skip-thought encoder is already assumed to accurately capture the meaning of sentences with the property of proximal semantic-relatedness.

6 Results

6.1 Example Paraphrase Sets

We can see below in Table 1 an example set of paraphrases given several candidate sentences using both sequential autoencoding and skip-thought vectors for reconstruction. Given that paraphrases are generating by applying Gaussian noise with variance σ, we examine two tasks in the table above. When σ is set to zero the task becomes autoencoding of the source sentence, so one would expect a perfect reconstruction of the source sentence. We can see that this property is more strongly preserved in the sequential autoencoder, as the encoder portion is trained to reconstruct the source sentence but not necessarily semantic meaning.

Also of interest is the poor quality of paraphrases generated by the skip-thought model. As noted by Kiros et al., skip-thought encoders require relatively large datasets to achieve satisfactory performance. It is apparent than a training set with 240K rows is not sufficient for a corpus as diverse as ours. We can see however, initial signs of proximal semantic-relatedness being demonstrated by skip-thoughts vectors. For instance, one candidate paraphrase

for the source sentence "What did you do yesterday?" is "What have you been doing lately?", which demonstrates the semantic-understanding property of skip-thoughts. It is highly probable that with a larger dataset more coherent paraphrases will be generated.

We also note the presence of an "island of stability" for paraphrases generated by sequential autoencoding. It seems that candidate paraphrases generated tend to be identical to the source sentence with Gaussian noise applied with σ close to 0. However, we find that once a certain threshold is reached with high sensitivity, the network completely dissociates the thought vector from the source sentence and the original meaning lost. We theorize that this is due to the nearest neighbor of a noised thought-vector changing from the source thought vector to an unrelated thought, which then yields completely semantically distinct paraphrases as sequential autoencoders do not preserve proximal semantic-relatedness.

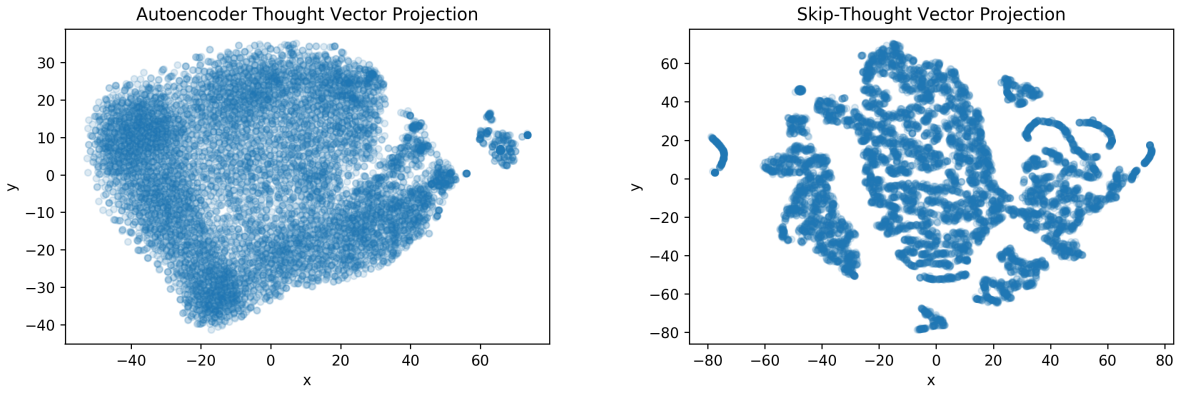
Table 1: Examples of Generated Paraphrases with Varying Noise

Source	σ	Autoencoder Paraphrase	Skip-Thought Paraphrase
He is a good man.	0	*he is a good man . <end>*	*he is a very good . <end>*
	0.1	*he is a good man . <end>*	*i never <UNK> .] <end>*
	0.2	*he is a good man . <end>*	*he asked , looking up . <end>*
What did you do yesterday?	0	*what did you do yesterday ? <end>*	*what have you been doing ? <end>*
	0.1	*what did you do yesterday ? <end>*	*what is it you ' re doing it ? <end>*
	0.2	*and did you saying ? <end>*	*" <UNK> ! <end>*
John asked her a question!	0	*john asked her a question ! <end>*	*he asked , with silence ! <end>*
	0.1	*john asked look any was careful . <end>*	*i am <UNK> . <end>*
	0.2	*returned lady her him question to ! <end>*	*said the skipper . <end>*
Do you have any cats?	0	*do you have any cats ? <end>*	*" what is the matter ? " <end>*
	0.1	*do you have any group ? <end>*	*" you . <end>*
	0.2	*do you by you . <end>*	*<UNK> <UNK> , do you mean ? " <end>*
John saw her in the park.	0	*john saw her in the park . <end>*	*he was a little man of her ! <end>*
	0.1	*john saw her in the park . <end>*	*the <UNK> is the only one , and i can do it , i assure you . " <end>*
	0.2	*john saw her in the park . <end>*	*i looked at the <UNK> <UNK> . <end>*

6.2 Thought Vector Distributions

In order to determine the effectiveness of our thought-vectorizing encoder, it is important to examine the distribution of the resulting vectors. Following the work by Kiros et al., who showed the semantic clustering properties of skip-thought encoders, we use t-SNE projections of 10,000 sequential autoencoding thought vectors and skip-thought vectors from our training set. Theoretically we would expect underlying structure to the thought vector distribution to have some structure and clustering if it preserves the property of proximal semantic-relatedness. Looking at Figures 1 and 2 above, we can see the distributional differences between the two contending encoders.

Whereas the t-SNE projection of the skip-thought vectors display the underlying structure and clustering reminiscent of the study done by Kiros et al., the projections of the sequentially-autoencoded thought vectors appear to lack this structure. We theorize that this is due to the sequential autoencoder learning a unique thought vector for each sequence without the proximal semantic-relatedness property, leading to a nebulous distribution with no structure. This is reinforced by our finding of the "island of stability" for paraphrases generated with an autoencoder, as once excessive noise is applied to the thought vector and the nearest neighbor changes, a completely semantically separate thought is expressed. However, this does not seem to be the case for skip-thought vectors, where a relatively larger amount of noise is required to shift a thought vector away from its cluster of related sentences.



(a) t-SNE projection of sequentially autoencoded thought vectors

(b) t-SNE projection of skip-thought vectors

6.3 Neighborhood-Aware Noise

As noted above, the proper tuning of σ for generating paraphrases is a key component of continuous-space unsupervised paraphrase generation. However, as shown by the t-SNE projections of the thought vector space above, proper structure of networks with proximal semantic-relatedness organizes similar thoughts into clusters of which a multivariate gaussian distribution cannot be assumed. During our paraphrase generation we apply zero-centered gaussian noise, but as shown by the paraphrase sets for skip-thought vectors this can either generate a coherent paraphrase or completely fail to represent the original thought. We believe that this is due to the randomization inherent in applying noise to a vector belonging to a non-gaussian cluster. We posit the successful construction of a paraphrase using a noised thought vector occurs when the source vector is sufficiently shifted away from the source thought but remains within the local thought cluster. We propose that a failed paraphrase occurs when noise is incorrectly applied that shifts the source vector to a neighboring, but significantly semantically-distinct, cluster of thoughts. We therefore propose that in future studies an approach to applying noise in a local thought cluster-aware fashion be examined in order to mitigate this problem.

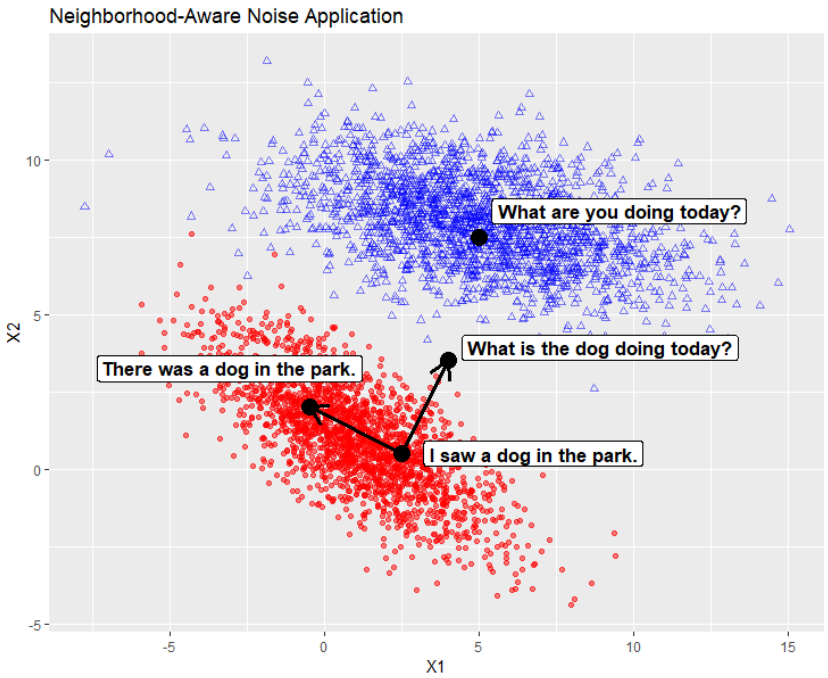


Figure 2: An example of gaussian noise causing a noised thought to deviate from the local thought cluster

7 Conclusions

Using a novel reformulation of the task of paraphrasing, we have introduced and demonstrated the importance of proximal semantic-relatedness in thought vectors for paraphrase generation. While resource constraints led to a training set of insufficient size for coherent paraphrases, we have demonstrated that skip-thought vectors more strongly preserve this property than sequentially-autoencoded thought vectors. We have also shown that the underlying structure of skip-thought vectors yields useful clustering that associates related thoughts, whereas sequential autoencoding thought vectors do not have the necessary structure for paraphrase generation. Furthermore, we have shown the role of the properly tuning σ of the gaussian noise applied to the thought vector during paraphrasing.

8 Future Study

8.1 Network Architecture

Given the resource constraints faced during our study, future research should experiment with deeper architectures. Prakash et al. showed that deep residual LSTM networks perform well in supervised paraphrase generation. Furthermore, Kiros et al. also showed that the quality of thought vectors can be improved by using bidirectional skip-thought encoders. Also of interest would be the performance difference of Gated Recurrent Units and Long Short-Term Memory cells in paraphrase generation, though GRU cells are conceptually simpler and therefore enable deeper architectures. We also propose that utilizing pre-trained contextual embeddings, such as those proposed by McCann et al., will speed up training improve model performance considerably.

In addition to the initial maximum-likelihood training of the encoding network, Li et al. showed that a fine-tuning approach using reinforcement learning can improve the quality of the final paraphrases. This fine-tuning approach could also theoretically be applied in an unsupervised context. We recommend that first a traditional skip-thought encoder be trained, then using the approach proposed by Li et al. or a sequential GAN [15] to fine-tune the paraphrase-generating decoder.

8.2 Quick-thought Vectorization

One key limitation of skip-thought vectors is the computational constraints. In their original paper, Kiros et al. required approximately two weeks for their skip-thoughts network to converge. Logeswaran et al. show that by shifting from a generative approach to a discriminative approach for the decoders, one can reduce the training time drastically and improve overall embedding quality. Rather than attempting to reconstruct the surrounding sentences token by token, the quick-thoughts architecture proposes selecting the true sentence from a set of randomly sampled possibilities. They show that in an identical task, quick-thought vectors converge roughly 31 times more quickly than their skip-thought counterparts. We recommend in future studies that quick-thoughts be used in lieu of skip-thoughts, as it allows for deeper network architectures with shorter train times and higher quality embeddings.

References

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Banerjee, Satanjeev, and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.*, 2005.

[3] Barzilay, Regina, and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics.*, 2003.

[4] Chung, Junyoung, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] Dolan, William B., and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. *Proceedings of the Third International Workshop on Paraphrasing (IWP2005).*, 2005.

[6] Gupta, Ankush, et al A deep generative framework for paraphrase generation. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] Iyyer, Mohit, et al. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

[8] Kingma, Diederik P., and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Kiros, Ryan, et al. Skip-thought vectors. *Advances in neural information processing systems.*, 2015.

[10] Li, Zichao, et al. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*, 2017.

[11] Logeswaran, Lajanugen, and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[12] McCann, Bryan, et al. Learned in translation: Contextualized word vectors. *Advances in Neural Information Processing Systems.*, 2017.

[13] Mikolov, Tomas, et al. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems.*, 2013.

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

[15] Mogren, Olof. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[16] Papineni, Kishore, et al. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics.*, 2002.

[17] Prakash, Aaditya, et al. Neural paraphrase generation with stacked residual LSTM networks. *arXiv preprint arXiv:1610.03098*, 2016.

[18] Project Gutenberg www.gutenberg.org.

[19] Quan, Zhe, et al. An Efficient Framework for Sentence Similarity Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[20] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems.*, 2014.

[21] Zhao, Shiqi, et al. Application-driven statistical paraphrase generation. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. Association for Computational Linguistics*, 2009.