Paraphrase Generation Using Deep Generative Models
Final Project Report CS-772

Nishit Asnani
14433
nishit@iitk.ac.in

Kanishk Gandhi
14235
kanishkg@iitk.ac.in

Kumar Kshitij Patel
150348
kishinmh@iitk.ac.in

Abstract

Among the most enticing things about natural language is the fact that it can be written and spoken in myriad ways. The act of paraphrasing, or saying something that has the same meaning as the original, but in a slightly different manner, has a plethora of possible applications, for e.g. sentence simplification, complication, reporting, vocabulary modulation, tone regulation. We provide a novel deep generative model for paraphrase generation using a variational auto-encoder. We also suggest a method for controlled paraphrase generation using pre-fixed latent variables. We obtain promising results on machine translation & paraphrase evaluation metrics and some interesting correlations in human evaluation. Our results are qualitatively and quantitatively comparable. We also report the current issues with paraphrase evaluation methods, and analyze the popular datasets in that respect.

1 Problem Motivation

Spoken and written content can be formulated in myriad ways to make it accessible to a diverse set of audiences. A classic by Charles Dickens would need its sentences to be expressed in a simpler format for children to be able to understand it. On the other hand, a children's story would need to be adapted in a more mature language with a wider variety in sentence structure and vocabulary in order to be appealing for adult readers. Reminders and queries can be rephrased to make them seem more actionable. Computer agents communicating with humans can use multiple wordings of the same intent to elicit the desired response from the interacting person. These demands can be fulfilled by an efficient paraphrase generation system, if trained on a proper dataset.

Deep learning has found applications in a variety of domains in the recent years. The availability of massive datasets and plenty of computational resources have been instrumental in aiding deep learning algorithms to find solutions to problems that were a long shot earlier. Its influence on problems in natural language processing has been immense as well, right from state of the art sentiment classifiers and spreading its net wide enough to entail impressive language understanding and generation techniques. Paraphrase generation is one of the many sequence to sequence generation tasks that have been tackled by deep learning. Although decent models exist, much work remains before we can arrive at industry grade paraphrasers that are expressive in and adaptable to a wide range of domains.

Probabilistic modeling of classification tasks has existed since decades, but its combination with deep neural nets has led to a recent surge in innovations leading up to deep generative models. Variational Autoencoders (Kingma and Welling, 2013) and Generative Adversarial Networks (Goodfellow et al., 2014) have been some of the more popular models. While their utility and applications in natural language tasks is being studied, a flurry of work on VAE modifications to be adaptable to sequence to sequence learning has appeared over the last couple of years.

Solving the problem of paraphrase generation using variational autoencoders thus seemed like an interesting idea for a project, and we went ahead with it. The recent work by (Gupta et al., 2017) has been a guiding light, since it has tackled the same problem, though with a different architecture.

2 Related Work

2.1 Paraphrase Generation

Generating paraphrases can be thought of as a sequence transformation problem. Traditionally, the task received a lot of attention from researchers in the natural language processing domain using techniques in statistical machine translation like work by (Tomuro, 2003) and (Zhao et al., 2008) that rely on rules to generate paraphrases for questions.

More recently techniques in sequence to sequence (seq2seq) learning (Sutskever et al., 2014) have been popularly used to tackle Natural Language Processing tasks, giving promising results in an array of problems. (Cao et al., 2017) use a sequence to sequence model, that focuses on the two basic tasks of copying and rewriting for the generation of paraphrases. (Prakash et al., 2016) improves upon a standard seq2seq model by using a stacked residual LSTM in the encoder and decoder modules. Several variations in this class of architectures have also involved the use of attention networks (Bahdanau et al., 2014) (Luong et al., 2015) and bidirectional layers resulting in marginal alleviation of results (Graves et al., 2013) (Schuster and Paliwal, 1997). Attention architectures seem intuitively significant on account of focusing certain words to produce a target Using deep RL for improving the training of traditional seq2seq models by directly optimizing the BLEU metric. It can be seen to dramatically improve the model performances on the downside of the required training time. In recent times, work on generative alternatives like using VAEs or GANs has been an active field of research.

2.2 VAEs for Natural Language

Variational Auto-Encoders (Kingma and Welling, 2013)(Rezende et al., 2014) have been a popular framework for modeling problems in the domain of computer vision. Drawing inspiration from these models, (Bowman et al., 2015) tried to model sentences in a continuous space with a Variational Auto-Encoder and have achieved decent results in the task of language modeling. Important techniques like the use of word dropouts in the decoder, KL cost annealing and fully connected layers for transformations in the latent space are key to training VAEs for natural language. Hu et. al. (Hu et al., 2017) append information in the latent space of a Variational Auto-Encoder to control the text generated by the VAE. Using GANs for generating text has still been an elusive problem with some attempts (Rajeswar et al., 2017) at trying to solve the problem being partially successful. The chief advantage of using VAEs as found in (Gupta et al., 2017) is the ability to generate several paraphrases conditioned on the same input. In our work, we take inspiration from the success of (Gupta et al., 2017) and the success of VAEs in transforming attributes of images conditioned on an input (Yan et al., 2016) to control certain semantics of the generated paraphrase while also conditioning it on the input sentence.

3 Methodology

3.1 Variational Auto-encoders

Variational auto-encoder (Kingma and Welling, 2013)(Rezende et al., 2014) is a deep generative latent variable model, which has been used extensively in the last three years for many machine learning applications(refer (Doersch, 2016), 2016 for an extensive survey). It has two components just like a normal autoencoder, an encoder that learns and maps the high dimensional input $\mathcal{X}$ into a low dimensional and rich latent space $\mathcal{Z}$, which is later used for reconstruction. Unlike an auto-encoder model which learns a deterministic latent coding, VAE learns a posterior distribution over the latent space $q_\phi(z|x)$. It is usually a diagonal Gaussian, which means that the parameters are defined by $\phi = \{\mu(x), \sigma^2(x)\}$. For the decoder part, it learns another function $p_\theta(x|z)$, which generates desired output using the learned encodings. These parameters (ϕ, θ) are learned using a neural network, suitable to the task in hand and input-output required. Parameter learning happens through the variational Bayes inference method, which defines the required ELBO function as:

$$\mathcal{L}(\theta, \phi, x) = \mathbb{E}_{q_\phi(z|x)}[p_\theta(x|z)] - \mathcal{KL}(q_\phi(z|x)||p_\theta(x|z)) \tag{1}$$

Here KL represents the KL-divergence. Just like normal VB inference the parameters here are obtained by maximizing this lower bound. The first term captures the reconstruction error while the second one keeps the proposal and prior close to

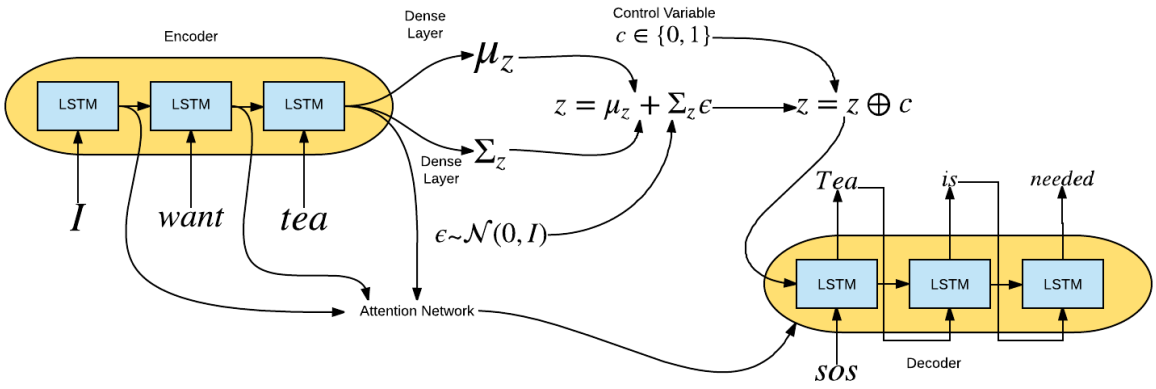

Figure 1: Controlled SIDVAE - model architecture

each other. Being a deep latent Gaussian model, we can produce realistic looking input through a VAE. This has been used extensively in computer vision, and (Bowman et al., 2015) tried to model sentences in a VAE framework using long short-term memory (LSTM) networks. Our model is very similar and is described below.

3.2 Model Architecture

Our high level model architecture broadly follows that of a Variational Autoencoder, but since it is being trained for a natural language task, suitable adjustments have been made. Given a source sentence - paraphrase pair $src[0 : L_s], tgt[0 : L_t]$, with L_s and L_t being the lengths (in words) of the source and the target sentence respectively, our aim is to generate a close approximation of tgt. Here, catering to the demands of our task, we care about the generated sentence gen preserving the meaning of the source, but also being distinct from it in terms of structure and/or vocabulary, in order for it to be called a proper paraphrase.

We use an LSTM network as an encoder over the inputs, and another LSTM network as a decoder to produce the output. The encoder corresponds to $q(z|x)$, the recognition function that models the distribution of latent variables z given the observed data x. The encoder's final cell state is used to produce $\{\mu(z|x), \Sigma(z|x)\}$, the parameters of $q(z|x) = \mathcal{N}(\mu(z|x), \Sigma(z|x))$.

In the basic VAE model, which we call Say-It-Differently VAE (SIDVAE), we sample a random z from this distribution by using the re-parameterization trick. This is then fed to a decoder network $p(y|z)$, that generates the output y, which is a sequence of words that we term gen. This model suffers a combination of the cross entropy loss at each time step of generation, and the KL divergence loss between $q(z|x)$ and $\mathcal{N}(0, I)$, as elucidated earlier. The cross entropy loss is given by:

$$CE = \Sigma_w T_t(w) \log P_t(w)$$

where CE stands for cross entropy loss, $\{w\}$ is the set of words in the vocabulary, $T_t(w)$ is the one hot vector over the vocabulary representing the target word at timestep t, and $P_t(w)$ is the probability distribution over the set of words predicted by the decoder network at timestep t.

The model diagram has been provided here to give a succinct overview.

In another model, we use a control variable c to control for sentence length while generating a paraphrase. This is called Controlled SIDVAE (CSIDVAE), which is trained to produce short paraphrases when the control variable c is set to 0, and longer ones when c is set to 1. Here, z is augmented with c, and we have,

$$z \leftarrow z \oplus c$$

And this z follows along to the decoder as stated earlier. The dataset is modified to include a control variable with each input and a correspondingly smaller / larger sentence in the target.

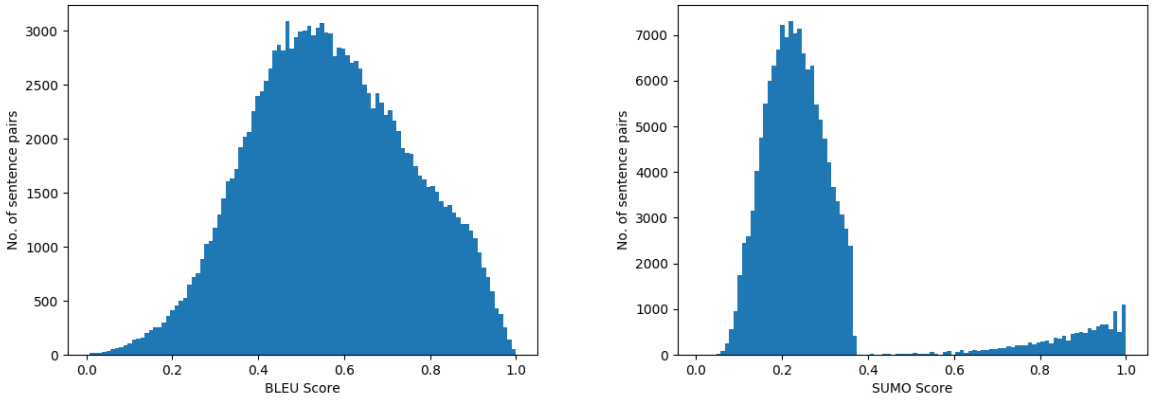

Figure 2: Evaluation of the Quora Dataset against BLEU and SUMO

4 Experiments

4.1 Training Tweaks

Since a VAE model for text generation is not as straightforward to train (and we did encounter many of the problems that we had anticipated), we follow some techniques illustrated by (Bowman et al., 2015) to make it work for our experiments.

Firstly, we use **KL cost annealing**, which essentially means that the weight given to the KL divergence loss is varied in a sigmoidal fashion over the number of training steps, starting close to zero and going up to 1. We also use **word dropout**, which leads to dropping words at certain timesteps in the decoding phase with a predefined probability, and allowing the model to decode only based on the latent representation z on those timesteps. This allows the model to learn richer representations of the input in the latent space that z is drawn from. Both these techniques are borrowed from Bowman et. al., and are critical for the model to be trained.

We also found it useful to increase the weight given to the cross entropy loss relative to the KL divergence by a constant factor. This helped training, as the model initially focused all its attention in replicating the target, and eventually moved on to maintaining a balance between that objective and staying close to the prior. Thus, our loss is given as:

$$\mathcal{L} = \eta * CE - \alpha * \mathcal{KL}(.,.)$$

where, η is the CE **scale up factor**, generally set to 10, and α is the KL cost annealing factor, which varies as a sigmoid, as described above.

4.2 Datasets

The validation for the architectures is done using three datasets that provide paraphrases or approximate paraphrases of the source sentence. The three data sets are as follows:

4.2.1 MSCOCO (Lin et al., 2014)

This dataset is popular for image captioning tasks. Most of the image captions describe the image in terms of its composition containing varying degrees of information while also varying in complexity of language used. The captions for the images are approximated as paraphrases of each other. The dataset contains 165k images captioned by 5 annotators. Two pairs of paraphrases (input-target pairs) are generated from the 5 captions by omitting one caption. We therefore get 330k training captions. A Train-Val-Test split of 97%-1%-2% for training and evaluation is chosen. A high amount of data is used for training as we were unsure if the data would be sufficient.

For training the models based on control, we map the shortest and longest captions to each other for a 'create a long paraphrase' control parameter while the shortest and the second shortest caption are mapped for the 'create a short paraphrase' control parameter.

4.2.2 Quora (Quo)

The Quora question answer dataset contains paraphrases of questions that have a similar mean-

Table 1: Evaluation on BLEU metric

Model	Number of Layers	Bleu	Beam Size	Vocabulary Size	Data
Seq2Seq	2	16.5	10	30332	COCO
Attention	2	18.6	10	30332	COCO
VAE (Ours)	2	19.0	1	10000	COCO
VAE (Ours)	2	21.0	10	10000	COCO
VAE (Ours)	2	22.0	10	10000	Quora
VAE (Ours)	2	44.0	10	10000	SimpleWiki

ing. This seems like a more practical data set for paraphrase generation. Scraped and sampled by Quora, the data contains potentially 400,000 lines of duplicates. 149K input-target pairs of paraphrased questions are used for training and evaluation of the model. We use a Train-Val-Test split of 90%-5%-5%

4.2.3 Simple Wiki (Hwang et al., 2015)

The simple wikipedia is a version of the wikipedia encyclopedia written in a simpler form of English. We use 1-to-1 sentence mappings from the simple Wiki data to the original wikipedia to model paraphrase generation. This data is particularly useful as the paraphrases are explicitly simpler versions of the original input. The disadvantage of using this dataset is the large number of duplicate source and target pairs present. The number of pairs are 154k with a Train-Val-Test split of 97%-1%-2%

4.3 Baselines

A vanilla seq2seq model with 2 layers is used to find baseline scores. Another variant consists of using a model with attention.

4.4 Experimental Setup

For our experiments, we use 2 LSTM layers at both the encoder and decoder sides with 512 units in each layer. Our latent variable space is also 512 dimensional. We use ADAM to train our model, with a learning rate of 0.001. Scaled Luong (Luong et al., 2015) attention model is used.

4.5 Evaluation

Merriam Webster's dictionary defines paraphrase as *"a restatement of a text, passage, or work giving the meaning in another form"*. In the light of this definition we need to consider three different factors while evaluating the quality of paraphrases (Liu et al., 2010):

1. **Adequacy:** It refers the amount of semantic co-relation, and hence equivalence. The paraphrases should completely convey the meaning and not add additional meaning to the sentence.

2. **Lexical Dissimilarity:** Since we want to generate paraphrases, copying the same sentence is not acceptable. There must be some difference in the writing style, structure etc.

3. **Fluency:** It refers to the readability, clarity and continuity of a sentence. It should neither be grammatically absurd nor very complicated lingually.

4.5.1 Quantitative and Qualitative Analysis

Usually, the machine translation metrics have been used in literature for evaluating the quality of paraphrases generated. (Madnani et al., 2012) showed that many of them actually have a high correlation with human judgment and/or explain the variability in the data well. We tested our results using the most commonly used metric BLEU (Papineni et al., 2002) which considers exact match between reference paraphrase(s) and system generated paraphrase(s) using the concept of modified n-gram precision and brevity penalty. However, there has been a lot of criticism surrounding the score mainly regarding the fact that in natural language the sentences which are semantically close might not have an appreciable amount of overlap. Essentially, something like BLEU would not reward lexical dissimilarity, and can underrate some really good paraphrases too.

To overcome this difficulty many task specific metrics have been defined in the recent years ((Liu et al., 2010), (Callison-Burch et al., 2008) etc.). Since there isn't clarity on which is essentially better, we also report the results for SUMO (Liu et al., 2010), a metric which doesn't suffer from this one issue which BLEU does. It has also been

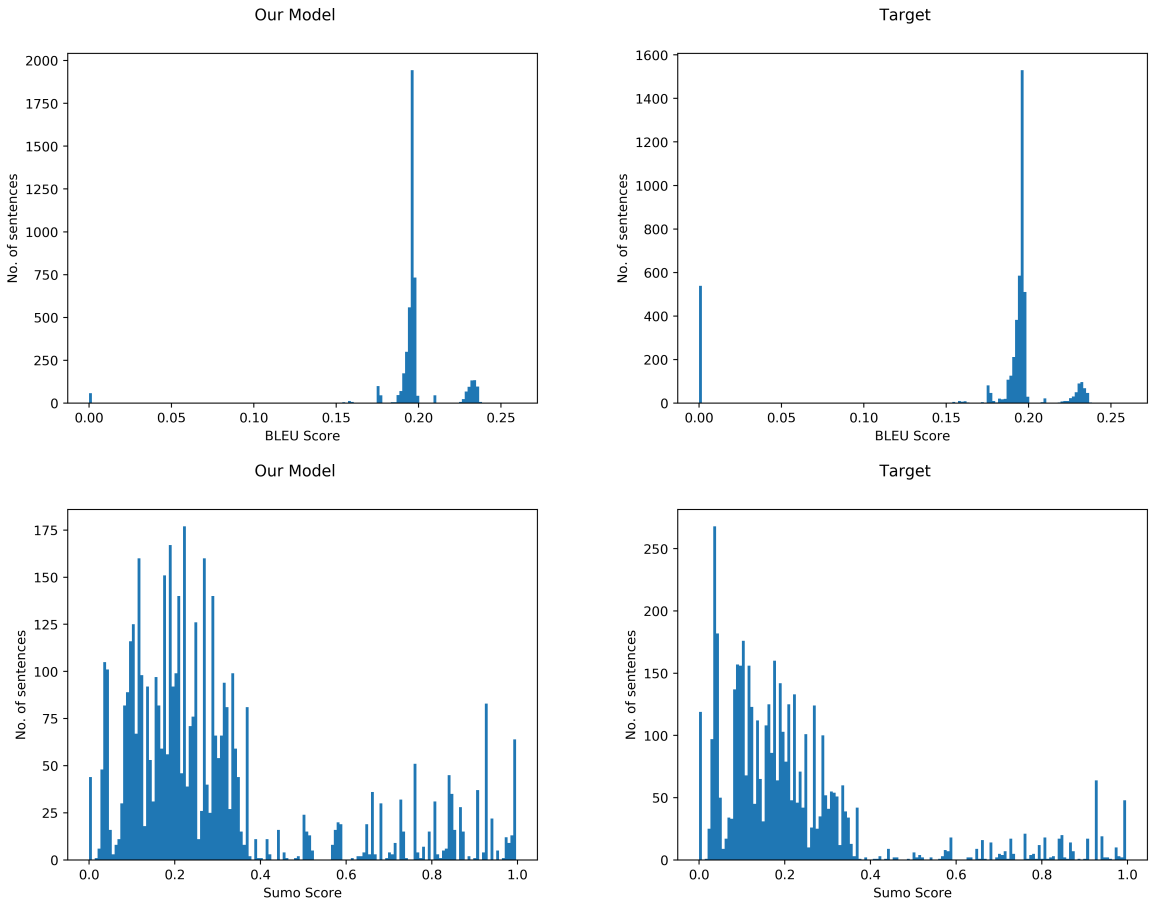

Figure 3: Evaluation of the MSCOCO target sentences and our paraphrases

Table 2: Evaluation of quora dataset on different metrics

Original sentence	Paraphrase	BLEU score	SUMO score	Remark
What is a narcissistic personality disorder?	What is narcissistic personality disorder?	0.93	0.26	BLEU stupidly marks a duplicate high
How I can speak English fluently?	How can I learn to speak English fluently?	0.70	0.75	Similar Scoring on a decent sentence pair
What are the application of binary search trees?	What are the applications of binary search trees?	0.94	0.23	Almost duplicate sentences marked good by BLEU
Why does 0! Equal 1?	Why the factorial of 0 is 1?	0.17	0.97	A good but lexically dissimilar paraphrase rated bad by BLEU

Table 3: Qualitative results on various datasets using the SIDVAE and the CSIDVAE architectures

Source	a person riding a horse near trees in the background
Generated	a woman is riding a horse on a sunny day a woman riding a horse down a trail a man riding on the back of a brown horse
Source	Daniel Ek: When is Spotify coming to India?
Reference	Daniel Ek: Why is spotify not available in India?
Generated	Why hasn't Daniel Ek brought Spotify to India?
Source	groening grew up in portland , and attended ainsworth elementary school and lincoln high school
Reference	groening grew up in portland , and attended ainsworth elementary school, lincoln high school and also the evergreen state college in olympia , washington
Generated	groening went to ainsworth elementary school and lincoln high school
Source	a couple of men riding motorcycles down a street
Short	a group of motorcycles riding down a city street
Long	a group of men on motorcycles on a road with trees in the background
Source	two girls are rollerblading on a city street
Short	two kids are rollerblading in front of a bus
Long	two girls are on roller skates with one holding up a supporting team sign

proven to correlate with human judgment better for the task of paraphrasing.

Table 1 shows the performance of some of the models on multiple datasets. It can be seen that we perform better as compared to normal seq2seq model as well as the model with attention. We have shown the distribution of score for the quora dataset and it can be observed that the distributions are somewhat similar except the higher peak for SUMO dataset. Since the quora dataset is a collection of duplicate sentences, most of it is suitable for paraphrase training. In fact on analysis the sentences with a higher sumo score are better paraphrases, while sentences these very sentences have a low bleu score. For instance, have a look at Table 3, which shows some archetypal examples. Thus, the BLEU score obtained which is lower than the state of the art should be taken with a pinch of salt, because neither is the data perfect nor is the metric. A similar issue occurs with the Simple wiki dataset. Another interesting result in found in MS COCO data set. Since it is an image captioning data-set, we don't expect very good paraphrases to be present in it, mainly because every image has a lot of information which captions may chose to ignore. So, the alternate captions won't be good paraphrases in the sense that they add or remove attributes. Figure 3. shows that our model is as good as the original dataset, any better BLEU score beyond that is a matter of suspicion in fact, because the model might have learned to do simple copying.

4.5.2 Human Evaluation

Human evaluation is performed on paraphrases generated by the model for the MSCOCO data. The important metrics deemed fit to measure the quality of a paraphrase in consistency with (Gupta et al., 2017) were readability of the paraphrase and its relevance to the input. For the qualitative analysis of the generated paraphrases, we generate paraphrases from the model and mix them randomly with ground truth reference paraphrases. 13 humans evaluators were asked to rate 30 source-target pairs on the relevance of the target to the source and the readability of the target on a 5-point scale. 15 of these pairs were ground truth references while 15 were generated by our model.

- Number Of Times Readability Better Than Reference: 5/13
- Number Of Times Relevance Better Than Reference: 8.5/13

It is found that the generated sentences rate higher on maintaining relevance but lower on readability, but the average rating for readability is quite close to the one for the reference sentences.

Table 4: Human Evaluation from 13 Users

User	Tgt Rel	Tgt Read	Gen Rel	Gen Read
Avg	2.37	3.79	2.55	3.82

5 Conclusions

Our deep generative models for paraphrase generation do well, both on quantitative as well as qualitative metrics, on three different datasets. This reaffirms the view that VAEs are a step forward in approaching natural language generation tasks. They allow the added advantage of naturally incorporating control variables to have a further grip on the type of paraphrases generated.

We also conclude, based on quantitative evidence, that popular metrics like BLEU are not very suitable for paraphrase generation, and others like SUMO could be used in mainstream research to judge paraphrase quality. It makes sense to probe this line of research further and come up with better metrics for this particular task, or even validate our initial findings about the inadequacy of BLEU with a solid statistical analysis.

6 Learning Experience

6.1 Insights and Learnings

We had a practical experience of making a deep generative model work for a complex text generation task. Even though the model is easy to implement, the issues that arise while training such a model over a vast text dataset are unique. We couldn't get our initial model to train well, but thankfully, (Bowman et al., 2015) had already figured out how to tackle them, and to see their tweaks work in action was a humbling experience. Also, we wanted to make the control based model as simple as possible, while retaining its capabilities for controlled generation, so we had to wrap our brains around modifying the dataset in a manner that the training mechanism wouldn't need a discriminator. We learnt the value of an attention mechanism when we saw our model improve drastically in terms of rate of convergence when we started using one.

The analysis of the evaluation metrics in general gave a good idea of what paraphrasing is doing, through which we verified how BLEU is actually bad at all the places it is cited for being bad. The loss function used for SUMO was a novel one which gave us some idea about developing an even better evaluation metric for paraphrase generation in future.

6.2 Tools Used

Our code mainly develops upon the neural machine translation repository by Google. We used python 2 along with TensorFlow to build and evaluate our models. For text preprocessing metric, NLTK library was used, while some basic implementations were taken from the Internet. The model was mainly run on the department GPU server.

6.3 Issues Faced

We faced the following issues while working on the project:

1. **Hardware Issues:** Owing to the queues in the department GPU server, we initially faced some issues in running our program till convergence. This rendered us unable to quickly iterate among various hyperparameter settings.

2. **Rapidly Decaying KL Divergence:** Initially our KL divergence loss nullified very rapidly (in a few hundred iterations). This essentially made the encoder obsolete and our model refused to learn from then on. We overcame the same by using a parameter to anneal the KL cost that varied as a sigmoid of a linear function of the number of steps, as suggested in (Bowman et al., 2015).

3. **Slow Model Convergence:** While comparing against the sequence to sequence model used for machine translation, our model could not converge to a significant BLEU score even after significant number of iterations. We shifted to the ADAM from SGD, while tuning our decay rate which solved this issue to some extent. We added the attention mechanism as well, which drastically improved the model's convergence rate.

4. **Dataset:** The simple Wiki dataset has many almost similar looking sentences. Same is the case with the Quora dataset which is even worse that way. In MS COCO dataset though the sentences are not duplicates but since they are image captions, a sentence might choose to ignore an attribute of the sentence which other mentions. This leads to generation of paraphrases which have completely new attributes and adverbials.

5. **Evaluation:** Having shown how bad can BLEU score actually be, we were rendered helpless because majority of literature cites the score for paraphrase generation while it is of limited value as shown above. Hence there are no baselines for any new metric. Moreover, even with the new metric there isn't any consensus yet on semantic equivalence, which is expected to come from the dataset itself. This assumption might fail in the MSCOCO dataset as mentioned above due to addition of extra information.

7 Future Work

We want to continue working on the project, since the problem of paraphrase generation is far from solved. In order to improve the generation of multiple legitimate paraphrases starting from a single input sentence, we plan on using a conditional VAE architecture (**?**). This would help make our model more sound and possibly better at using the "randomness" in the latent variable.

We also plan to train a bigger model with 4 hidden layers, and also use residual connections between them, so that we can compare our results with (Prakash et al., 2016) and see if we can do better with a generative approach. We also plan on trying out a larger model with more than 1000 dimensional latent spaces, in order to be able to compare with gupta2017deep who use 1100 dimensional latent spaces to train their VAE.

Finally, we wish to try and control our generation process even further, and see if we are able to do well on more meaningful controllable attributes than sentence length. This could entail controlling our generation on the level of complexity (which, in a limited sense, sentence length already does), urgency (actionable rephrasing) or sentiment. This could lead to many potentially useful applications for such paraphrase generation tools in industry.

References

???? Quora Paraphrase Dataset quora paraphrase dataset. https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs. Accessed: 2017-11-26.

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

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* .

Chris Callison-Burch, Trevor Cohn, and Mirella Lapata. 2008. Parametric: An automatic evaluation metric for paraphrasing. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 97–104.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *AAAI*. pages 3152–3158.

C. Doersch. 2016. Tutorial on Variational Autoencoders. *ArXiv e-prints* .

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. pages 2672–2680.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, pages 273–278.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2017. A deep generative framework for paraphrase generation. *arXiv preprint arXiv:1709.05074* .

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*. pages 1587–1596.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *HLT-NAACL*. pages 211–217.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

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*. Springer, pages 740–755.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. Pem: A paraphrase evaluation metric exploiting parallel texts. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 923–932.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 182–190.

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*. Association for Computational Linguistics, pages 311–318.

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* .

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929* .

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* .

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

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.

Noriko Tomuro. 2003. Interrogative reformulation patterns and acquisition of question paraphrases. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics, pages 33–40.

Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*. Springer, pages 776–791.

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