Character Based Language Models Through Variational Sentence and Word Embeddings

Kevin Bradley Dsouza and Zaccary Alperstein

1

Abstract

Language models have come of age recently with the introduction of Long-Short-Term-Memory based encoders, decoders and the advent of the attention mechanism. These models however work by generating one word at a time and cannot account for character level similarities and differences. In this project we propose a novel character based hierarchical variational autoencoder framework that can learn the word and sentence embeddings at the same time. We couple this with an attention mechanism over the latent word embeddings to realize the end-to-end autoencoder framework.

I. Introduction

Recurrent Neural Network language models (RNNLM) have shown superior results in unsupervised generative modelling [1]. Supervised techniques like Machine Translation [2] and Image Captioning [3] have also achieved state of the art using these models. However, because RNNLM is an autoregressive probabilistic model that makes a series of step by step predictions, it does not model the global features such as topics, outside of its rich conditional distributions. This also means that the RNNLM will overfit to N-grams which appear frequently in the training data and thus stifles a diverse exploration of language.

It has been shown that variational autoencoder based language models are able to capture high level features, generate diverse sentences and also smoothly interpolate between sentences [4]. Unlike traditional autoencoders [8], the Variational Autoencoder [6] drives the representation to a dense region of space instead of a single deterministic vector thus building a generative model which has been observed to compress the data into a semantically meaningful space [4], [7]. In the VAE framework reconstruction is phrased as sampling from an approximate posterior

distribution, with a strong regularization term that I-projects the posterior distributions into the prior distribution (shown below). Although variational Seq2Seq models are more challenging to train, it is shown in [4] that it is possible to train a VAE to generate sentences from a continuous latent space.

$$L_{ELBO} = E_{z\sim q(z|x)}[-log(p(x|z))] + \int q(z|x)log(\frac{q(z|x)}{p(z)})$$

Where $q(z|x)$ is the posterior distribution or encoder, $p(x|z)$ is the likelihood or decoder, and p(z) is a $N(0, I)$ Gaussian in the non-hierarchical case.

Most of the traditional approaches employ a simple prior over the latent space like a normal distribution. This is convenient for training but converting the data distribution to a single mode can often lead to overly simplified representations which cannot faithfully reproduce the true distributions of the data. Hierarchical VAEs are proposed in [11], [12] to allow for greater modelling flexibility and structure.

The state of the art language models connect the encoder and the decoder through an attention mechanism. Attention mechanisms have become an indispensable part in sequence modelling because of their power to model dependencies without regard for the position in the input or the output sequence [2], [5]. An Attention Mechanism helps us retain long term information which RNNs have trouble with, and can significantly improve performance of the Seq2Seq model [2], [9].

The above mentioned word level language models cannot handle languages with a large vocabulary and rich morphology like German, Russian and Turkish. They also cannot handle out of vocabulary words as they are built on a fixed predefined vocabulary. In addition to that, when there is a large number of words in the dictionary, the parameters for the softmax itself dwarf the rest of those in the model for any given layer, as well as being a speed bottleneck in the network. Towards this end character level language models have been explored. For example, a character level convolutional neural network coupled with a highway network and LSTM is used in [15]. It is shown that on languages with rich morphology (German, Spanish, Russian) the model outperforms the word level LSTM baselines at the same time using fewer parameters. Google's Neural Machine Translation System [14] addresses the issue of out of vocabulary words by dividing the words into a limited set of common sub word units called "wordpieces" which handle the translation of rare words and improves the overall accuracy of the system. They

also go a step further, training hybrid models with characters for rare words. However it is not obvious that these characters actually allow them to translate out of vocabulary words, but is more for copying words like names between the input and output sentence.

In this project we develop a novel variational, character level language model to overcome the limitations of the word level models. We propose two hierarchical frameworks which differ in the way they incorporate the hierarchy, while both can jointly generate word and sentence level embeddings thus allowing us to form latent representations with sentence level context. We use a variational autoencoder to produce these latent representations and employ an attention mechanism over the latent word embeddings to account for long term dependencies in the sentence. This approach is not only novel from a language modelling perspective, but also from the perspective of VAEs as such a deep latent hierarchical structure has never been explored. If successful, this approach may serve to augment neural machine translation exchanging the encoder and decoder for ours, or for writing where one may need a different sentence with the same context.

II. Framework

A. Framework 1: Hierarchy in the Sentence Based Latent representation

This model builds independent word-based latent representations, saving the rich conditional hierarchy for encodings of the latent word representations into global sentence representations, which encode positional dependence (Figure 1, without hierarchy). This allows us to make universal word representations for any arbitrary sequence of characters independent of context (at the extreme when the kl goes to 0 they are totally independent although this will likely not happen as it would harm reconstruction). This can be mostly taken care of by using a hierarchical prior (Figure 3). In the simple case, we encode our characters into an RNN piece-by-piece until we hit an EOW (end of word) token, at which point we output the mean and variance of a posterior latent variable, and sample using the re-parametrization trick. This is repeated across the encoded sentence (Figure 1). We then average these random variables, which after a dense layer outputs a mean and variance as the hierarchical global latent representations. After this we can decode directly, or build a hierarchy with a deep attention mechanism (Figure 1). In the case that we build a hierarchy, our global latent variable is used as a query vector against the word-based latent variables, the output of which is pushed through a linear layer, or a dense then linear layer to parametrize the output distribution. This can be repeated n times where n is the depth

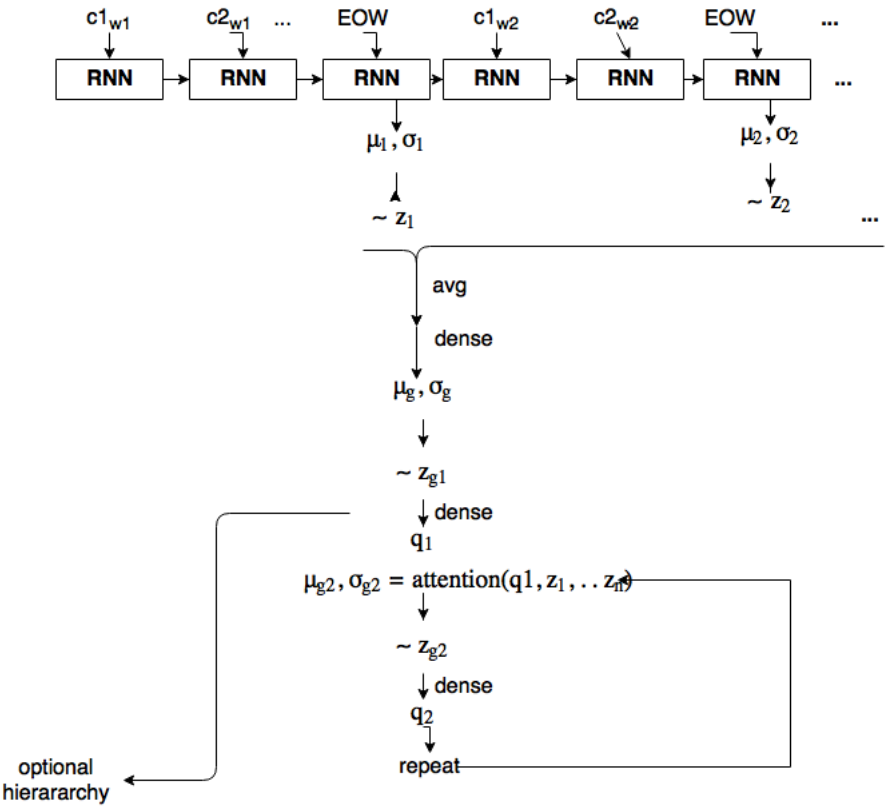

Fig. 1: Uncorrelated word embeddings, with hierarchy in the sentence latent representation only. Here 'Zg' is a hierarchical random variable which includes correlation with the latent word embeddings.

of the hierarchy. In the decoder, we proceed from our global latent representation, and build a query by processing it with an RNN, which we use to reference the word embeddings with an attention mechanism (Figure 2).This allows us to map from our word-based latent representations to a character based predictions. The output of this is then fed through the final RNN which outputs the probability of a character, given the latent representation. Here we decode greedily which is reasonable as every output character should be conditionally independent of every other output character given the latent representation. This decoding is repeated until an EOS symbol is hit (Figure 2). The prior distribution can mimic the encoder, starting by sampling a standard Gaussian for each of the desired words and the global latent variables. Indeed the generative model here will likely not work very well, as is usual for unigram models. However the posterior still has useful application. To make a more rich generative model, we can make our prior fully dependent (Figure 3).

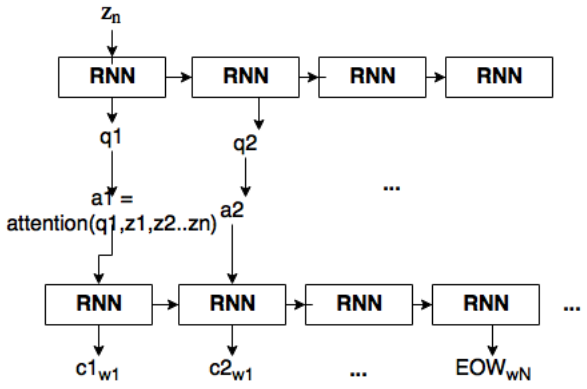

Fig. 2: Decoder.

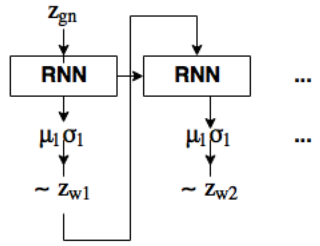

Fig. 3: Correlated prior distribution.

B. Framework 2: Hierarchy in the RNN

Here our model is quite similar to the one mentioned above, with the key difference being that our encoding RNN builds latent representations conditioned of the previous latent variable sampled. To do this we propose to run the RNN across each character, after the first EOW we output a mean and variance as before. However this time we reset the hidden state of the RNN, and only pass in the latent variable as a representations of the previous parts of the sentence. In this case our last hidden variable is highly correlated with the previous random variables, serving not only as a word embedding, but a word embedding which is highly context specific. Due to the immense depth of this stochastic network as is, our initial experiments will not include adding a deeper hierarchy like in the previous section, however it is optional and so it is depicted in Figure 4. The decoding in this case is done exactly like the previous architecture (Figure 2), and the prior used in this case is the same as that which was already mentioned to solve the problem of independent word representations (Figure 3). In practice we find that resetting the cell state of the LSTM with our latent variable sample to be corrupting of the LSTMs hidden

state. Not doing this makes our latent variables independent of each other. It is also possible that the increase of long term dependencies when passing in the sample hinders gradient descent.

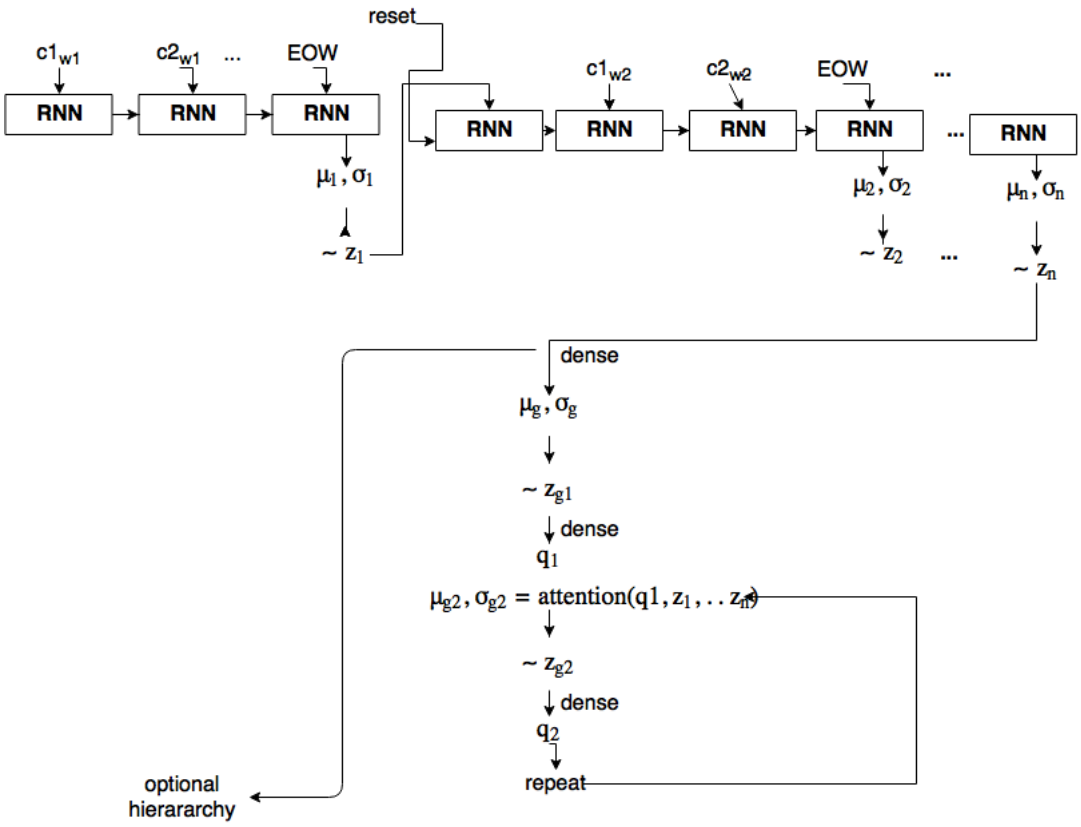

Fig. 4: Hierarchical approximate posterior, or encoder with a correlated hierarchy in the RNN. Optionally more correlation may be employed to make a deep hierarchical sentence representation 'Zg'

III. Evaluation

For evaluation we use an upper bound on the commonly used bits-per-character (BPC) metric for character level language models as well as a lower bound on the log-likelihood. To compare our models we also use reconstruction accuracy.

$$BPC \leq -\frac{1}{T} log_2(p_{ELBO}(Char_{true}))$$

where T is the number of characters. We average this over the test set. Our model is at an unfair disadvantage relative to the models we compare against as the negative ELBO gives us an upper bound on the log likelihood, and therefore and upper bound on the BPC. In future work we plan on getting a more accurate estimate of the BPC with importance weighting [16].

We test our framework on the English Penn TreeBank (PTB) dataset [17], utilizing the standard training (0-20), validation (21-22), and test (23-24) splits. Our vocabulary has 61 tokens including special characters. We preprocess the data to get rid of empty sentences and replace the $< \mathbf{unk} >$ token with a special character that the network almost never sees.

IV. Results

During hyper-parameter tuning it was noticed that sampling the generative model often resulted in unidentifiable sequence of characters that were far from words. This may be because the RNN in the prior can easily find itself exploring a space in its conditional distributions that was not well trained. To correct for this we use a technique akin to curriculum learning in the prior, where instead of feeding in the sample from the posterior at the previous time step to the parametrize the prior distribution, we feed in a sample from the priors own distribution at the previous time step. This acclimatizes the prior to it's actual output distributions. Furthermore to help increase the reconstruction accuracy we experimented with teacher forcing. Finally we tried a model in which we use a sample KL instead of a partially analytical KL. The intuition here is that the higher variance in the gradient may help regularize our model. From our results we can conclude that some of the proposed fixes to the aforementioned problems seem to be working to the extent that we are observing better results in a shorter period of time. Unfortunately all these models require a lot of time to train and although they may have the potential to do better, limited resources would not allow for more exploration.

In Table I we show the performance of some of the variants of our proposed model with upper bound on the BPC, lower bound on the NLL and the character level accuracy as the metrics. In addition we show the magnitude of the KL term for each model, this gives us an idea of how informative the latent variables are, if the KL=0, then our posterior latent variables are not differentiable from that of the prior, and are therefore uninformative. We make a note of the time taken for training for our different variants. Also we compare our performance with two state-of-the-art character level language models, the one that uses Zoneout regularization in the RNN [20] and the approach that uses Hypernetworks [21]. We can see that the teacher forcing model achieves high reconstruction accuracy, but this is because it has lost pressure to put information into the latent variables, as can be seen from the low KL and the sample from the prior (appendix). Furthermore we see that the longer training time makes our base model much stronger, in terms of both high kl and good reconstruction. While curriculum learning

seems to be making the posterior more uninformative. Our best BPC is a few bits off state of the art as can be seen in Table I, this may be due to a variety of reasons, but most likely due to the generative model. As our model is observing a reasonable reconstruction accuracy, there is a need for building a more robust generative model. It was observed that when sampling the posterior distributions of the word representations, there was no diversity in the decodings. This is happening because the standard deviations for the posterior distributions are very small. We hypothesize that our network is doing this because it facilitates a more faithful encoding, as otherwise there is higher variance in samples with larger standard deviations. We propose to attenuate this with an **entropy penalty** on our prior distribution, forcing our posterior to spread its density out more, making for a denser latent space, being easier to decode from and thus giving us a lower BPC. Finally, we can see that our generative samples from our model with a longer training period (appendix) makes some identifiable words, however they certainly do not form a semantically meaning-full sentence representation. This is likely due to the fact that our latent variables meant to capture global context have been set equal to the prior (we have observed the KL to be very near 0 in most cases). Although this is unfavorable, our main goal is to build word based latent representations, not sentence based latent representations, which would still be useful in applications such as machine translation.

TABLE I: Performance of Variants. For each model we use a learning rate of 1e-3, 512 hidden units for each LSTM layer and fully connected layers. In the curriculum learning model P is the probability of keeping a posterior sample

Variant	U-BPC	L-LL	C-Acc	KL	time
regular model	2.9	225.4	60	79.5	21hr
regular model + longer training time	-	239.4	81.2	93.4	1d13hr
teacher forcing	-	239.3	91	76.3	1d13hr
curriculum learning in prior (P= 0.8)	-	229.5	57	45.01	20hr
sample based KL	3.2	264.5	66	94	21hr
independent prior	-	-	72.12	2000	20hr
Hypernetworks	1.281	-	-	-	-
Zoneout	1.362	-	-	-	-

V. Conclusion

In conclusion we provide results for a novel neural network architecture on the difficult problem of character level language modelling. From our results, it is clear that our model trains slowly, making for a bottleneck in model development and hyperparameter tuning. Indeed this is a general problem with deep RNN-based models and has motivated recent interest in **fully attentional architectures** [22]. We plan to migrate to this architecture to make our model faster and more robust.

VI. Appendix

A. Qualitative Results:

1) Prior samples from the basic model with a long training time: $< SOS > american < EOW > noes < EOW > manaeers < EOW > from < EOW > an < EOW > woetlr < EOW > crash < EOW >$

2) Prior samples from the teacher-forcing model: $< SOS > tyummmmmmxhh < EOW > , v < EOW > v < EOW > mtdmvmoowyyur < EOW >, < EOW > rttytttsesesssss$

3) Prior samples from the curriculum-learning model: $< SOS > dffdreorsts < EOW > eeette < EOW >< EOW >$

4) Prior samples from the sample-KL model: $< SOS > foiley < EOW > as < EOW > coller < EOW > invert < EOW >$

B. Informative TensorFlow Summaries

1) Bits per Character estimate: It's observed from Figure 5 that the BPC estimate is high until the annealing starts after which it reduces significantly. This is expected as the model is not regularized until the KL term is turned on and we would expect to see a better and compact representation of information after regularization .

2) Histogram of kl terms for the latent words: It can be seen from Figure 6 that the kl terms are clustered near zero before annealing. After annealing begins, the terms shoot up and slowly decrease over time also narrowing down in the process (decrease in standard deviation).

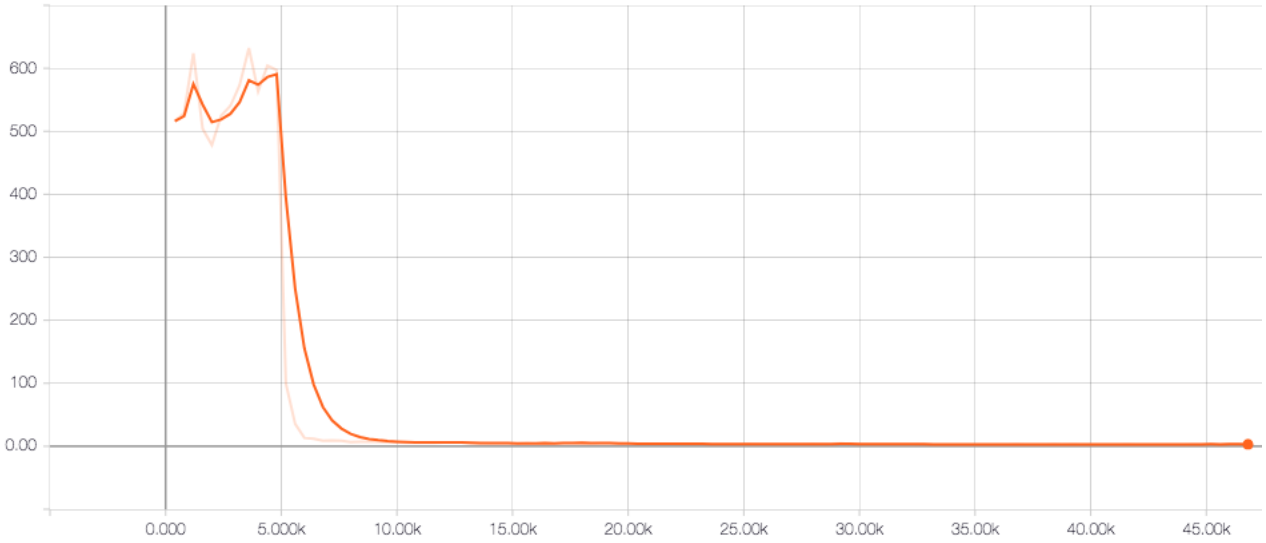

Fig. 5: The Bits per Character estimate over time. It's observed that the BPC estimate is high until the annealing starts after which it reduces significantly. This is because of the presence of a term which calculates the distance between posterior and prior distributions, which reduces as time goes on.

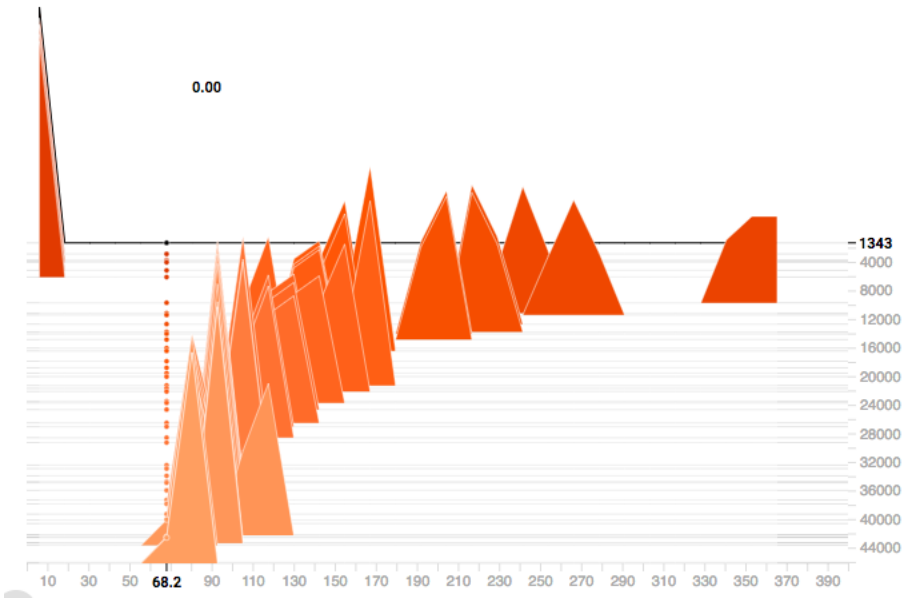

Fig. 6: The histogram of kl terms for the latent words. The z-axis coming into the page is the direction of increasing time steps. The x-axis is the bins and the y-axis is the number of latent variables that fall into that particular bin. The reduction in the standard deviation tells us that the model is more sure about the distribution as time progresses. This surety might not be very good for generation as the model will fail to generate diverse samples.

REFERENCES

[1] Tom as Mikolov, Stefan Kombrink, Luk as Burget, Jan Honza Cernocky‘, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In Proc. ICASSP.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proc. ICLR.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In Proc. CVPR.

[4] Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. CoNLL.

[5] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In International Conference on Learning Representations, 2017.

[6] Diederik P Kingma and Max Welling. 2013. Auto- encoding variational Bayes. arXiv preprint arXiv:1312.6114.

[7] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward con- trolled generation of text. In Proceedings of the 34th International Conference on Machine Learning. vol- ume 70, pages 15871596.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural net- works. Science 313(5786):504507.

[9] Alexander M. Rush, Sumit Chopra, and Jason We- ston. 2015. A neural attention model for ab- stractive sentence summarization. In Proceed- ings of the 2015 Conference on Empirical Methods in Natural Language Processing. pages 379389. https://doi.org/10.18653/v1/D15-1044.

[10] Bahuleyan, Hareesh, et al. ”Variational Attention for Sequence-to-Sequence Models.” arXiv preprint arXiv: 1712.08207 (2017).

[11] Snderby, Casper Kaae, et al. ”Ladder variational autoencoders.” Advances in neural information processing systems. 2016.

[12] Goyal, Prasoon, et al. ”Nonparametric variational auto-encoders for hierarchical representation learning.” arXiv preprint arXiv:1703.07027 (2017).

[13] Hwang, Kyuyeon, and Wonyong Sung. ”Character-level language modeling with hierarchical recurrent neural networks.” Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017.

[14] Wu, Yonghui, et al. ”Google’s neural machine translation system: Bridging the gap between human and machine translation.” arXiv preprint arXiv:1609.08144 (2016).

[15] Kim, Yoon, et al. ”Character-Aware Neural Language Models.” AAAI. 2016.

[16] Burda, Grosse, et al. ”Importance Weighted Autoencoders.” arxiv. 2016.

[17] Marcus, M.; Santorini, B.; and Marcinkiewicz, M. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. Compu- tational Linguistics 19:331330.

[18] Available online: http://www.statmt.org/wmt13/translation-task.html

[19] Available online: http://opus.nlpl.eu/News-Commentary.php

[20] Krueger, D., Maharaj, T., Kramr, J., Pezeshki, M., Ballas, N., Ke, N. R., ... Pal, C. (2016). Zoneout: Regularizing rnns by randomly preserving hidden activations. arXiv preprint arXiv:1606.01305.

[21] Ha, D., Dai, A., Le, Q. V. (2016). Hypernetworks. arXiv preprint arXiv:1609.09106.

[22] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in Neural Information Processing Systems. 2017.