TOPIC-AWARE ABSTRACTIVE TEXT SUMMARIZATION

STAT 8056 COURSE PROJECT REPORT

Yu Yang*
School of Statistics
University of Minnesota
yang6367@umn.edu

ABSTRACT

Neural sequence-to-sequence models have made great progress in abstractive text summarization. In this work, we focus on topic-aware abtractive text summarization, which incorporates the global semantic information revealed by topic models, for example, Replicated Softmax. Such incorporation allows the decoder to access the corpus-level words co-occurrence and provides valuable inductive bias for language generation. We experiment the idea of combining the pointer-generator model and the Replicated Softmax topic model via a simple linear transformation of the latent topic vector and show that such a simple mechanism is insufficient to produce better summaries.

***K*eywords** Abstractive summarization · Topic modeling · Pointer-Generator · Replicated Softmax

1 Introduction

Automatic summarization focuses on capturing the most salient information in the source text and producing a condensed shorter version of the text. Extractive summarization (Cheng and Lapata, 2016; Nallapati, Zhai, et al., 2017) identifies and concatenates parts of the input document, while abstractive summarization (See et al., 2017; Paulus et al., 2018; L. Wang et al., 2018) focuses on generating a summary by rephrasing or introducing novel words and is in general more challenging. In this paper, we focus on abstractive summarization, for which a lot of progress has been made with neural-based sequence-to-sequence models (Nallapati, Zhou, et al., 2016; See et al., 2017).

In this paper, we consider the problem of incorporating topic representations in the summarization model so that we can get richer output summaries. Topic representations capture the global semantic information of the document and the words co-occurrence statistics at the corpus-level(Ailem et al., 2019), while sequence-to-sequence models usually focus on local and sequential information. It provides a more global context for language generation and hence we expect it to be likely to boost the performance of text summarization models. Many recent topic-aware text summarization papers utilize LDA (Blei et al., 2003) for topic modeling, however, LDA assumes that the topics are drawn from multinomial distributions across an article, which won't hold when the articles are from diverse domains. Therefore, we consider a different topic model – Replicated Softmax (Hinton and Salakhutdinov, 2009), whose hidden layer can be used as the latent topic representation of the document.

We propose a topic-aware model where the pointer-generator based decoder (See et al., 2017) is provided with latent topic representations of the input documents by a simple linear transformation mechanism. We apply the proposed model to the CNN/Daily Mail benchmark dataset and compare it with the original pointer-generator model. We show that such a simple linear transformation is insufficient to achieve a better performance and that more investigations are needed in the future.

The rest of the paper goes as follows. Section 2 briefly summarizes the related work; Section 3 describes the architecture of the proposed model in detail; Section 4 gives the implementation settings and shows the experiment results; Section 5 concludes the paper with some discussion.

*Check `https://github.umn.edu/YANG6367/PGNet` for code and detailed results.

2 Related Work

Our model follows the attention-based sequence-to-sequence models (Nallapati, Zhou, et al., 2016) and is built on top of the pointer-generator network (See et al., 2017), which introduces a copy mechanism to address the out-of-vocabulary problem and is used as the baseline for model comparison in many papers due to its good performance (Ailem et al., 2019; Narayan et al., 2018; Z. Wang et al., 2020; Fu et al., 2020).

In terms of topic modeling, it has been extensively studied for document modeling and information retrieval, including the probabilistic topic models like pLSA (Hofmann, 1999) and LDA (Blei et al., 2003), graph-based models such as Replicated Softmax (Hinton and Salakhutdinov, 2009) and Deep Boltzmann Machine (DBM) (Srivastava et al., 2013), and neural-based topic models (Miao et al., 2017). Among these models, LDA has been applied by a couple of works (L. Wang et al., 2018; Narayan et al., 2018) to text summarization. However, few papers considered Replicated Softmax or DBM, probably due to its difficulty and inconvenience in implementation. In this paper, we try to fill the gap and choose Replicated Softmax as the topic model.

Recently, several approaches on leveraging topic information have been proposed. Most of them used external topic models such as LDA. (Narayan et al., 2018) concatenated the word embeddings with the point-wise multiplication of the topic distribution of the document and the topic distributions of the words to obtain topic sensitive embeddings of the input. And their decoder conditioned each word generation on the document topic vector. (L. Wang et al., 2018) passed the word embeddings and the topic embeddings to two convolutional blocks separately and incorporated the topic information by a joint attention and biased probability generation mechanism. (Ailem et al., 2019) leveraged topic information by transforming the inner product of the topic-word parameters and the topic vector, both given by LDA, into a new mixture component in the generative probability for decoding. (Fu et al., 2020), however, didn't use external topic models, and instead, they accomplished topic inference and summarization in an end-to-end manner via variational encoder decoder and induced both the paragraph-level and document-level latent topics to expose the critical words and paragraphs for summarization.

3 Proposed Model

The proposed model consists of two components: the Pointer-Generator model (See et al., 2017) and the Replicated Softmax model (Hinton and Salakhutdinov, 2009), as described in Figure 1. The Replicated Softmax model is firstly trained to produce latent topic representations, and then the pointer-generator model is trained using the source text and the latent topic representation as its input.

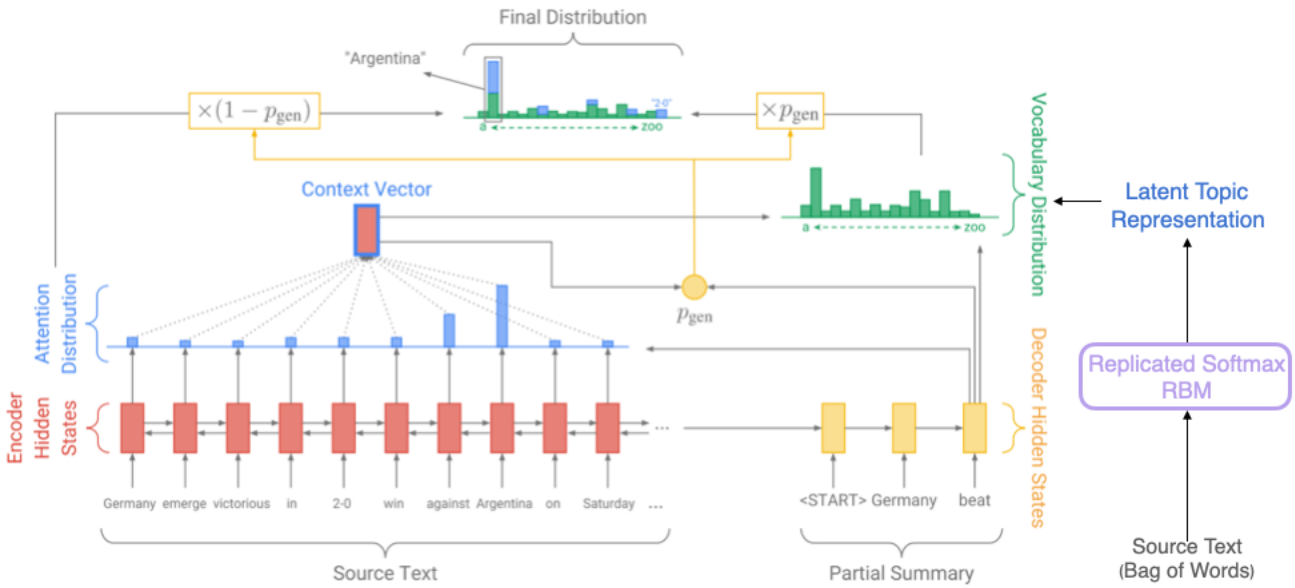

Figure 1: Model Architecture

3.1 Replicated Softmax

Replicated Softmax (Hinton and Salakhutdinov, 2009) is a family of Restricted Boltzmann Machines (RBMs) and it can be used to learn the latent topic representations of the documents. It has a two-layer architecture as shown in Figure 3 in the Appendix. Let K denote the dictionary size, D denote the document size, and F denote the number of hidden units. Let $\mathbf{v} \in \{1, \ldots, K\}^D$ be the discrete visible units, and let $\mathbf{h} \in \{0, 1\}^F$ be binary stochastic hidden topic features. Let $\mathbf{V}$ be a $K \times D$ binary matrix with $v_i^k = 1$ if the ith visible unit takes on the kth value in the dictionary. Let $\mathcal{H}_j$ denote the value space of h_j, which in our case is $\{0, 1\}$.

Energy-based Model The energy function of the state $\{\mathbf{V}, \mathbf{h}\}$ is defined in Equation (1) and the corresponding free energy is shown in Equation (2), which can be computed tractably.

$$E(\mathbf{V}, \mathbf{h}) = -\sum_{j=1}^{F}\sum_{k=1}^{K} W_j^k h_j \hat{v}^k - \sum_{k=1}^{K} \hat{v}^k b^k - D\sum_{j=1}^{F} h_j a_j, \tag{1}$$

$$\text{FreeEnergy}(\mathbf{V}) = -\sum_{k=1}^{K} \hat{v}^k b^k - \sum_{j=1}^{F} \log \sum_{hj \in \mathcal{H}_j} \exp(D h_j a_j + \sum_{k=1}^{K} W_j^k h_j \hat{v}^k), \tag{2}$$

where $\hat{v}^k = \sum_{i=1}^{D} v_i^k$ denotes the count for the kth word in the dictionary.

One good property of energy-based models is that the log-likelihood gradient has an interesting form (Bengio, 2009) and we can utilize such a good property to learn the model parameters. In the case where the free energy can be computed tractably, the first term in (3) can be easily computed by plugging in the observed data, and the second term can be approximated by a stochastic estimator, as long as we can draw samples from the model distribution P.

$$E_{\hat{P}}\left[\frac{\partial \log P(V)}{\partial \theta}\right] = -E_{\hat{P}}\left[\frac{\partial \,\text{FreeEnergy}(V)}{\partial \theta}\right] + E_P\left[\frac{\partial \,\text{FreeEnergy}\,(V)}{\partial \theta}\right], \tag{3}$$

where $\hat{P}$ is the empirical distribution of the training data and P is the model distribution.

Gibbs Sampling Due to the bipartite formulation of RBM, the hidden units are conditionally independent of each other given the visible units, and the same for the visible units. This property allows us to perform Gibbs sampling efficiently to draw samples from the model distribution P. The conditional distributions are derived as in Equation (4, 5). And note that when sampling the visible layer given the hidden layer, we don't sample $\hat{v}^k$, instead, we sample v_i from multinomial distribution repeatedly for $i = 1, 2, \cdots, D$ to get a binary matrix $\mathbf{V}$ and then use $\mathbf{V}$ to calculate the word counts $\hat{v}^k$ for $k = 1, 2, \cdots, K$.

$$P\left(v_i^k = 1 \mid \mathbf{h}\right) = \frac{\exp\left(b^k + \sum_{j=1}^{F} h_j W_j^k\right)}{\sum_{q=1}^{K} \exp\left(b^q + \sum_{j=1}^{F} h_j W_j^q\right)}, \tag{4}$$

$$P\left(h_j = 1 \mid \mathbf{V}\right) = \sigma\left(D a_j + \sum_{k=1}^{K} W_j^k \hat{v}^k\right), \tag{5}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function.

Contrastive Divergence We use Contrastive Divergence (Hinton, 1999; Hinton, 2002) to learn the parameters. The idea of k-step Contrastive Divergence (CD-k) is to run the MCMC chain for only k steps starting from the observed example. The CD-k update after seeing example $\mathbf{V}$ is

$$\Delta\theta = -\frac{\partial \,\text{FreeEnergy}\,(\mathbf{V})}{\partial \theta} + \frac{\partial \,\text{FreeEnergy}\,(\tilde{\mathbf{V}})}{\partial \theta}, \tag{6}$$

where $\tilde{\mathbf{V}}$ is a sample from the Gibbs sampling chain after k steps.

3.2 Pointer-Generator Model

Throughout this section, let H denote the dimension of the encoder hidden states, and D denote the length of the source text, h_i denote the encoder hidden state, x_t denote the decoder input, s_t denote the decoder hidden state, w_t^* denote the target word, and l denote the latent topic representation of the document.

Encoder The encoder takes the embedding vectors of the tokens as input, obtained by a simple look up table that is learnable, and uses a one-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to learn the temporal dependencies of the input sequences. The output is the concatenation of the forward and backward hidden states. Denote the encoded sequences as $h_1, h_2, \cdots, h_D \in \mathbb{R}^{2H}$.

Attention We use the same attention mechanism as that in (Bahdanau et al., 2014) to obtain the attention distribution and the context vector h_t^*.

$$e_i^t = v^T \tanh\left(W_h h_i + W_s s_t + b_{\text{attn}}\right),$$
$$a^t = \text{softmax}\left(e^t\right),$$
$$h_t^* = \sum_i a_i^t h_i,$$

where v, W_h, W_s and b_{attn} are learnable parameters.

Decoder (incorporating topic information) The context vector and the decode hidden state s_t are concatenated and then fed through two fully-connected layers and a softmax layer to produce the vocabulary distribution P_{vocab}. Additionally, to incorporate the topic information of the document, a linear transformation of the latent representation vector l is added before applying the softmax layer.

$$P_{\text{vocab}} = \text{softmax}\left(V'\left(V\left[s_t, h_t^*\right] + b\right) + b' + Wl\right)$$

To help solve the out-of-vocabulary (OOV) problem, (See et al., 2017) introduced the copy mechanism. In more detail, at time step t, the generation probability $p_{\text{gen}} \in [0, 1]$ is calculated from the concatenation of the context vector h_t^*, the decoder state s_t, and the decoder input x_t:

$$p_{\text{gen}} = \sigma\left(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{\text{ptr}}\right)$$

where vectors w_{h^*}, w_s, w_x and scalar b_{ptr} are learnable parameters and σ is the sigmoid function.

Then, p_{gen} works as a soft gate controlling whether the next word is generated from the vocabulary distribution or directly copied from the source text. For a word w, the final probability distribution over the extended vocabulary is:

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t.$$

Loss During training, the loss for time step t is the negative log-likelihood of the target word W_t^* and the overall loss for the whole sequence is:

$$\text{loss} = \frac{1}{D} \sum_{t=0}^{D} \log P\left(w_t^*\right).$$

3.3 Inference

At test time, the sample document will first go through the Replicated Softmax model to generate the topic latent representation l, then the latent vector and the embedded document vector will be passed on to the modified pointer-generator model, producing a probability distribution over the vocabulary at each time step. And then beam search is used to produce the summaries and we set beam size to be 4.

4 Experiments

4.1 Setup

Datasets We use the CNN/Daily Mail dataset (Nallapati, Zhou, et al., 2016). It consists of 312,085 online news articles, where each article (781 tokens on average) is paired with a multi-sentence summary (3.75 sentences on average). As in (See et al., 2017), we use the non-anonymized version of the data[2], which is split into 287,226, 13,368 , and 11,490 pairs for training, validation and testing respectively. An example [3] of a $\langle text, summary \rangle$ pair is shown in Appendix 6.2.

[2]Follow the instructions at `https://github.com/abisee/cnn-dailymail`.

[3]Check more examples at CNN/Daily Mail dataset viewer: `https://huggingface.co/datasets/viewer/?dataset=cnn_dailymail&config=3.0.0`.

Models in comparison The baseline model is a pointer-generator model with a copy mechanism(See et al., 2017), referred as **PGNet** for short. It has three variants: the model given by the original paper, the one trained by us with 500k steps, and the one trained with 100k steps. Our approach is referred as **PGNet + Text RBM** and also has two variants: trained with 500k steps and with 100k steps.

Evaluation metrics We use the standard ROUGE scores(Lin, 2004) to evaluate the models and report the F_1 scores for ROUGE-1, ROUGE-2 and ROUGE-L, which respectively refers to the overlap of unigram, bigrams, and the longest common sequence between the generated summary and the reference summary.

Model specification We follow the instructions in (See et al., 2017) for the **PGNet** model architecture. And for the Replicated Softmax model, we use $F = 200$ as the hidden latent dimension and set $K = 15,000$ as the dictionary size.

Training We apply the same training strategies as in (See et al., 2017) for the **PGNet** model: preprocess the dataset by the Stanford CoreNLP tool, set the batch size as 8, and use the Autograd optimizer with learning rate 0.15 and an initial accumulator value of 0.1. As for the Replicated Softmax model, we first remove the common stopwords, do lemmatization, and then keep the 15,000 most frequent words in the training set. In Contrastive Divergence training, the number of sampling iterations k is set as 1, the batch size is set as 128, and the number of training steps is 2000 (after iterating all instances).

4.2 Results

The main results are reported in Table 1. We can clearly see that the **PGNet + Text RBM** doesn't outperform the original **PGNet** model. This implies that a simple linear transformation of the latent representation vector is insufficient for a performance boost and that more complex model structure should be considered to fully capture the global semantic information. Another surprising finding is that the performance of **PGNet** after 100k training steps outperforms the one after 500k steps. We suppose this might be due to overfitting, as the training loss curve becomes flat after 100k steps (see Figure 2 in the Appendix). Additionally, we can see from Figure 2 that at a large scale, the two training curves are very close, and this implies that the insertion of the topic vector doesn't make a big difference and this might be due to the small amount of additional parameters and the insertion manner.

At the same time, we need to note that the ROUGE scores is not directly related to the log-likelihood loss function, so it is possible that ROUGE cannot fully reflect the effect of our modification, and therefore, more relevant evaluation metrics should be considered in the future.

Method	ROUGE		
	1	2	L
PGNet (original paper results)	36.44	15.66	33.42
PGNet (my 100k run)	**36.98** (36.74, 37.21)	**15.93** (15.72, 16.15)	**33.55** (33.32, 33.77)
PGNet (my 500k run)	35.92 (35.70, 36.16)	15.51 (15.31, 15.73)	32.87 (32.65, 33.11)
PGNet + Text RBM (my 100k run)	36.33 (36.10, 36.56)	15.62 (15.41, 15.84)	33.05 (32.82, 33.28)
PGNet + Text RBM (my 500k run)	36.33 (36.08, 36.57)	15.83 (15.60, 16.07)	33.25 (33.00, 33.48)

Table 1: ROUGE F_1 scores on the test set
(95% confidence intervals are displayed in the parentheses.)

5 Conclusions and Future Work

In this paper, we propose to combine the PGNet model and the Replicated Softmax model and incorporate the topic information of the document by adding a simple linear transformation to the decoding distribution. The results show that such a proposal is not as effective as desired and that more complex models should be considered. Current work is a good start but is far from sufficient. There are a few potential directions for future work: (1) examine the quality of the topic representation given by the Replicated Softmax after training; (2) do qualitative analysis on the generated summaries to see where the model goes wrong; (3) try inserting the topic representation by manipulating the attention mechanism; (4) use a different evaluation metric that is more related to the loss function.

References

Ailem, Melissa, Bowen Zhang, and Fei Sha (2019). "Topic augmented generator for abstractive summarization". In: *arXiv preprint arXiv:1908.07026*.

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

Bengio, Yoshua (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *the Journal of machine Learning research* 3, pp. 993–1022.

Cheng, Jianpeng and Mirella Lapata (2016). "Neural Summarization by Extracting Sentences and Words". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 484–494.

Fu, Xiyan et al. (2020). "Document summarization with vhtm: Variational hierarchical topic-aware mechanism". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 7740–7747.

Hinton, Geoffrey E (1999). "Products of experts". In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99.(Conf. Publ. No. 470)*. Vol. 1. IET, pp. 1–6.

– (2002). "Training products of experts by minimizing contrastive divergence". In: *Neural computation* 14.8, pp. 1771–1800.

Hinton, Geoffrey E and Russ R Salakhutdinov (2009). "Replicated softmax: an undirected topic model". In: *Advances in neural information processing systems* 22, pp. 1607–1614.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hofmann, Thomas (1999). "Probabilistic latent semantic indexing". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57.

Lin, Chin-Yew (2004). "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*, pp. 74–81.

Miao, Yishu, Edward Grefenstette, and Phil Blunsom (2017). "Discovering discrete latent topics with neural variational inference". In: *International Conference on Machine Learning*. PMLR, pp. 2410–2419.

Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou (2017). "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1.

Nallapati, Ramesh, Bowen Zhou, et al. (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290.

Narayan, Shashi, Shay B Cohen, and Mirella Lapata (2018). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807.

Paulus, Romain, Caiming Xiong, and Richard Socher (2018). "A Deep Reinforced Model for Abstractive Summarization". In: *International Conference on Learning Representations*.

See, Abigail, Peter J Liu, and Christopher D Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083.

Srivastava, Nitish, Ruslan Salakhutdinov, and Geoffrey Hinton (2013). "Modeling documents with a Deep Boltzmann Machine". In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 616–624.

Wang, Li et al. (2018). "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization". In: *arXiv preprint arXiv:1805.03616*.

Wang, Wenlin et al. (2019). "Topic-guided variational autoencoders for text generation". In: *arXiv preprint arXiv:1903.07137*.

Wang, Zhengjue et al. (2020). "Friendly topic assistant for transformer based abstractive summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 485–497.

6 Appendix

6.1 Figures

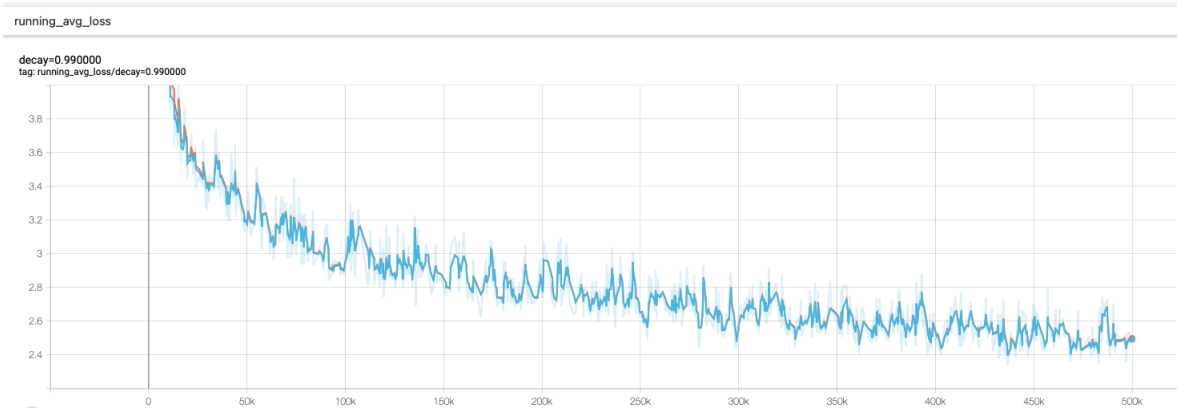

Figure 2: Training Loss Curve: x-axis represents the training steps and y-axis represents the average loss.The blue curve represents **PGNet + Text RBM** model and the orange curve represents the **PGNet** model.

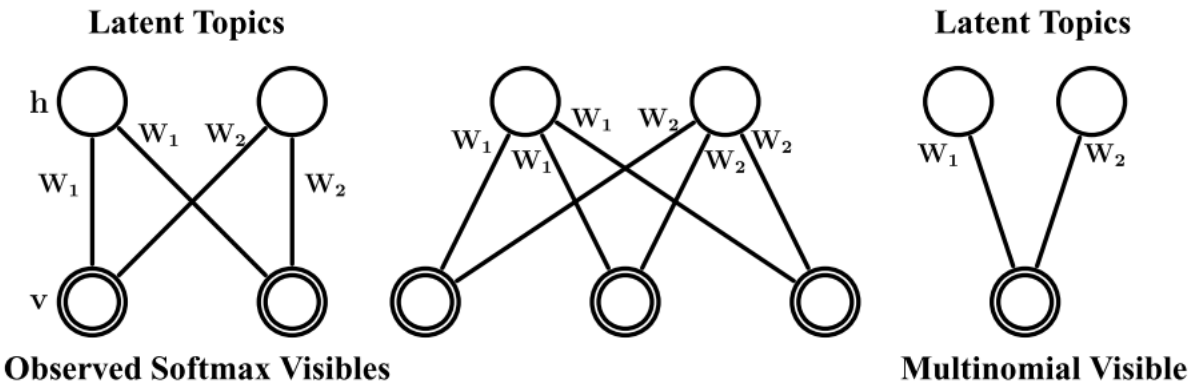

Figure 3: Replicated Softmax Model. The top layer represents a vector h of stochastic, binary topic features and and the bottom layer represents softmax visible units v. All visible units share the same set of weights, connecting them to binary hidden units. Left: The model for a document containing two and three words. Right: A different interpretation of the Replicated Softmax model, in which D softmax units with identical weights are replaced by a single multinomial unit which is sampled D times.

6.2 Data Example

Original Text: (CNN) – An American woman died aboard a cruise ship that docked at Rio de Janeiro on Tuesday, the same ship on which 86 passengers previously fell ill, according to the state-run Brazilian news agency, Agencia Brasil. The American tourist died aboard the MS Veendam, owned by cruise operator Holland America. Federal Police told Agencia Brasil that forensic doctors were investigating her death. The ship's doctors told police that the woman was elderly and suffered from diabetes and hypertension, according the agency. The other passengers came down with diarrhea prior to her death during an earlier part of the trip, the ship's doctors said. The Veendam left New York 36 days ago for a South America tour.
Ground Truth Summary: The elderly woman suffered from diabetes and hypertension, ship's doctors say .Previously, 86 passengers had fallen ill on the ship, Agencia Brasil says .