Neural Variational Inference for Text Processing

Yishu Miao[1] YISHU.MIAO@CS.OX.AC.UK
Lei Yu[1] LEI.YU@CS.OX.AC.UK
Phil Blunsom[12] PHIL.BLUNSOM@CS.OX.AC.UK
[1]University of Oxford, [2]Google Deepmind

Abstract

Recent advances in neural variational inference have spawned a renaissance in deep latent variable models. In this paper we introduce a generic variational inference framework for generative and conditional models of text. While traditional variational methods derive an analytic approximation for the intractable distributions over latent variables, here we construct an inference network conditioned on the discrete text input to provide the variational distribution. We validate this framework on two very different text modelling applications, generative document modelling and supervised question answering. Our neural variational document model combines a continuous stochastic document representation with a bag-of-words generative model and achieves the lowest reported perplexities on two standard test corpora. The neural answer selection model employs a stochastic representation layer within an attention mechanism to extract the semantics between a question and answer pair. On two question answering benchmarks this model exceeds all previous published benchmarks.

1. Introduction

Probabilistic generative models underpin many successful applications within the field of natural language processing (NLP). Their popularity stems from their ability to use unlabelled data effectively, to incorporate abundant linguistic features, and to learn interpretable dependencies among data. However these successes are tempered by the fact that as the structure of such generative models becomes deeper and more complex, true Bayesian inference becomes intractable due to the high dimensional integrals required. Markov chain Monte Carlo (MCMC) (Neal, 1993; Andrieu et al., 2003) and variational inference (Jordan et al., 1999; Attias, 2000; Beal, 2003) are the standard approaches for approximating these integrals. However the computational cost of the former results in impractical training for the large and deep neural networks which are now fashionable, and the latter is conventionally confined due to the underestimation of posterior variance. The lack of effective and efficient inference methods hinders our ability to create highly expressive models of text, especially in the situation where the model is non-conjugate.

This paper introduces a neural variational framework for generative models of text, inspired by the variational autoencoder (Rezende et al., 2014; Kingma & Welling, 2014). The principle idea is to build an inference network, implemented by a deep neural network conditioned on text, to approximate the intractable distributions over the latent variables. Instead of providing an analytic approximation, as in traditional variational Bayes, neural variational inference learns to model the posterior probability, thus endowing the model with strong generalisation abilities. Due to the flexibility of deep neural networks, the inference network is capable of learning complicated non-linear distributions and processing structured inputs such as word sequences. Inference networks can be designed as, but not restricted to, multilayer perceptrons (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN), approaches which are rarely used in conventional generative models. By using the reparameterisation method (Rezende et al., 2014; Kingma & Welling, 2014), the inference network is trained through back-propagating unbiased and low variance gradients w.r.t. the latent variables. Within this framework, we propose a Neural Variational Document Model (NVDM) for document modelling and a Neural Answer Selection Model (NASM) for question answering, a task that selects the sentences that correctly answer a factoid question from a set of candidate sentences.

The NVDM (Figure 1) is an unsupervised generative model of text which aims to extract a continuous semantic latent variable for each document. This model can be interpreted as a variational auto-encoder: an MLP encoder (inference

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

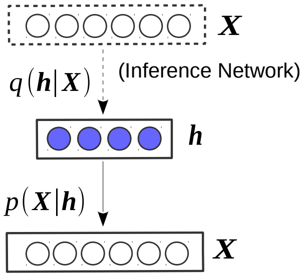

$\boldsymbol{X}$
$q(\boldsymbol{h}|\boldsymbol{X})$
(Inference Network)
$\boldsymbol{h}$
$p(\boldsymbol{X}|\boldsymbol{h})$
$\boldsymbol{X}$

Figure 1. NVDM for document modelling.

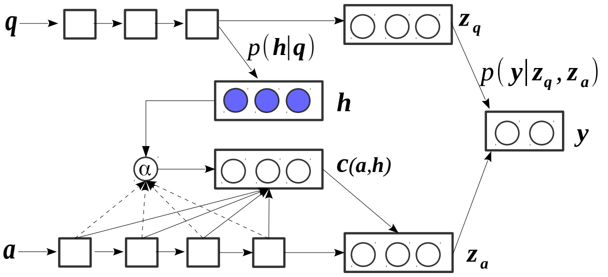


Figure 2. NASM for question answer selection.

network) compresses the bag-of-words document representation into a continuous latent distribution, and a softmax decoder (generative model) reconstructs the document by generating the words independently. A primary feature of NVDM is that each word is generated directly from a dense continuous document representation instead of the more common binary semantic vector (Hinton & Salakhutdinov, 2009; Larochelle & Lauly, 2012; Srivastava et al., 2013; Mnih & Gregor, 2014). Our experiments demonstrate that our neural document model achieves the state-of-the-art perplexities on the *20NewsGroups* and *RCV1-v2*.

The NASM (Figure 2) is a supervised conditional model which imbues LSTMs (Hochreiter & Schmidhuber, 1997) with a latent stochastic attention mechanism to model the semantics of question-answer pairs and predict their relatedness. The attention model is designed to focus on the phrases of an answer that are strongly connected to the question semantics and is modelled by a latent distribution. This mechanism allows the model to deal with the ambiguity inherent in the task and learns pair-specific representations that are more effective at predicting answer matches, rather than independent embeddings of question and answer sentences. Bayesian inference provides a natural safeguard against overfitting, especially as the training sets available for this task are small. The experiments show that the LSTM with a latent stochastic attention mechanism learns an effective attention model and outperforms both previously published results, and our own strong non-stochastic attention baselines.

In summary, we demonstrate the effectiveness of neural variational inference for text processing on two diverse tasks. These models are simple, expressive and can be trained efficiently with the highly scalable stochastic gradient back-propagation. Our neural variational framework is suitable for both unsupervised and supervised learning tasks, and can be generalised to incorporate any type of neural networks.

2. Neural Variational Inference Framework

Latent variable modelling is popular in many NLP problems, but it is non-trivial to carry out effective and efficient inference for models with complex and deep structure. In this section we introduce a generic neural variational inference framework that we apply to both the unsupervised NVDM and supervised NASM in the follow sections.

We define a generative model with a latent variable $\boldsymbol{h}$, which can be considered as the stochastic units in deep neural networks. We designate the observed parent and child nodes of $\boldsymbol{h}$ as $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively. Hence, the joint distribution of the generative model is $p_\theta(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{h}} p_\theta(\boldsymbol{y}|\boldsymbol{h})p_\theta(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{x})$, and the variational lower bound $\mathcal{L}$ is derived as:

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{h})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{h})p_\theta(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{x}) - \log q(\boldsymbol{h})] \qquad (1)$$
$$\leqslant \log \int \frac{q(\boldsymbol{h})}{q(\boldsymbol{h})} p_\theta(\boldsymbol{y}|\boldsymbol{h})p_\theta(\boldsymbol{h}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{h} = \log p_\theta(\boldsymbol{x},\boldsymbol{y})$$

where θ parameterises the generative distributions $p_\theta(\boldsymbol{y}|\boldsymbol{h})$ and $p_\theta(\boldsymbol{h}|\boldsymbol{x})$. In order to have a tight lower bound, the variational distribution $q(\boldsymbol{h})$ should approach the true posterior $p(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{y})$. Here, we employ a parameterised diagonal Gaussian $\mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}(\boldsymbol{x},\boldsymbol{y}), \mathrm{diag}(\boldsymbol{\sigma}^2(\boldsymbol{x},\boldsymbol{y})))$ as $q_\phi(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{y})$. The three steps to construct the inference network are:

1. Construct vector representations of the observed variables: $\boldsymbol{u} = f_x(\boldsymbol{x})$, $\boldsymbol{v} = f_y(\boldsymbol{y})$.
2. Assemble a joint representation: $\boldsymbol{\pi} = g(\boldsymbol{u},\boldsymbol{v})$.
3. Parameterise the variational distribution over the latent variable: $\boldsymbol{\mu} = l_1(\boldsymbol{\pi}), \log\boldsymbol{\sigma} = l_2(\boldsymbol{\pi})$.

$f_x(\cdot)$ and $f_y(\cdot)$ can be any type of deep neural networks that are suitable for the observed data; $g(\cdot)$ is an MLP that concatenates the vector representations of the conditioning variables; $l(\cdot)$ is a linear transformation which outputs the parameters of the Gaussian distribution. By sampling from the variational distribution, $\boldsymbol{h} \sim q_\phi(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{y})$, we are able to carry out stochastic back-propagation to optimise the lower bound (Eq. 1).

During training, the model parameters θ together with the inference network parameters ϕ are updated by stochastic back-propagation based on the samples $\boldsymbol{h}$ drawn from $q_\phi(\boldsymbol{h}|\boldsymbol{x},\boldsymbol{y})$. For the gradients w.r.t. θ, we have the form:

$$\nabla_\theta\mathcal{L} \simeq \tfrac{1}{L}\textstyle\sum_{l=1}^{L} \nabla_\theta \log p_\theta(\boldsymbol{y}|\boldsymbol{h}^{(l)})p_\theta(\boldsymbol{h}^{(l)}|\boldsymbol{x}) \qquad (2)$$

For the gradients w.r.t. ϕ we reparameterise $\boldsymbol{h} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$ and sample $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \boldsymbol{I})$ to reduce the variance in stochastic estimation (Rezende et al., 2014; Kingma & Welling, 2014). The update of ϕ can be carried out by back-propagating the gradients w.r.t. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$:

$$s(\boldsymbol{h}) = \log p_\theta(\boldsymbol{y}|\boldsymbol{h}) p_\theta(\boldsymbol{h}|\boldsymbol{x}) - \log q_\phi(\boldsymbol{h}|\boldsymbol{x}, \boldsymbol{y})$$

$$\nabla_\mu \mathcal{L} \simeq \tfrac{1}{L} \textstyle\sum_{l=1}^{L} \nabla_{\boldsymbol{h}^{(l)}} [s(\boldsymbol{h}^{(l)})] \quad (3)$$

$$\nabla_\sigma \mathcal{L} \simeq \tfrac{1}{2L} \textstyle\sum_{l=1}^{L} \boldsymbol{\epsilon}^{(l)} \nabla_{\boldsymbol{h}^{(l)}} [s(\boldsymbol{h}^{(l)})] \quad (4)$$

It is worth mentioning that unsupervised learning is a special case of the neural variational framework where $\boldsymbol{h}$ has no parent node $\boldsymbol{x}$. In that case $\boldsymbol{h}$ is directly drawn from the prior $p(\boldsymbol{h})$ instead of the conditional distribution $p_\theta(\boldsymbol{h}|\boldsymbol{x})$, and $s(\boldsymbol{h}) = \log p_\theta(\boldsymbol{y}|\boldsymbol{h}) p_\theta(\boldsymbol{h}) - \log q_\phi(\boldsymbol{h}|\boldsymbol{y})$.

Here we only discuss the scenario where the latent variables are continuous and the parameterised diagonal Gaussian is employed as the variational distribution. However the framework is also suitable for discrete units, and the only modification needed is to replace the Gaussian with a multinomial parameterised by the outputs of a softmax function. Though the reparameterisation trick for continuous variables is not applicable for this case, a policy gradient approach (Mnih & Gregor, 2014) can help to alleviate the high variance problem during stochastic estimation. (Kingma et al., 2014) proposed a variational inference framework for semi-supervised learning, but the prior distribution over the hidden variable $p(\boldsymbol{h})$ remains as the standard Gaussian prior, while we apply a conditional parameterised Gaussian distribution, which is jointly learned with the variational distribution.

3. Neural Variational Document Model

The Neural Variational Document Model (Figure 1) is a simple instance of unsupervised learning where a continuous hidden variable $\boldsymbol{h} \in \mathbb{R}^K$, which generates all the words in a document independently, is introduced to represent its semantic content. Let $\boldsymbol{X} \in \mathbb{R}^{|V|}$ be the bag-of-words representation of a document and $\boldsymbol{x}_i \in \mathbb{R}^{|V|}$ be the one-hot representation of the word at position i.

As an unsupervised generative model, we could interpret NVDM as a variational autoencoder: an MLP encoder $q(\boldsymbol{h}|\boldsymbol{X})$ compresses document representations into continuous hidden vectors ($\boldsymbol{X} \rightarrow \boldsymbol{h}$); a softmax decoder $p(\boldsymbol{X}|\boldsymbol{h}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{h})$ reconstructs the documents by independently generating the words ($\boldsymbol{h} \rightarrow \{\boldsymbol{x}_i\}$). To maximise the log-likelihood $\log \sum_{\boldsymbol{h}} p(\boldsymbol{X}|\boldsymbol{h}) p(\boldsymbol{h})$ of documents, we derive the lower bound:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{X})} \left[\sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) \right] - D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X}) \| p(\boldsymbol{h})] \quad (5)$$

where N is the number of words in the document and $p(\boldsymbol{h})$ is a Gaussian prior for $\boldsymbol{h}$. Here, we consider N is observed for all the documents. The conditional probability over words $p_\theta(\boldsymbol{x}_i|\boldsymbol{h})$ (decoder) is modelled by multinomial logistic regression and shared across documents:

$$p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) = \frac{\exp\{-E(\boldsymbol{x}_i; \boldsymbol{h}, \theta))\}}{\sum_{j=1}^{|V|} \exp\{-E(\boldsymbol{x}_j; \boldsymbol{h}, \theta)\}} \quad (6)$$

$$E(\boldsymbol{x}_i; \boldsymbol{h}, \theta) = -\boldsymbol{h}^T \boldsymbol{R} \boldsymbol{x}_i - \boldsymbol{b}_{x_i} \quad (7)$$

where $\boldsymbol{R} \in \mathbb{R}^{K \times |V|}$ learns the semantic word embeddings and $\boldsymbol{b}_{x_i}$ represents the bias term.

As there is no supervision information for the latent semantics, $\boldsymbol{h}$, the posterior approximation $q_\phi(\boldsymbol{h}|\boldsymbol{X})$ is only conditioned on the current document $\boldsymbol{X}$. The inference network $q_\phi(\boldsymbol{h}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}(\boldsymbol{X}), diag(\boldsymbol{\sigma}^2(\boldsymbol{X})))$ is modelled as:

$$\boldsymbol{\pi} = g(f_X^{\mathrm{MLP}}(\boldsymbol{X})) \quad (8)$$

$$\boldsymbol{\mu} = l_1(\boldsymbol{\pi}), \log \boldsymbol{\sigma} = l_2(\boldsymbol{\pi}) \quad (9)$$

For each document $\boldsymbol{X}$, the neural network generates its own parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ that parameterise the latent distribution over document semantics $\boldsymbol{h}$. Based on the samples $\boldsymbol{h} \sim q_\phi(\boldsymbol{h}|\boldsymbol{X})$, the lower bound (Eq. 5) can be optimised by back-propagating the stochastic gradients w.r.t. θ and ϕ.

Since $p(\boldsymbol{h})$ is a standard Gaussian prior, the Gaussian KL-Divergence $D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X}) \| p(\boldsymbol{h})]$ can be computed analytically to further lower the variance of the gradients. Moreover, it also acts as a regulariser for updating the parameters of the inference network $q_\phi(\boldsymbol{h}|\boldsymbol{X})$.

4. Neural Answer Selection Model

Answer sentence selection is a question answering paradigm where a model must identify the correct sentences answering a factual question from a set of candidate sentences. Assume a question $\boldsymbol{q}$ is associated with a set of answer sentences $\{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_n\}$, together with their judgements $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n\}$, where $\boldsymbol{y}_m = 1$ if the answer $\boldsymbol{a}_m$ is correct and $\boldsymbol{y}_m = 0$ otherwise. This is a classification task where we treat each training data point as a triple $(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$ while predicting $\boldsymbol{y}$ for the unlabelled question-answer pair $(\boldsymbol{q}, \boldsymbol{a})$.

The Neural Answer Selection Model (Figure 2) is a supervised model that learns the question and answer representations and predicts their relatedness. It employs two different LSTMs to embed raw question inputs $\boldsymbol{q}$ and answer inputs $\boldsymbol{a}$. Let $\boldsymbol{s}_q(j)$ and $\boldsymbol{s}_a(i)$ be the state outputs of the two LSTMs, and i, j be the positions of the states. Conventionally, the last state outputs $\boldsymbol{s}_q(|\boldsymbol{q}|)$ and $\boldsymbol{s}_a(|\boldsymbol{a}|)$, as the independent question and answer representations, can be used for relatedness prediction. In NASM, however, we aim to learn pair-specific representations through a latent attention mechanism, which is more effective for pair relatedness prediction.

NASM applies an attention model to focus on the words in the answer sentence that are prominent for predicting the answer matched to the current question. Instead of using a deterministic question vector, such as $\boldsymbol{s}_q(|\boldsymbol{q}|)$, NASM employs a latent distribution $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ to model the question semantics, which is a parameterised diagonal Gaussian $\mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}(\boldsymbol{q}), \text{diag}(\boldsymbol{\sigma}^2(\boldsymbol{q})))$. Therefore, the attention model extracts a context vector $\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h})$ by iteratively attending to the answer tokens based on the stochastic vector $\boldsymbol{h} \sim p_\theta(\boldsymbol{h}|\boldsymbol{q})$. In doing so the model is able to adapt to the ambiguity inherent in questions and obtain salient information through attention. Compared to its deterministic counterpart (applying $\boldsymbol{s}_q(|\boldsymbol{q}|)$ as the question semantics), the stochastic units incorporated into NASM allow multi-modal attention distributions. Further, by marginalising over the latent variables, NASM is more robust against overfitting, which is important for small question answering training sets.

In this model, the conditional distribution $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ is:

$$\boldsymbol{\pi}_\theta = g_\theta(f_q^{\text{LSTM}}(\boldsymbol{q})) = g_\theta(\boldsymbol{s}_q(|\boldsymbol{q}|)) \tag{10}$$

$$\boldsymbol{\mu}_\theta = l_1(\boldsymbol{\pi}_\theta), \log \boldsymbol{\sigma}_\theta = l_2(\boldsymbol{\pi}_\theta) \tag{11}$$

For each question $\boldsymbol{q}$, the neural network generates the corresponding parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ that parameterise the latent distribution over question semantics $\boldsymbol{h}$. Following Bahdanau et al. (2015), the attention model is defined as:

$$\alpha(i) \propto \exp(\boldsymbol{W}_\alpha^T \tanh(\boldsymbol{W}_h \boldsymbol{h} + \boldsymbol{W}_s \boldsymbol{s}_a(i))) \tag{12}$$

$$\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) = \sum\nolimits_i \boldsymbol{s}_a(i)\alpha(i) \tag{13}$$

$$\boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h}) = \tanh\left(\boldsymbol{W}_a \boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) + \boldsymbol{W}_n \boldsymbol{s}_a(|\boldsymbol{a}|)\right) \tag{14}$$

where $\alpha(i)$ is the normalised attention score at answer token i, and the context vector $\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h})$ is the weighted sum of all the state outputs $\boldsymbol{s}_a(i)$. We adopt $\boldsymbol{z}_q(\boldsymbol{q}), \boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h})$ as the question and answer representations for predicting their relatedness $\boldsymbol{y}$. $\boldsymbol{z}_q(\boldsymbol{q})$ is a deterministic vector that is equal to $\boldsymbol{s}_q(|\boldsymbol{q}|)$, while $\boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h})$ is a combination of the sequence output $\boldsymbol{s}_a(|\boldsymbol{a}|)$ and the context vector $\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h})$ (Eq. 14). For the prediction of pair relatedness $\boldsymbol{y}$, we model the conditional probability distribution $p_\theta(\boldsymbol{y}|\boldsymbol{z}_q, \boldsymbol{z}_a)$ by sigmoid function:

$$p_\theta(\boldsymbol{y} = 1|\boldsymbol{z}_q, \boldsymbol{z}_a) = \sigma\left(\boldsymbol{z}_q^T \mathbf{M} \boldsymbol{z}_a + b\right) \tag{15}$$

To maximise the log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{a})$ we use the variational lower bound:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{q_\phi(\boldsymbol{h})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{z}_q(\boldsymbol{q}), \boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h}))] - D_{\text{KL}}(q_\phi(\boldsymbol{h})||p_\theta(\boldsymbol{h}|\boldsymbol{q})) \\
&\leqslant \log \int p_\theta(\boldsymbol{y}|\boldsymbol{z}_q(\boldsymbol{q}), \boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h})) p_\theta(\boldsymbol{h}|\boldsymbol{q}) d\boldsymbol{h} \\
&= \log p(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{a})
\end{aligned} \tag{16}$$

Following the neural variational inference framework, we construct a deep neural network as the inference network $q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}_\phi(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})))$:

$$\begin{aligned}
\boldsymbol{\pi}_\phi &= g_\phi(f_q^{\text{LSTM}}(\boldsymbol{q}), f_a^{\text{LSTM}}(\boldsymbol{a}), f_y(\boldsymbol{y})) \\
&= g_\phi(\boldsymbol{s}_q(|\boldsymbol{q}|), \boldsymbol{s}_a(|\boldsymbol{a}|), \boldsymbol{s}_y)
\end{aligned} \tag{17}$$

$$\boldsymbol{\mu}_\phi = l_3(\boldsymbol{\pi}_\phi), \log \boldsymbol{\sigma}_\phi = l_4(\boldsymbol{\pi}_\phi) \tag{18}$$

where $\boldsymbol{q}$ and $\boldsymbol{a}$ are also modelled by LSTMs[1], and the relatedness label $\boldsymbol{y}$ is modelled by a simple linear transformation into the vector $\boldsymbol{s}_y$. According to the joint representation $\boldsymbol{\pi}_\phi$, we then generate the parameters $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}_\phi$, which parameterise the variational distribution over the question semantics $\boldsymbol{h}$. To emphasise, though both $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ and $q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$ are modelled as parameterised Gaussian distributions, $q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$ as an approximation only functions during inference by producing samples to compute the stochastic gradients, while $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ is the generative distribution that generates the samples for predicting the question-answer relatedness $\boldsymbol{y}$.

Based on the samples $\boldsymbol{h} \sim q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$, we use SGVB to optimise the lower bound (Eq.16). The model parameters θ and the inference network parameters ϕ are updated jointly using their stochastic gradients. In this case, similar to the NVDM, the Gaussian KL divergence $D_{\text{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}))||p_\theta(\boldsymbol{h}|\boldsymbol{q})]$ can be analytically computed during training process.

5. Experiments

5.1. Dataset & Setup for Document Modelling

We experiment with NVDM on two standard news corpora: the *20NewsGroups*[2] and the Reuters *RCV1-v2*[3]. The former is a collection of newsgroup documents, consisting of 11,314 training and 7,531 test articles. The latter is a large collection from Reuters newswire stories with 794,414 training and 10,000 test cases. The vocabulary size of these two datasets are set as 2,000 and 10,000.

To make a direct comparison with the prior work we follow the same preprocessing procedure and setup as Hinton & Salakhutdinov (2009), Larochelle & Lauly (2012), Srivastava et al. (2013), and Mnih & Gregor (2014). We train NVDM models with 50 and 200 dimensional document representations respectively. For the inference network, we use an MLP (Eq. 8) with 2 layers and 500 dimension rectifier linear units, which converts document representations into embeddings. During training we carry out stochastic estimation by taking one sample for estimating the stochastic gradients, while in prediction we use 20 samples for predicting document perplexity. The model is trained by

[1] In this case, the LSTMs for $\boldsymbol{q}$ and $\boldsymbol{a}$ are shared by the inference network and the generative model, but there is no restriction on using different LSTMs in the inference network.

[2] http://qwone.com/ jason/20Newsgroups

[3] http://trec.nist.gov/data/reuters/reuters.html

Model	Dim	20News	RCV1
LDA	50	1091	1437
LDA	200	1058	1142
RSM	50	953	988
docNADE	50	896	742
SBN	50	909	784
fDARN	50	917	724
fDARN	200	—	598
NVDM	50	**836**	563
NVDM	200	852	**550**

(a) Perplexity on test dataset.

Word	**weapons**	**medical**	**companies**	**define**	**israel**	**book**
NVDM	guns	medicine	expensive	defined	israeli	books
	weapon	health	industry	definition	arab	reference
	gun	treatment	company	printf	arabs	guide
	militia	disease	market	int	lebanon	writing
	armed	patients	buy	sufficient	lebanese	pages
NADE	weapon	treatment	demand	defined	israeli	reading
	shooting	medecine	commercial	definition	israelis	read
	firearms	patients	agency	refer	arab	books
	assault	process	company	make	palestinian	relevent
	armed	studies	credit	examples	arabs	collection

(b) The five nearest words in the semantic space.

Table 1. For the experimental results in (a), LDA (Blei et al., 2003) is a traditional topic model that models documents by mixtures of topics, RSM (Hinton & Salakhutdinov, 2009) is an undirected topic model implemented by restricted Boltzmann machines, and docNADE (Larochelle & Lauly, 2012) is a neural topic model based on autoregressive assumption. The models based on Sigmoid Belief Networks (SBN) and Deep AutoRegressive Neural Network (DARN) structures are implemented by Mnih & Gregor (2014), which employs an MLP to build a Monte Carlo control variate estimator for stochastic estimation.

Adam (Kingma & Ba, 2015) and tuned by hold-out validation perplexity. We alternately optimise the generative model and the inference network by fixing the parameters of one while updating the parameters of the other.

5.2. Experiments on Document Modelling

Table 1a presents the test document perplexity. The first column lists the models, and the second column shows the dimension of latent variables used in the experiments. The final two columns present the perplexity achieved by each topic model on the *20NewsGroups* and *RCV1-v2* datasets. In document modelling, perplexity is computed by $exp(-\frac{1}{D}\sum_{n}^{N_d}\frac{1}{N_d}\log p(\boldsymbol{X}_d))$, where D is the number of documents, N_d represents the length of the dth document and $\log p(\boldsymbol{X}) = \log\int p(\boldsymbol{X}|\boldsymbol{h})p(\boldsymbol{h})d\boldsymbol{h}$ is the log probability of the words in the document. Since $\log p(\boldsymbol{X})$ is intractable in the NVDM, we use the variational lower bound (which is an upper bound on perplexity) to compute the perplexity following Mnih & Gregor (2014).

While all the baseline models listed in Table 1a apply discrete latent variables, here NVDM employs a continuous stochastic document representation. The experimental results indicate that NVDM achieves the best performance on both datasets. For the experiments on *RCV1-v2* dataset, the NVDM with latent variable of 50 dimension performs even better than the fDARN with 200 dimension. It demonstrates that our document model with continuous latent variables has higher expressiveness and better generalisation ability. Table 1b compares the 5 nearest words selected according to the semantic vector learned from NVDM and docNADE.

In addition to the perplexities, we also qualitatively evaluate the semantic information learned by NVDM on the *20NewsGroups* dataset with latent variables of 50 dimension. We assume each dimension in the latent space represents a topic that corresponds to a specific semantic meaning. Table 2 presents 5 randomly selected topics with 10 words that have the strongest positive connection with the topic. Based on the words in each column, we can deduce their corresponding topics as: *Space, Religion, Encryption, Sport* and *Policy*. Although the model does not impose independent interpretability on the latent representation dimensions, we still see that the NVDM learns locally interpretable structure.

Space	*Religion*	*Encryption*	*Sport*	*Policy*
orbit	muslims	rsa	goals	bush
lunar	worship	cryptography	pts	resources
solar	belief	crypto	teams	charles
shuttle	genocide	keys	league	austin
moon	jews	pgp	team	bill
launch	islam	license	players	resolution
fuel	christianity	secure	nhl	mr
nasa	atheists	key	stats	misc
satellite	muslim	escrow	min	piece
japanese	religious	trust	buf	marc

Table 2. The topics learned by NVDM on 20News.

5.3. Dataset & Setup for Answer Sentence Selection

We experiment on two answer selection datasets, the *QASent* and the *WikiQA* datasets. *QASent* (Wang et al., 2007) is created from the TREC QA track, and the *WikiQA* (Yang et al., 2015) is constructed from Wikipedia, which is less noisy and less biased towards lexical overlap[4]. Table 3 summarises the statistics of the two datasets.

[4] Yang et al. (2015) provide detailed explanation of the differences between the two datasets.

Source	Set	Questions	QA Pairs	Judgement
QASent	Train	1,229	53,417	automatic
	Dev	82	1,148	manual
	Test	100	1,517	manual
WikiQA	Train	2,118	20,360	manual
	Dev	296	2,733	manual
	Test	633	6,165	manual

Table 3. Statistics of *QASent* and *WikiQA*. Judgement denotes whether correctness was determined automatically or by human annotators.

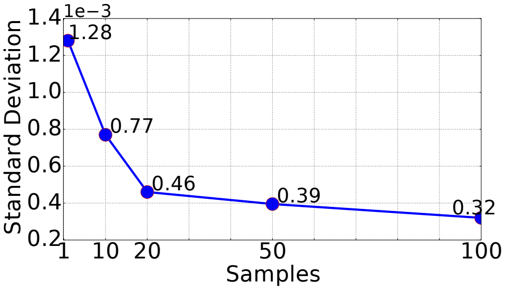


Figure 3. The standard deviations of MAP scores computed by running 10 NASM models on WikiQA with different numbers of samples.

Model	QASent		WikiQA	
	MAP	MRR	MAP	MRR
Published Models				
PV	0.5213	0.6023	0.5110	0.5160
Bigram-CNN	0.5693	0.6613	0.6190	0.6281
Deep CNN	0.5719	0.6621	—	—
PV + Cnt	0.6762	0.7514	0.5976	0.6058
WA	0.7063	0.7740	—	—
LCLR	0.7092	0.7700	0.5993	0.6068
Bigram-CNN + Cnt	0.7113	0.7846	0.6520	0.6652
Deep CNN + Cnt	0.7186	0.7826	—	—
Our Models				
LSTM	0.6436	0.7235	0.6552	0.6747
LSTM + Att	0.6451	0.7316	0.6639	0.6828
NASM	**0.6501**	**0.7324**	**0.6705**	**0.6914**
LSTM + Cnt	0.7228	0.7986	0.6820	0.6988
LSTM + Att + Cnt	0.7289	0.8072	0.6855	0.7041
NASM + Cnt	**0.7339**	**0.8117**	**0.6886**	**0.7069**

Table 4. Results of our models (LSTM, LSTM + Att, NASM) in comparison with other state of the art models on the *QASent* and *WikiQA* dataset. PV is the paragraph vector (Le & Mikolov, 2014). Bigram-CNN is the simple convolutional model reported in (Yu et al., 2014). Deep CNN is the deep convolutional model from (Severyn, 2015). WA is a model based on word alignment (Wang & Ittycheriah, 2015). LCLR is the SVM-based classifier trained using a set of features. Model + Cnt means that the result is obtained from a combination of a lexical overlap feature and the output from the distributional model.

In order to investigate the effectiveness of our NASM model we also implemented two strong baseline models — a vanilla LSTM model (LSTM) and an LSTM model with a deterministic attention mechanism (LSTM+Att). The former directly applies the QA matching function (Eq. 15) on the independent question and answer representations which are the last state outputs $\boldsymbol{s}_q(|\boldsymbol{q}|)$ and $\boldsymbol{s}_a(|\boldsymbol{a}|)$ from the question and answer LSTM models. The latter adds an attention model to learn pair-specific representation for prediction on the basis of the vanilla LSTM. Moreover, LSTM+Att is the deterministic counterpart of NASM, which has the same neural network architecture as NASM. The only difference is that it replaces the stochastic units $\boldsymbol{h}$ with deterministic ones, and no inference network is required to carry out stochastic estimation. Following previous work, for each of our models we also add a lexical overlap feature by combining a co-occurrence word count feature with the probability generated from the neural model. MAP and MRR are adopted as the evaluation metrics for this task.

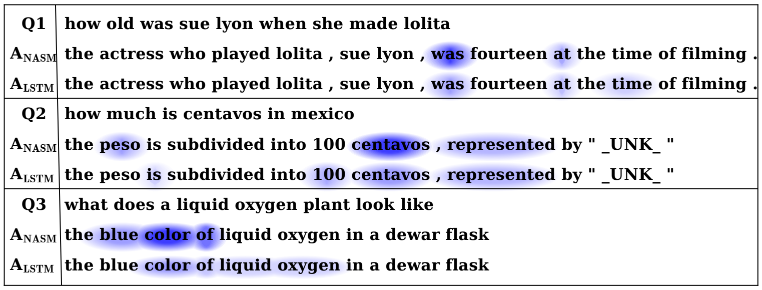
To facilitate direct comparison with previous work we follow the same experimental setup as Yu et al. (2014) and Severyn (2015). The word embeddings ($K = 50$) are obtained by running the `word2vec` tool (Mikolov et al., 2013) on the English Wikipedia dump and the *AQUAINT*[5] corpus. We use LSTMs with 3 layers and 50 hidden units, and apply 40% dropout after the embedding layer. For the construction of the inference network, we use an MLP (Eq. 10) with 2 layers and tanh units of 50 dimension, and an MLP (Eq. 17) with 2 layers and tanh units of 150 dimension for modelling the joint representation. During training we carry out stochastic estimation by taking one sample for computing the gradients, while in prediction we use 20 samples to calculate the expectation of the lower bound. Figure 3 presents the standard deviation of NASM's MAP scores while using different numbers of samples. Considering the trade-off between computational cost and variance, we chose 20 samples for prediction in all the experiments. The models are trained using Adam (Kingma & Ba, 2015), with hyperparameters selected by optimising the MAP score on the development set.

5.4. Experiments on Answer Sentence Selection

Table 4 compares the results of our models with current state-of-the-art models on both answer selection datasets. On the *QASent* dataset, our vanilla LSTM model outperforms the deep CNN [6] model by approximately 7% on

[5] https://catalog.ldc.upenn.edu/LDC2002T31

[6] As stated in (Yih et al., 2013) that the evaluation scripts used by previous work are noisy — 4 out of 72 questions in the test set are treated answered incorrectly. This makes the MAP and MRR scores $\sim 4\%$ lower than the *true* scores. Since Severyn (2015) and Wang & Ittycheriah (2015) use a cleaned-up evaluation scripts, we apply the original noisy scripts to re-evaluate their outputs in



Q1	how old was sue lyon when she made lolita
A_{NASM}	the actress who played lolita , sue lyon , was fourteen at the time of filming .
A_{LSTM}	the actress who played lolita , sue lyon , was fourteen at the time of filming .
Q2	how much is centavos in mexico
A_{NASM}	the peso is subdivided into 100 centavos , represented by " _UNK_ "
A_{LSTM}	the peso is subdivided into 100 centavos , represented by " _UNK_ "
Q3	what does a liquid oxygen plant look like
A_{NASM}	the blue color of liquid oxygen in a dewar flask
A_{LSTM}	the blue color of liquid oxygen in a dewar flask

Figure 4. A visualisation of attention scores on answer sentences.

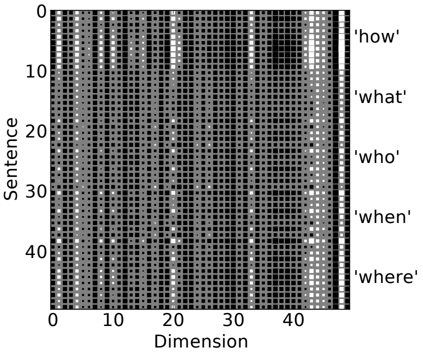


Figure 5. Hinton diagrams of the log standard deviations.

MAP and 6% on MRR. The LSTM+Att performs slightly better than the vanilla LSTM model, and our NASM improves the results further. Since the *QASent* dataset is biased towards lexical overlapping features, after combining with a co-occurrence word count feature, our best model NASM outperforms all the previous models, including both neural network based models and classifiers with a set of hand-crafted features (e.g. LCLR). Similarly, on the *WikiQA* dataset, all of our models outperform the previous distributional models by a large margin. By including a word count feature, our models improve further and achieve the state-of-the-art. Notably, on both datasets, our two LSTM-based models have set strong baselines and NASM works even better, which demonstrates the effectiveness of introducing stochastic units to model question semantics in this answer sentence selection task.

In Figure 4, we compare the effectiveness of the latent attention mechanism (NASM) and its deterministic counterpart (LSTM+Att) by visualising the attention scores on the answer sentences. For most of the negative answer sentences, neither of the two attention models can attend to reasonable words that are beneficial for predicting relatedness. But for the correct answer sentences, such as the ones in Figure 4, both attention models are able to capture crucial information by attending to different parts of the sentence based on the question semantics. Interestingly, compared to the deterministic counterpart LSTM+Att, our NASM assigns higher attention scores on the prominent words that are relevant to the question, which forms a more peaked distribution and in turn helps the model achieve better performance.

In order to have an intuitive observation on the latent distributions, we present Hinton diagrams of their log standard deviation parameters (Figure 5). In a Hinton diagram, the size of a square is proportional to a value's magnitude, and the colour (black/white) indicates its sign (positive/negative). In this case, we visualise the parameters of 50 conditional distributions $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ with the questions selected from 5 different groups, which start with 'how', 'what', 'who', 'when' and 'where'. All the log standard deviations are initialised as zero before training. According to Figure 5, we can see that the questions starting with 'how' have more white areas, which indicates higher variances or more uncertainties are in these dimensions. By contrast, the questions starting with 'what' have black squares in almost every dimension. Intuitively, it is more difficult to understand and answer the questions starting with 'how' than the others, while the 'what' questions commonly have explicit words indicating the possible answers. To validate this, we compute the stratified MAP scores based on different question type. The MAP of 'how' questions is 0.524 which is the lowest among the five groups. Hence empirically, 'how' questions are harder to 'understand and answer'.

6. Discussion

As shown in the experiments, neural variational inference brings consistent improvements on the performance of both NLP tasks. The basic intuition is that the latent distributions grant the ability to sum over all the possibilities in terms of semantics. From the perspective of optimisation, one of the most important reasons is that Bayesian learning guards against overfitting.

According to Eq. 5 in NVDM, since we adopt $p(\boldsymbol{h})$ as a standard Gaussian prior, the KL divergence term $D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})\|p(\boldsymbol{h})]$ can be analytically computed as $\frac{1}{2}(K - \|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\sigma}\|^2 + \log|\operatorname{diag}(\boldsymbol{\sigma}^2)|)$. It is not difficult to find that it actually acts as L2 regulariser when we update the $\boldsymbol{\mu}$. Similarly, in NASM (Eq. 16), we also have the KL divergence term $D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}))\|p_\theta(\boldsymbol{h}|\boldsymbol{q})]$. Different from NVDM, it attempts to minimise the distance between $q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}))$ and $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ that are both conditional distributions. Because $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ as well as $q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}))$ are learned during training, the two distributions are mutually restrained while being updated. Therefore, NVDM simply penalises the large $\boldsymbol{\mu}$ and encourages $q_\phi(\boldsymbol{h}|\boldsymbol{X})$ to

order to make the results directly comparable with previous work.

approach the prior $p(\boldsymbol{h})$ for every document $\boldsymbol{X}$, but in NASM, $p_\theta(\boldsymbol{h}|\boldsymbol{q})$ acts like a moving baseline distribution which regularises the update of $q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}))$ for every different conditions. In practice, we carry out early stopping by observing the prediction performance on development dataset for the question answer selection task. Using the same learning rate and neural network structure, LSTM+Att reaches optimal performance and starts to overfit on training dataset generally at the 20th iteration, while NASM starts to overfit around the 35th iteration.

More interestingly, in the question answer selection experiments, NASM learns more peaked attention scores than its deterministic counterpart LSTM+Att. For the update process of LSTM+Att, we find there exists a relatively big variance in the gradients w.r.t. question semantics (LSTM+Att applies deterministic $\boldsymbol{s}_q(|\boldsymbol{q}|)$ while NASM applies stochastic $\boldsymbol{h}$). This is because the training dataset is small and contains many negative answer sentences that brings no benefit but noise to the learning of the attention model. In contrast, for the update process of NASM, we observe more stable gradients w.r.t. the parameters of latent distributions. The optimisation of the lower bound on one hand maximises the conditional log-likelihood (that the deterministic counterpart cares about) and on the other hand minimises the KL-divergence (that regularises the gradients). Hence, each update of the lower bound actually keeps the gradients w.r.t. $\boldsymbol{\mu}$ from swinging heavily. Besides, since the values of $\boldsymbol{\sigma}$ are not very significant in this case, the distribution of attention scores mainly depends on $\boldsymbol{\mu}$. Therefore, the learning of the attention model benefits from the regularisation as well, and it explains the fact that NASM learns more peaked attention scores which in turn helps achieve a better prediction performance.

Since the computations of NVDM and NASM can be parallelised on GPU and only one sample is required during training process, it is very efficient to carry out the neural variational inference. Moreover, for both NVDM and NASM, all the parameters are updated by backpropagation. Thus, the increased computation time for the stochastic units only comes from the added parameters of the inference network.

7. Related Work

Training an inference network to approximate the variational distribution was first proposed in the context of Helmholtz machines (Hinton & Zemel, 1994; Hinton et al., 1995; Dayan & Hinton, 1996), but applications of these directed generative models come up against the problem of establishing low variance gradient estimators. Recent advances in neural variational inference mitigate this problem by reparameterising the continuous random variables (Rezende et al., 2014; Kingma & Welling, 2014), using control variates (Mnih & Gregor, 2014) or approximating the posterior with importance sampling (Bornschein & Bengio, 2015). The instantiations of these ideas (Gregor et al., 2015; Kingma et al., 2014; Ba et al., 2015) have demonstrated strong performance on the tasks of image processing. The recent variants of generative auto-encoder (Louizos et al., 2015; Makhzani et al., 2015) are also very competitive. Tang & Salakhutdinov (2013) applies the similar idea of introducing stochastic units for expression classification, but its inference is carried out by Monte Carlo EM algorithm with the reliance on importance sampling, which is less efficient and lack of scalability.

Another class of neural generative models make use of the autoregressive assumption (Larochelle & Murray, 2011; Uria et al., 2014; Germain et al., 2015; Gregor et al., 2014). Applications of these models on document modelling achieve significant improvements on generating documents, compared to conventional probabilistic topic models (Hofmann, 1999; Blei et al., 2003) and also the RBMs (Hinton & Salakhutdinov, 2009; Srivastava et al., 2013). While these models that use binary semantic vectors, our NVDM employs dense continuous document representations which are both expressive and easy to train. The semantic word vector model (Maas et al., 2011) also employs a continuous semantic vector to generate words, but the model is trained by MAP inference which does not permit the calculation of the posterior distribution. A very similar idea to NVDM is Bowman et al. (2015), which employs VAE to generate sentences from a continuous space.

Apart from the work mentioned above, there is other interesting work on question answering with deep neural networks. One of the popular streams is mapping factoid questions with answer triples in the knowledge base (Bordes et al., 2014a;b; Yih et al., 2014). Moreover, Weston et al. (2015); Sukhbaatar et al. (2015); Kumar et al. (2015) further exploit memory networks, where long-term memories act as dynamic knowledge bases. Another attention-based model (Hermann et al., 2015) applies the attentive network to help read and comprehend for long articles.

8. Conclusion

This paper introduced a deep neural variational inference framework for generative models of text. We experimented on two diverse tasks, document modelling and question answer selection tasks to demonstrate the effectiveness of this framework, where in both cases our models achieve state of the art performance. Apart from the promising results, our model also has the advantages of (1) simple, expressive, and efficient when training with the SGVB algorithm; (2) suitable for both unsupervised and supervised learning tasks; and (3) capable of generalising to incorporate any type of neural network.

References

Andrieu, Christophe, De Freitas, Nando, Doucet, Arnaud, and Jordan, Michael I. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

Attias, Hagai. A variational bayesian framework for graphical models. In *Proceedings of NIPS*, 2000.

Ba, Jimmy, Grosse, Roger, Salakhutdinov, Ruslan, and Frey, Brendan. Learning wake-sleep recurrent attention models. In *Proceedings of NIPS*, 2015.

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015.

Beal, Matthew James. *Variational algorithms for approximate Bayesian inference*. University of London, 2003.

Blei, David M, Ng, Andrew Y, and Jordan, Michael I. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

Bordes, Antoine, Chopra, Sumit, and Weston, Jason. Question answering with subgraph embeddings. In *Proceedings of EMNLP*, 2014a.

Bordes, Antoine, Weston, Jason, and Usunier, Nicolas. Open question answering with weakly supervised embedding models. In *Proceedings of ECML*, 2014b.

Bornschein, Jörg and Bengio, Yoshua. Reweighted wake-sleep. In *Proceedings of ICLR*, 2015.

Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M., Józefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015. URL http://arxiv.org/abs/1511.06349.

Dayan, Peter and Hinton, Geoffrey E. Varieties of helmholtz machine. *Neural Networks*, 9(8):1385–1403, 1996.

Germain, Mathieu, Gregor, Karol, Murray, Iain, and Larochelle, Hugo. Made: Masked autoencoder for distribution estimation. In *Proceedings of ICML*, 2015.

Gregor, Karol, Mnih, Andriy, and Wierstra, Daan. Deep autoregressive networks. In *Proceedings of ICML*, 2014.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. Draw: A recurrent neural network for image generation. In *Proceedings of ICML*, 2015.

Hermann, Karl Moritz, Kociský, Tomás, Grefenstette, Edward, Espeholt, Lasse, Kay, Will, Suleyman, Mustafa, and Blunsom, Phil. Teaching machines to read and comprehend. In *Proceedings of NIPS*, 2015.

Hinton, Geoffrey E and Salakhutdinov, Ruslan. Replicated softmax: an undirected topic model. In *Proceedings of NIPS*, 2009.

Hinton, Geoffrey E and Zemel, Richard S. Autoencoders, minimum description length, and helmholtz free energy. In *Proceedings of NIPS*, 1994.

Hinton, Geoffrey E, Dayan, Peter, Frey, Brendan J, and Neal, Radford M. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hofmann, Thomas. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, 1999.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. In *Proceedings of ICLR*, 2014.

Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Proceedings of NIPS*, 2014.

Kumar, Ankit, Irsoy, Ozan, Su, Jonathan, Bradbury, James, English, Robert, Pierce, Brian, Ondruska, Peter, Gulrajani, Ishaan, and Socher, Richard. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*, 2015.

Larochelle, Hugo and Lauly, Stanislas. A neural autoregressive topic model. In *Proceedings of NIPS*, 2012.

Larochelle, Hugo and Murray, Iain. The neural autoregressive distribution estimator. In *Proceedings of AISTATS*, 2011.

Le, Quoc V. and Mikolov, Tomas. Distributed representations of sentences and documents. In *Proceedings of ICML*, 2014.

Louizos, Christos, Swersky, Kevin, Li, Yujia, Welling, Max, and Zemel, Richard. The variational fair auto encoder. *arXiv preprint arXiv:1511.00830*, 2015.

Maas, Andrew L, Daly, Raymond E, Pham, Peter T, Huang, Dan, Ng, Andrew Y, and Potts, Christopher. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, 2011.

Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, and Goodfellow, Ian J. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015. URL http://arxiv.org/abs/1511.05644.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Gregory S., and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 2013.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *Proceedings of ICML*, 2014.

Neal, Radford M. Probabilistic inference using markov chain monte carlo methods. *Technical report: CRG-TR-93-1*, 1993.

Rezende, Danilo J, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of ICML*, 2014.

Severyn, Aliaksei. *Modelling input texts: from Tree Kernels to Deep Learning*. PhD thesis, University of Trento, 2015.

Srivastava, Nitish, Salakhutdinov, RR, and Hinton, Geoffrey. Modeling documents with deep boltzmann machines. In *Proceedings of UAI*, 2013.

Sukhbaatar, Sainbayar, Szlam, Arthur, Weston, Jason, and Fergus, Rob. End-to-end memory networks. In *Proceedings of NIPS*, 2015.

Tang, Yichuan and Salakhutdinov, Ruslan R. Learning stochastic feedforward neural networks. In *Proceedings NIPS*, 2013.

Uria, Benigno, Murray, Iain, and Larochelle, Hugo. A deep and tractable density estimator. In *Proceedings of ICML*, 2014.

Wang, Mengqiu, Smith, Noah A, and Mitamura, Teruko. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of EMNLP-CoNLL*, 2007.

Wang, Zhiguo and Ittycheriah, Abraham. Faq-based question answering via word alignment. *arXiv preprint arXiv:1507.02628*, 2015.

Weston, Jason, Chopra, Sumit, and Bordes, Antoine. Memory networks. In *Proceedings of ICLR*, 2015.

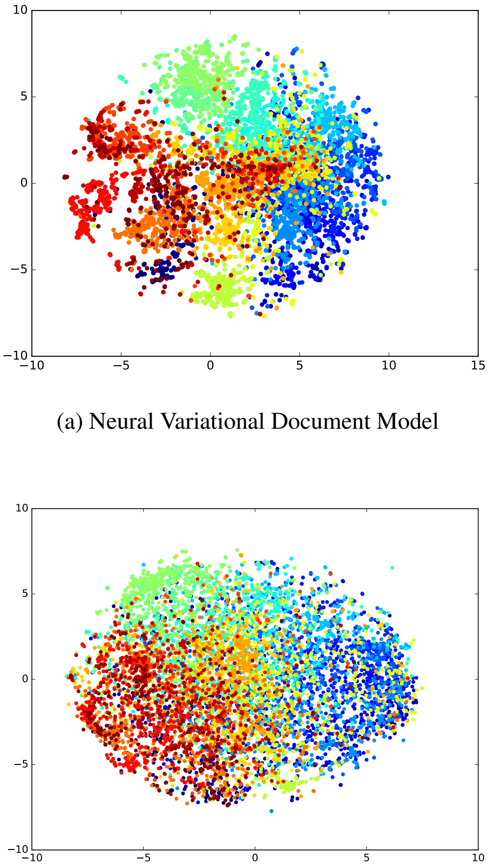
Yang, Yi, Yih, Wen-tau, and Meek, Christopher. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, 2015.

Yih, Wen-tau, Chang, Ming-Wei, Meek, Christopher, and Pastusiak, Andrzej. Question answering using enhanced lexical semantic models. In *Proceedings of ACL*, 2013.

Yih, Wen-tau, He, Xiaodong, and Meek, Christopher. Semantic parsing for single-relation question answering. In *Proceedings of ACL*, 2014.

Yu, Lei, Hermann, Karl Moritz, Blunsom, Phil, and Pulman, Stephen. Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*, 2014.

A. t-SNE Visualisation of Document Representations



(a) Neural Variational Document Model

(b) Semantic Word Vector

Figure 6. t-SNE visualisation of the document representations achieved by (a) NVDM and (b) SWV (Maas et al., 2011) on the held-out test dataset of *20NewsGroups*. The documents are collected from 20 different news groups, which correspond to the points with different colour in the figure.

B. Details of the Deep Neural Network Structures

B.1. Neural Variational Document Model

(1) Inference Network $q_\phi(\boldsymbol{h}|\boldsymbol{X})$:

$$\boldsymbol{\lambda} = \text{ReLU}(\boldsymbol{W}_1\boldsymbol{X} + \boldsymbol{b}_1) \quad (19)$$

$$\boldsymbol{\pi} = \text{ReLU}(\boldsymbol{W}_2\boldsymbol{\lambda} + \boldsymbol{b}_2) \quad (20)$$

$$\boldsymbol{\mu} = \boldsymbol{W}_3\boldsymbol{\pi} + \boldsymbol{b}_3 \quad (21)$$

$$\log\boldsymbol{\sigma} = \boldsymbol{W}_4\boldsymbol{\pi} + \boldsymbol{b}_4 \quad (22)$$

$$\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}), \text{diag}(\boldsymbol{\sigma}^2(\boldsymbol{X}))) \quad (23)$$

(2) Generative Model $p_\theta(\boldsymbol{X}|\boldsymbol{h})$:

$$\boldsymbol{e}_i = \exp(-\boldsymbol{h}^T\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{b}_{x_i}) \quad (24)$$

$$p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) = \frac{\boldsymbol{e}_i}{\sum_j^{|V|}\boldsymbol{e}_j} \quad (25)$$

$$p_\theta(\boldsymbol{X}|\boldsymbol{h}) = \prod_i^N p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) \quad (26)$$

(3) KL Divergence $D_{\text{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})||p(\boldsymbol{h})]$:

$$D_{\text{KL}} = -\frac{1}{2}(K - \|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\sigma}\|^2 + \log|\text{diag}(\boldsymbol{\sigma}^2)|)(27)$$

The variational lower bound to be optimised:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{X})}\left[\sum_{i=1}^{N}\log p_\theta(\boldsymbol{x}_i|\boldsymbol{h})\right] - D_{\text{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})||p(\boldsymbol{h})] \quad (28)$$

$$\approx \sum_{l=1}^{L}\sum_{i=1}^{N}\log p_\theta(\boldsymbol{x}_i|\boldsymbol{h}^{(l)}) + \frac{1}{2}(K - \|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\sigma}\|^2 + \log|\text{diag}(\boldsymbol{\sigma}^2)|) \quad (29)$$

B.2. Neural Answer Selection Model

(1) Inference Network $q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$:

$$\boldsymbol{s}_q(|\boldsymbol{q}|) = f_q^{\text{LSTM}}(\boldsymbol{q}) \quad (30)$$

$$\boldsymbol{s}_a(|\boldsymbol{a}|) = f_a^{\text{LSTM}}(\boldsymbol{a}) \quad (31)$$

$$\boldsymbol{s}_y = \boldsymbol{W}_5\boldsymbol{y} + \boldsymbol{b}_5 \quad (32)$$

$$\boldsymbol{\gamma} = \boldsymbol{s}_q(|\boldsymbol{q}|)||\boldsymbol{s}_a(|\boldsymbol{a}|)||\boldsymbol{s}_y \quad (33)$$

$$\boldsymbol{\lambda}_\phi = \tanh(\boldsymbol{W}_6\boldsymbol{\gamma} + \boldsymbol{b}_6) \quad (34)$$

$$\boldsymbol{\pi}_\phi = \tanh(\boldsymbol{W}_7\boldsymbol{\lambda}_{\boldsymbol{\phi}} + \boldsymbol{b}_7) \quad (35)$$

$$\boldsymbol{\mu}_\phi = \boldsymbol{W}_8\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_8 \quad (36)$$

$$\log\boldsymbol{\sigma}_\phi = \boldsymbol{W}_9\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_9 \quad (37)$$

$$\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}), \text{diag}(\boldsymbol{\sigma}_\phi^2(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}))) \quad (38)$$

(2) Generative Model

$p_\theta(\boldsymbol{h}|\boldsymbol{q})$:

$$\boldsymbol{\lambda}_\theta = \tanh(\boldsymbol{W}_1\boldsymbol{s}_q(|\boldsymbol{q}|) + \boldsymbol{b}_1) \quad (39)$$

$$\boldsymbol{\pi}_\theta = \tanh(\boldsymbol{W}_2\boldsymbol{\lambda}_\theta + \boldsymbol{b}_2) \quad (40)$$

$$\boldsymbol{\mu}_\theta = \boldsymbol{W}_3\boldsymbol{\pi}_\theta + \boldsymbol{b}_3 \quad (41)$$

$$\log\boldsymbol{\sigma}_\theta = \boldsymbol{W}_4\boldsymbol{\pi}_\theta + \boldsymbol{b}_4 \quad (42)$$

$p_\theta(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{h})$:

$$\boldsymbol{e}(i) = \boldsymbol{W}_\alpha^T\tanh(\boldsymbol{W}_h\boldsymbol{h} + \boldsymbol{W}_s\boldsymbol{s}_a(i)) \quad (43)$$

$$\alpha(i) = \frac{\boldsymbol{e}(i)}{\sum_j \boldsymbol{e}(j)} \quad (44)$$

$$\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) = \sum_i \boldsymbol{s}_a(i)\alpha(i) \quad (45)$$

$$\boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h}) = \tanh(\boldsymbol{W}_a\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) + \boldsymbol{W}_n\boldsymbol{s}_a(|\boldsymbol{a}|)) \quad (46)$$

$$\boldsymbol{z}_q(\boldsymbol{q}) = \boldsymbol{s}_q(|\boldsymbol{q}|) \quad (47)$$

$$p_\theta(\boldsymbol{y} = 1|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{h}) = \sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a + b) \quad (48)$$

(3) KL Divergence $D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})||p_\theta(\boldsymbol{h}|\boldsymbol{q})]$:

$$\begin{aligned} D_{\mathrm{KL}} = -\frac{1}{2}(&K + \log|\operatorname{diag}(\boldsymbol{\sigma}_\phi^2)| - \log|\operatorname{diag}(\boldsymbol{\sigma}_\theta^2)| \\ &- \operatorname{Tr}(\operatorname{diag}(\boldsymbol{\sigma}_\phi^2)\operatorname{diag}^{-1}(\boldsymbol{\sigma}_\theta^2)) \\ &- (\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)^T \operatorname{diag}^{-1}(\boldsymbol{\sigma}_\theta^2)(\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)) \end{aligned} \tag{49}$$

The variational lower bound to be optimised:

$$\begin{aligned} \mathcal{L} = &\mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{h})] \\ &- D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})||p_\theta(\boldsymbol{h}|\boldsymbol{q})] \end{aligned} \tag{50}$$

$$\begin{aligned} \approx &\sum_{l=1}^{L}[\boldsymbol{y}\log\sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a^{(l)} + b) \\ &+ (1-\boldsymbol{y})\log(1-\sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a^{(l)} + b))] \\ &+ \frac{1}{2}(K + \log|\operatorname{diag}(\boldsymbol{\sigma}_\phi^2)| - \log|\operatorname{diag}(\boldsymbol{\sigma}_\theta^2)| \\ &- \operatorname{Tr}(\operatorname{diag}(\boldsymbol{\sigma}_\phi^2)\operatorname{diag}^{-1}(\boldsymbol{\sigma}_\theta^2)) \\ &- (\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)^T \operatorname{diag}^{-1}(\boldsymbol{\sigma}_\theta^2)(\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)) \end{aligned} \tag{51}$$

C. Computational Complexity

The computational complexity of NVDM for a training document is $C_\phi + C_\theta = O(LK^2 + KSV)$. Here, $C_\phi = O(LK^2)$ represents the cost for the inference network to generate a sample, where L is the number of the layers in the inference network and K is the average dimension of these layers. Besides, $C_\theta = O(KSV)$ is the cost of reconstructing the document from a sample, where S is the average length of the documents and V represents the volume of words applied in this document model, which is conventionally much lager than K.

The computational complexity of NASM for a training question-answer pair is $C_\phi + C_\theta = O((L+S)K^2 + SW)$. The inference network needs $C_\phi = 2SW + 2K + LK^2 = O(LK^2 + SW)$. It takes $2SW + 2K$ to produce the joint representation for a question-answer pair and its label, where W is the total number of parameters of an LSTM and S is the average length of the sentences. Based on the joint representation, an MLP spends LK^2 to generate a sample, where L is the number of layers and K represents the average dimension. The generative model requires $C_\theta = 2SW + LK^2 + SK^2 + 5K^2 + 2K^2 = O((L+S)K^2 + SW)$. Similarly, it costs $2SW + LK^2$ to construct the generative latent distribution , where $2SW$ can be saved if the LSTMs are shared by the inference network and the generative model. Besides, the attention model takes $SK^2 + 5K^2$ and the relatedness prediction takes the last $2K^2$.

Since the computations of NVDM and NASM can be parallelised in GPU and only one sample is required during training process, it is very efficient to carry out the neural variational inference. As NVDM is an instantiation of variational auto-encoder, its computational complexity is the same as the deterministic auto-encoder. In addition, the computational complexity of LSTM+Att, the deterministic counterpart of NASM, is also $O((L+S)K^2 + SW)$. There is only $O(LK^2)$ time increase by introducing an inference network for NASM when compared to LSTM+Att.