Received: 16 July 2019 | Revised: 21 July 2020 | Accepted: 29 July 2020

DOI: 10.1111/exsy.12639

ORIGINAL ARTICLE

Expert Systems WILEY

Nested variational autoencoder for topic modelling on microtexts with word vectors

Trung Trinh | Tho Quan | Trung Mai

Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh, Vietnam

Correspondence
Tho Quan, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh, Vietnam.
Email: qttho@hcmut.edu.vn

Abstract

Most of the information on the Internet is represented in the form of *microtexts*, which are short text snippets such as news headlines or tweets. These sources of information are abundant, and mining these data could uncover meaningful insights. Topic modelling is one of the popular methods to extract knowledge from a collection of documents; however, conventional topic models such as latent Dirichlet allocation (LDA) are unable to perform well on short documents, mostly due to the scarcity of word co-occurrence statistics embedded in the data. The objective of our research is to create a topic model that can achieve great performances on microtexts while requiring a small runtime for scalability to large datasets. To solve the lack of information of microtexts, we allow our method to take advantage of word embeddings for additional knowledge of relationships between words. For speed and scalability, we apply autoencoding variational Bayes, an algorithm that can perform efficient black-box inference in probabilistic models. The result of our work is a novel topic model called the *nested variational autoencoder*, which is a distribution that takes into account word vectors and is parameterized by a neural network architecture. For optimization, the model is trained to approximate the posterior distribution of the original LDA model. Experiments show the improvements of our model on microtexts as well as its runtime advantage.

KEYWORDS

microtext, neural network, topic modelling, variational autoencoder, word embedding

1 | INTRODUCTION

The ubiquity of microtexts, which is due to the emergence of social media sites such as Facebook and Twitter, has become an increasingly valuable asset for mining information about the real world. In health care, by monitoring information posted by users on social networks, one can observe the status of public health (Paul et al., 2015). nEmesis (Sadilek et al., 2017) is a system deployed in combination with social media to prevent foodborne illness. On a broader scale, social platform data can be used to observe public ideas (Kennedy & Moss, 2015) and offer emergency service (Pandey & Purohit, 2018). The RSC system (Costa, Yamaguchi, Traina, Traina, & Faloutsos, 2015) is a system that monitors temporal human activities. More sophisticated mining tasks can be employed to detect fake news (Shu, Sliva, Wang, Tang, & Liu, 2017) or to detect themes on social media (Lazard, Scheinfeld, Bernhardt, Wilcox, & Suran, 2015).

Many of these applications could be distilled to inferring topics from these sources of information. Probabilistic models such as probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) and latent Dirichlet allocation (LDA) (Blei, Ng, & Jordan, 2003) have been successfully applied to long texts. These models operate on the assumption that each document comprises a small number of topics, each of which in turn consists of a subset of words. The topic proportions of each document as well as the distribution over the vocabulary of each topic are then learned from the

corpus using statistical methods such as Gibbs sampling (Griffiths & Steyvers, 2004) or variational inference (Blei et al., 2003). The effectiveness of these models strongly depends on the patterns of word cooccurrences within the corpus, which is fully and correctly represented in a large corpus of long texts. The sparsity of microtexts, on the other hand, has proven to be inadequate in presenting the relationship between words. Thus, microtexts remain challenging for these conventional topic modelling methods. One way to alleviate this problem is to introduce additional information that could help the model to uncover the true semantic relationships between words. Such methods include utilizing search results for matching similar text snippets (Sahami & Heilman, 2006), using a knowledgebase for microtext conceptualization (Song, Wang, Wang, Li, & Chen, 2011), and leveraging auxiliary long texts to enhance microtext clustering performance (O. Jin, Liu, Zhao, Yu, & Yang, 2011).

Recently, the introduction of word embeddings Mikolov, Chen, Corrado, and Jeffrey (2013); Pennington, Socher, and Manning (2014) has led to improvements in many natural language processing (NLP) tasks due to their ability to capture semantic relationships between words in a distributed fashion. In this model, each word is represented using a dense vector that contains the semantic and syntactic information about that word, and similar words tend to stay close to each other in the embedding space. One could expect that these latent features of words could be used to improve the performance of topic modelling on microtexts. In fact, there have been some studies that incorporate word vectors with available topic modelling methods such as LDA and have shown promising results (Das, Zaheer, & Dyer, 2015; Nguyen, Billingsley, Du, & Johnson, 2015). While the improvements are remarkable, these papers use Gibb sampling as the method to infer the parameters of the model, and although it has been proven theoretically to produce the best results compared to other methods, it requires a long time to converge.

With the advancement of deep learning in recent years, these methods have been used to replace the traditional mean field approximator in variational inference, which has led to the invention of variational autoencoders (VAEs) (Kingma & Welling, 2013; Rezende, Mohamed, & Wierstra, 2014). Most studies on VAEs employ the Gaussian distribution since the reparameterization trick (RT) for the Gaussian distribution is readily available (Kingma & Welling, 2013; Rezende et al., 2014). To approximate the Dirichlet prior of LDA, Srivastava and Sutton (2017) used a logistic normal distribution to approximate the Laplace approximation of the Dirichlet prior on the softmax basis. Their study provides evidence that VAEs could be effective for topic modelling, showing better results than traditional LDA with Gibbs sampling while enjoying much faster convergence.

The recent development in RT allows more distributions to be used in VAEs (Figurnov, Mohamed, & Mnih, 2018; Jang, Gu, & Poole, 2016; Maddison, Mnih, & Teh, 2016). This has opened up new possibilities for VAEs to better approximate complex probabilistic models, which include LDA. The experiment in Figurnov et al. (2018) indicates that using the Dirichlet prior directly in VAE produces lower perplexity than that of Srivastava and Sutton (2017), and the model also has the ability to further fine-tune the parameters of the Dirichlet prior, which has proven to greatly improve the performance of LDA (Wallach, Mimno, & McCallum, 2009).

In this paper, we introduce a new VAE model that can leverage additional information from word embeddings to perform topic modelling on microtexts. Our motivation originates from the need of an algorithm that is effective in detecting latent topics in a corpus of short texts, which is achieved through the assistance of word embeddings, and has a good performance in order to scale up to a large dataset by using amortized variational inference with a neural network. However, this approach poses two major challenges:

- One must design a model that can appropriately use word vectors to their advantage.
- The original VAE approach only supports approximating the Gaussian distribution. Meanwhile, the topics generated for each document by LDA follow the Dirichlet distribution, and the relationships between each topic and each word are categorical.

To overcome these difficulties, we first posit a probability distribution, denoted $\boldsymbol{q}$, that factors into two conditional distributions: the first one is the topic distribution of each word conditioned on its embedding and current context; the second one is the topic distribution of each document based on all the topic assignments of all the words in that document. We then approximate this probability distribution using a neural network with a purposely designed architecture. The resulting network is a two-layered nested structure of latent variables, which we aptly name *nested variational autoencoder* (*N-VAE*). To find the parameter of the neural network, we minimize the Kullback–Leibler divergence between $\boldsymbol{q}$ and the posterior probability distribution of the LDA model by maximizing the corresponding evidence lower bound (ELBO) function, as normally done in the variational inference method. We use the Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2016) and the technique described by Figurnov et al. (2018), respectively, to optimize the parameters of the word-to-topic distribution (a categorical distribution) and the document-to-topic distribution (a Dirichlet distribution) in $\boldsymbol{q}$ using stochastic gradient descent.

Compared to other LDA-based approaches for topic modelling, our approach enjoys the following advantages.

- Since word vectors are encoded with rich semantic information, we can potentially generate more meaningful topics. Moreover, the word vectors can be further fine-tuned in the training process through backpropagation.
- Using a neural network to parameterize the approximator in the variational inference setting gives us a shorter convergence time (since neural networks allow concurrent processing) and smaller memory requirements (by using mini-batch gradient descent for training).

Indeed, experiments on various datasets of microtexts show that our model produces much better results on document clustering tasks and competitive results on topic coherence evaluation compared to other methods, while requiring a much smaller amount of time for convergence. Our implementation is available at https://github.com/trungtrinh44/N-VAE.

2 | RELATED WORKS

With the ability to infer latent topics from a collection of documents, topic modelling has many interesting applications in various fields: detecting themes in historical newspapers (Newman & Block, 2006; Yang, Torget, & Mihalcea, 2011), studying scholarly literature (Goldstone & Underwood, 2014; Mimno, 2012; Riddell 2014), and analysing biological data (Liu, Tang, Dong, Yao, & Zhou, 2016). In terms of social media, the topic model has been utilized for identifying influential users on Twitter (Weng, Lim, Jiang, & He, 2010), detecting communities (Tamimi, Lamrani, & Kamili, 2017), and comparing topics of interest between different regions (Z. Yin, Cao, Han, Zhai, & Huang, 2011). Even though popular methods such as LDA (Blei et al., 2003) have been successfully applied to collections of long texts, microtexts remain a huge challenge for these models. The sparsity of microtexts provides insu-cient information on word cooccurrences required to infer the topic-to-word and document-to-topic distributions. Therefore, various solutions have been proposed to overcome this problem.

One way to solve the lack of signals from microtexts is to aggregate them into a large document using some heuristics. This method is more commonly used for social media data, by leveraging the structure of the network and the relationship between the data. Hong and Davison (2010) explore the author-based aggregate scheme, that is, combining tweets of each user to form a document. Such aggregation is also used in Weng et al. (2010). Mehrotra, Sanner, Buntine, and Xie (2013), further introducing other aggregate schemes such as pooling tweets containing similar hashtags (Hashtag-based pooling), pooling tweets posted within the same hour to detect major events (temporal pooling), and so forth. These methods have been shown to improve the performance of LDA on tweet data, but they depend on the specific properties of the dataset, which means they cannot be generalized to other use cases of microtext-based topic modelling.

Another direction is to invent new topic models that are more suitable for the characteristics of microtexts. J. Yin and Wang (2014) study the effectiveness of the Dirichlet multinomial mixture model on microtext clustering. Yan, Guo, Lan, and Cheng (2013) introduce the biterm topic model whose generative process is specifically designed to adapt to the sparseness of microtexts. Quan, Kit, Ge, and Pan (2015) propose a model with a two-phase generative process, where the first phase is similar to the LDA model while the second phase assumes that each text snippet is derived from a hidden pseudo-document. While these models are more effective than LDA on microtexts, they can only use the limited information provided by the training corpus, which may not truly reflect the semantic relationships between words within the vocabulary.

To improve the results of topic models on microtexts, auxiliary information could be provided to augment or act as an alternative to the word cooccurrence statistics from the current corpus. Petterson, Buntine, Narayanamurthy, Caetano, and Smola (2010) use additional information on word similarities from thesauri and dictionaries to help place synonyms into the same topic. Song et al. (2011) employ a probabilistic knowledgebase to improve short text comprehension. O. Jin et al. (2011) develop an extension to the LDA model called dual LDA, where the target topics of a short text dataset are jointly learned with the supporting topics from an external collection of long texts. Other simpler approaches include training a topic model on a very large and universal corpus and then using the trained model to infer topics on a microtext corpus (Phan et al., 2011).

Word embedding Mikolov et al. (2013a, 2013b); Pennington et al. (2014) is a distributed representation of words and encodes the semantic relationships between them, which could be useful in aiding topic modelling. Some studies on using word embeddings in topic modelling have resulted in better performances on microtexts. Qiang, Chen, Wang, and Wu (2017) combine word embeddings with a text aggregation method and the Markov random field regularized model. Das et al. (2015); Nguyen et al. (2015) integrate LDA with word vectors via adding new components to the original model. However, these papers only explore Gibbs sampling as the training method, which is slow and cannot scale up to a large dataset, rendering them impractical for real use cases.

Recently, using word embedding for topic modelling problems has been attracting much attention. Since work embedding can encode word semantics based on the frequency of co-occurrence, related research spends much efforts on approximating word distribution to get the most accurate results. In Batmanghelich, Saeedi, Narasimhan, and Gershman (2016), the von Mises-Fisher (vMF) distribution was used to model word density. After that, Hierarchical Dirichlet process was used to approximate the probabilities needed for topic modelling. However, this method still relies on sampling to approximate hyper parameters, and so it will be difficult to scale-up with huge corpus. In Li, Wang, Zhang, Sun, and Ma (2016), auxiliary word embedding is used to enrich the encode semantics, enabling Dirichlet Multinomial Mixture model (DMM) to be used for the task of topic modelling. However, in this method, the categorical distribution assumed for the connection between document and words has not been addressed properly. Meanwhile, in Xun, Li, Zhao, Jing, and Aidong (2017), the proposed Correlated Topic model, instead of the Dirichlet-based approach, allows the use of embedded words in the distribution approximation process. However, this model still uses Gibbs sampling for distribution approximation, so suffering of computational cost once dealing with large corpus and still has not proved effective when dealing with short text. Hence, the word embedding method shows the ability to encode the semantic relationship

of words, thereby potentially supporting topic modelling on short text. However, this method still needs to be combined with a model that can approximate Dirichlet process and handle efficiently categorical relationship between words and topics, which is also the direction of our research.

The aim of the LDA technique is to form topics from textual data. Meanwhile, with the recent development of the deep learning techniques, the deep learning architecture based on *Recurrent Neural Network* (RNN) (Lu, Xie, Kang, Wang, & Xie, 2017) such as *Long Short-term Memory* (LSTM) (Jin, Luo, Zhu, et al., 2018) and *Gated Recurrent Units* (GRU) (Bansal, Belanger, & McCallum, 2016) are often used to process text data. Therefore, the combination of LDA and RNN was proposed naturally. In (Zhang, Li, & Wang, 2019), the authors propose a combination of LDA and LSTM to improve the performance of the topic modelling process. First, LSTM is used to create hidden representation of documents, then LDA is used to model topics from this hidden representation. Many other applications combination of LDA and LSTM are proposed for mortality prediction Jo, Lee, and Palaskar (2017), named entity detection on social media (Jansson & Liu, 2017), or developing recommendations from review of e-commerce.

However, those applications require that the input documents must be lengthy, in order to take advantage of the processing capability of LSTM. For short text, LSTM suffers from poor performance since it does not capture sufficient information. Perhaps the approach closest to our approach is (Lu et al., 2017), when RNN is combined with topic modelling. In this model, authors propose the BTM model to generate topics based on co-occurrence of word pairs. However, the author's method is still based on Gibb sampling, which consumes much memory resources since the whole corpus must be loaded for sampling. We overcome this problem by simulating LDA by VAE process, so that we can train faster and use less resources, enabling our method to be scalable.

With the recent advancement of the attention mechanism for deep learning, the application of attention-based RNN were considered in various areas when processing sequence data. Thus, the extension of RNN-based techniques with attention was also introduced for topic modelling in textual documents. In (Li, Zhang, Pan, Mao, & Yang, 2017), each document is treated as a sequence of text, enabling an attention-enhanced Bayes-based process to be performed for topic discovery. In (Mei, Bansal, & Walter, 2017), an attention-based language model approach was presented which enjoyed improvement on topic coherent. Notably, the simple RNN gains better performance in this work, as compared to other RNN-based improved models such as LSTM or GRU. However, in those approaches, in order to enforce the Dirichlet distributions among the words and topics in documents, those approaches employ the Gibbs sampling process, which is hardly scalable since it requires the whole corpus to be entirely loaded for sampling.

In addition, in the attention-based approach, a *context representation*, or *context*, has to firstly be formed, usually as a vector. Then, from the formed. Context, the corresponding *attention weights* are learned for every element in the analysed sequence. Nevertheless, such the context is sufficiently informative only if the sequence size is considerably long. Hence, the aforementioned approaches suffer difficulty once dealing with short text. In (Tian & Fang, 2019), an attention-based autoencoder approach was discussed to perform topic modelling on short text, which is relatively similar to our work. However, instead of performing reparameterizing to make the autoencoder learn the parameters of the Dirichlet distribution, this approach relies on the attention mechanism to improve the quality of the generated topics. The context and attention weights in this approach are inferred based on the *tf.idf* calculation from *Wordnet* in (Fellbaum, 2012). Thus, for on a specific domain, which is not fully covered by Wordnet, this approach may face problems.

Variational inference with the mean field approximator could be used to train the LDA model (Blei et al., 2003; Teh, Newman, & Welling, 2006), which is faster than Gibbs sampling. Kingma and Welling (2013); Rezende et al. (2014) replace the mean field approximator with the neural network, resulting in the VAE. Srivastava and Sutton (2017) present a VAE model for topic modelling based on LDA that has a much faster training and inference time while maintaining a competitive performance to the original LDA model trained using conventional methods. However, this model suffers from the component collapsing problem when training on microtexts.

The VAE-based approach for topic modelling is also introduced in Miao, Yu, and Blunsom (2016). This method uses neural variational inference for text processing, in which deep neural networks are used for the inference processes of probability, thereby handling common text processing problems such as document modelling or answer sentence selection. This method is based on probability and word distribution, making it difficult to process short text. A similar approach is presented in Nan, Ding, Nallapati, and Xiang (2019), in which the deep neural network is used to model the Dirichlet process in LDA. Work reported in Zhu, Feng, and Li (2018) introduces an approach that is similar to the clustering process, when an enhanced graph-based VAE is proposed to capture varied learned words from large corpus and apply it to topic modelling problem. In this approach, the BiTerm topic modelling model is adopted as a replacement of the familiar LDA-based topic modelling structure.

We take inspiration from those previous works and create a new model that employs a neural network architecture to approximate the posterior distribution of LDA and can benefit from word embeddings for knowledge of semantic relationships between words. We believe that using auxiliary information for assistance is the most general solution to tackle the problems of constructing a topic model on microtexts. This method not only works better on microtexts but could also enjoy improvements on long texts. We choose the stochastic variational inference with a neural network approximator because we want our model to scale up to large datasets without needing a long time to converge and consuming a large amount of memory.

3 | BACKGROUND

3.1 | Word embeddings

Word embeddings encode the information of each word using a dense vector. Two popular unsupervised methods for training word embeddings are Word2Vec Mikolov et al. (2013a, 2013b) and GloVe (Pennington et al., 2014). Word2Vec uses a shallow neural network consisting of an input layer, a projection layer and an output layer to predict neighbouring words. There are two versions of Word2Vec: continuous bag-of-words (CBOW) and skip-gram. The CBOW model attempts to predict a word based on its context, that is, its surrounding words, while the skip-gram model does the inverse, predicting context words based on the current word input.

Instead of learning the relationships between one word and its neighbours similar to Word2Vec, the GloVe method trains a log-bilinear regression model directly on the matrix of the global word cooccurrence statistics of a corpus. Its objective is to approximate the log probability of the cooccurrence of two words using the dot product of their word vectors.

Both Word2Vec and GloVe produce vectors that can encapsulate semantic relationships between words, which are usually encoded in the differences between two vectors. For example, vec(Vietnam) – vec(Hanoi) ≈ vec(France) – vec(Paris) ≈ vec(Germany) – vec(Berlin). Likewise, words that are near each other in the embedding space usually have certain similarities in their meanings.

3.2 | Latent Dirichlet allocation

LDA (Blei et al., 2003) is a probabilistic generative model popularly used to extract latent topics from a collection of documents. The model assumes that each document is a mixture of topics and each topic is a probability distribution over a fixed vocabulary. The generative process of LDA is described in Algorithm 1 and depicted in Figure 1, where:

- β_k represents the parameters of the categorical distribution over the vocabulary of topic k and K is the number of topics;
- θ_d represents the parameters of the categorical distribution over topics of document d and D is the number of documents;
- $z_{d,i}$ is the topic assignment for the i-th word of document d;
- $w_{d,i}$ is the i-th word of document d and N_d is the number of words in document d.
- α is the parameter of the Dirichlet prior distribution.

Subsequently, the joint probability distribution of LDA is defined as:

$$p(w,z,\theta|\alpha,\beta) = \prod_d p(\theta_d|\alpha)\prod_i p(z_{d,i}|\theta_d)p\left(w_{d,i}|\beta_{z_{d,i}}\right) \quad (1)$$

Algorithm 1

The generative process of LDA

for each document d **do**
  Sample a document-to-topic distribution $\theta_d \sim$ Dirichlet(α)
  for each word i in document d **do**
    Sample a topic $z_{d,i} \sim$ Categorical(θ_d)
    Sample a word $w_{d,i} \sim$ Categorical($\beta_{z_{d,i}}$)
  end for
end for

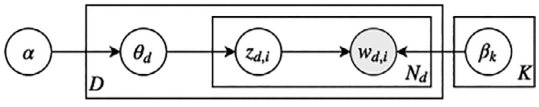


FIGURE 1 The latent Dirichlet allocation graphical model

3.3 | Variational inference

Probabilistic generative models such as LDA require Bayesian inference methods to induce the values of their latent variables from the corresponding posterior distribution. Recall that in Bayes' theorem, $p(z|x)$ can be evaluated as

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}. \tag{2}$$

However, such a distribution is usually intractable, leading to the employment of approximation methods. One such method is variational inference, where a tractable distribution q is used to approximate the true distribution p. To find q that most resembles p, we find q that minimizes the Kullback–Leibler (KL) divergence from q to p. Given a model with evidence x and a set of latent variables z, the KL-divergence from $q(z)$ to the posterior distribution $p(z|x)$ is defined as:

$$\begin{aligned} KL(q(z)||p(z|x)) &= -\mathbb{E}_{q(z)}\left[\log\frac{p(z|x)}{q(z)}\right] \\ &= \mathbb{E}_{q(z)}[\log q(z) - \log p(z|x)] \end{aligned} \tag{3}$$

Since directly minimizing this function requires knowing how to calculate $p(z|x)$, which is intractable in the first place, we instead maximize the ELBO function:

$$\begin{aligned} \text{ELBO} &= \mathbb{E}_{q(z)}[\log p(z,x) - \log q(z)] \\ &= \mathbb{E}_{q(z)}[\log p(x) + \log p(z|x) - \log q(z)] \\ &= \log p(x) - \mathbb{E}_{q(z)}[\log q(z) - \log p(z|x)] \\ &= \log p(x) - KL(q(z)||p(z|x)) \end{aligned} \tag{4}$$

The name ELBO stems from the fact that the term is always smaller or equal to the log probability of the evidence, that is, $\log p(x) \geq \text{ELBO}$ since $KL(q(z)||p(z|x)) \geq 0$. Because $\log p(x)$ is a constant with regard to the observed data x and the parameters of q, maximizing the ELBO corresponds to minimizing the $KL(q(z)||p(z|x))$.

The chosen approximate posterior q usually comes from the mean field family, where each latent variable is assumed to come from a distribution with its own parameters. The optimization process requires deriving the updating rules by hand analytically for the coordinate descent algorithm. The analytical solution only exists if the model is conjugate. LDA is one such model because of the conjugacy between the Dirichlet and the multinomial distributions. Hence, the largest limitation of mean field variational inference is that it can only be used with conjugate models.

3.4 | Variational autoencoder and reparameterization tricks

One way to mitigate the limitation of mean field variational inference is to use a neural network to parameterize the approximate posterior q. This idea is explored in Kingma and Welling (2013); Rezende et al. (2014), which gives rise to a new class of models called the VAE. Essentially, the VAE consists of two parts: the *encoder* $q_\phi(z|x)$ and the *decoder* $p_\theta(x|z)$, where ϕ and θ denote the parameters of the encoder and decoder respectively. The encoder role is to map each of the input x to its corresponding latent variables z, and the decoder role is to reconstruct x from z. Ideally, we want the encoder $q_\phi(z|x)$ to act as if it is the true posterior $p_\theta(z|x)$ as much as possible. That goal is achieved by minimizing the KL-divergence between the two distributions, similar to the variational inference method. Under this setting, Equation (3) is rewritten as:

$$KL\left(q_\phi(z|x)\middle||p_\theta(z|x)\right) = \mathbb{E}_{q_\phi(z|x)}\left[\log q_\phi(z|x) - \log p_\theta(z|x)\right] \tag{5}$$

and the new ELBO is:

$$\begin{aligned} \text{ELBO} &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(z,x) - \log q_\phi(z|x)\right] \\ &= -KL\left(q_\phi(z|x)\middle||p_\theta(z)\right) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \end{aligned} \tag{6}$$

Figure 2 depicts a VAE whose latent variables z are assumed to be generated from a Gaussian distribution.

The KL term in Equation (6) usually has a closed-form expression. However, samplings need to be done in order to approximate the expectation term. There is a direct formula to calculate the gradients of the expectation with respect to the variational parameters, but this method yields gradients with very high variance, making training impossible. Therefore, both papers introduce an alternative called the RT. This method uses a

differentiable and invertible function g_ϕ such that $\varepsilon = g_\phi(z)$ and $z = g_\phi^{-1}(\varepsilon)$. Here, $\varepsilon \sim p(\varepsilon)$ is called a noise variable, and this transformation helps remove the dependence of the sampling process on the variational parameters. Instead of sampling directly from $q_\phi(z|x)$, we now draw ε from its distribution and calculate the corresponding z. If $q_\phi(z|x)$ happens to be a Gaussian distribution $\mathcal{N}(\mu,\sigma)$, we can choose $p(\varepsilon) = \mathcal{N}(0,1)$ and $g_\phi(z) = (z - \mu)/\sigma$, that is, $z = \mu + \sigma\varepsilon$. Unfortunately, finding a pair of $p(\varepsilon)$ and a function g_ϕ is not a trivial task for other distributions, such as the Dirichlet or categorical distributions used in LDA. One universal solution for every continuous distribution is to consider $p(\varepsilon)$ = Uniform(0, 1) and $g_\phi(z) = F_\phi(z|x)$ where $F_\phi(z|x)$ is the cumulative distribution function of $q_\phi(z|x)$; however, to calculate z, we need to find the inverse of $F_\phi(z|x)$, which could be a very complicated process. The work in Figurnov et al. (2018) proposes a novel way to alleviate this problem called implicit reparameterization, which in contrast to the method discussed so far does not need to find the inverse of g_ϕ. This solution enables the RT to be used with a variety of continuous distributions, including the Dirichlet distribution. For discrete cases such as the categorical distribution, we use the Gumbel-Softmax (Jang et al., 2016; Maddison et al., 2016) as a continuous approximation, which also permits backpropagation to update the distribution parameters.

4 | A RUNNING EXAMPLE

We illustrate our method advantage as the following example. Suppose we have a set of documents as depicted in Figure 3. We may observe that the keywords in the documents can be categorized into two topics about *car* and *animal*. Typically, LDA can detect these topics based on the

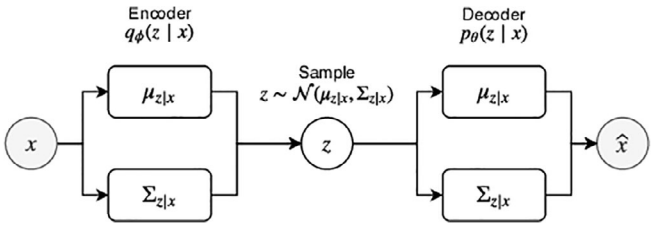


FIGURE 2 An example variational autoencoder with a Gaussian posterior and a Gaussian decoder

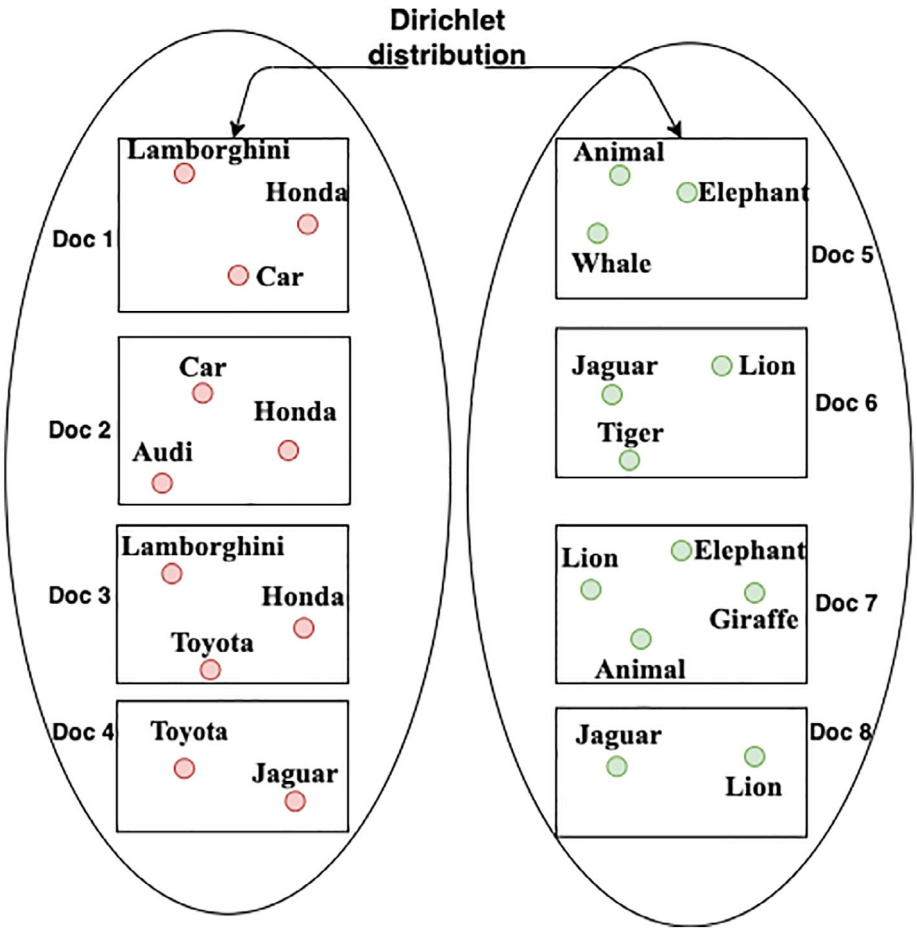


FIGURE 3 A set of short documents with keywords scattered

distribution of keywords in documents. However, if these keywords appear scattered in many short documents as illustrated, LDA suffers from difficulty because it does not have enough statistical information about the keywords distribution in each document.

By encoding words into word vectors based on their co-occurrence relationship, one can turn keywords into numerical vectors so that the words frequently appear together in documents will be transformed into similar vectors, as illustrated in Figure 4. By estimating the *categorical distribution* among these vectors and documents, we can form topics of word vectors as illustrated in Figure 5. These topics can be represented by *topic vectors*.

Thus, words that are semantically related will have the corresponding embedding vectors placed close together in the Euclidean space. If we give each topic a separate representative vector, then this vector will be close to the vectors of words belonging to that topic. Conversely, a word will also have its vector placed close to the topic vector to which it belongs. With this observation, we can infer the probability distribution of a word belonging to a topic by inspecting the distance between word vectors and topic vectors, as illustrated in Figure 6.

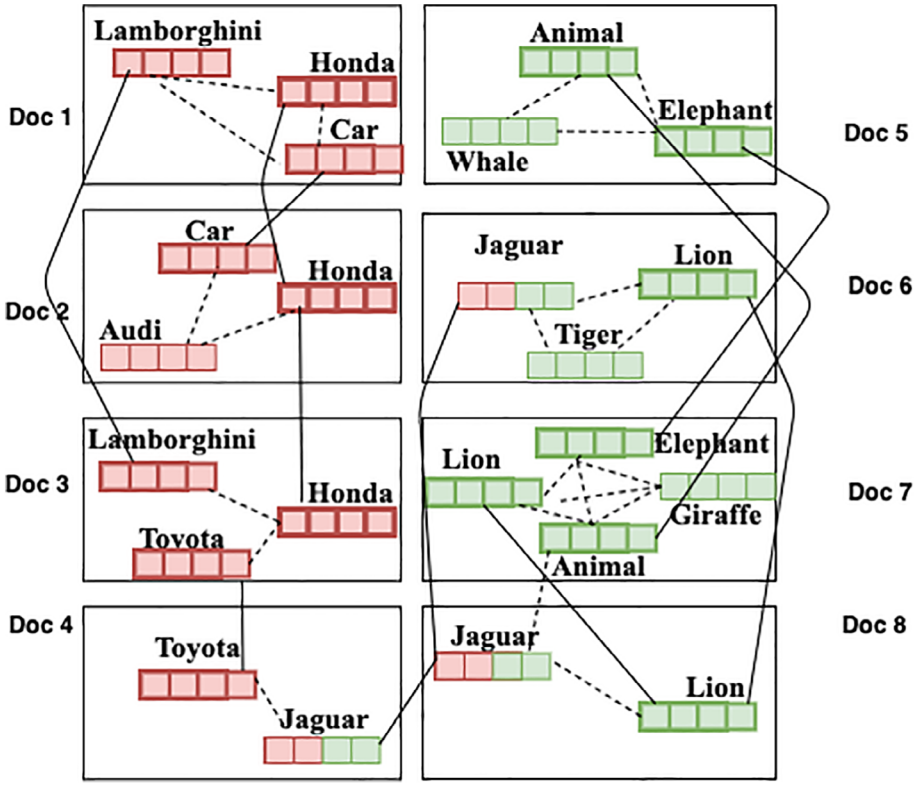


FIGURE 4 Words vectors and their relationship when generated from the document set

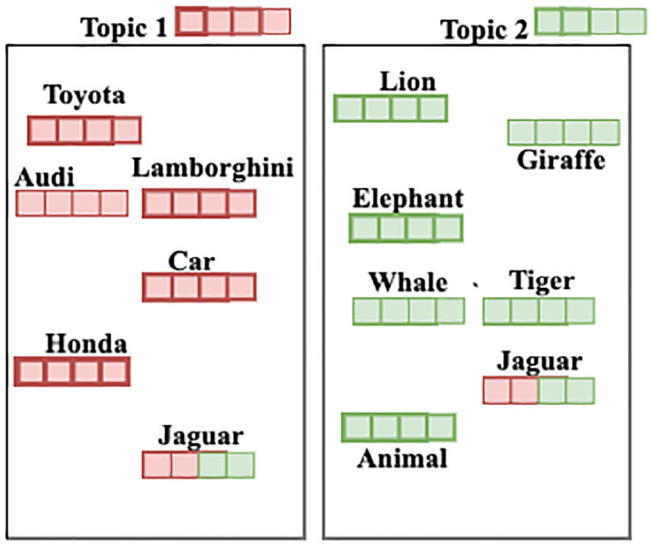
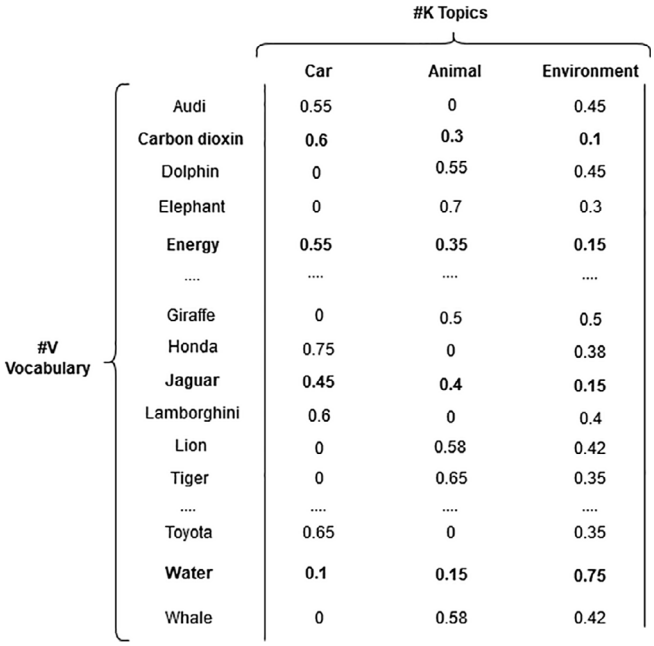


FIGURE 5 Topics and topic vectors generated from the word vectors and documents

FIGURE 6 The distance between word vectors and topic vectors



Based on the relationship between word vectors and topic vectors, one can estimate the distribution of words over topics. Meanwhile, the word vectors can also help us to estimate distribution of words in documents. Based on those distribution, we can identify topics in documents based on the *Dirichlet distribution*. For example, we can identify that Document 4 are of Topic 1 and Document 8 of Topic 2.

Thus, to realize this idea, we will come up with the proposed *Nested VAE* (N-VAE) architecture as discussed in the following section, in which VAE will be implemented twice in a nested manner to handle the following tasks.

- Approximating categorical distribution among words in topics.
- Approximating Dirichlet distribution among topics in documents.

Since the original VAE is only designed for approximating Gaussian distribution, we use *reparameterization* tricks to make VAE enable to approximate the categorical distribution and the Dirichlet distribution, as presented in the following section.

5 | NESTED VARIATIONAL AUTOENCODER (*N-VAE*) FOR TOPIC MODELLING ON MICROTEXT WITH WORD VECTORS

In this section, we introduce *N-VAE*, a novel VAE architecture for topic modelling with word embeddings. Our approach differs from other works on combining word embeddings and LDA since instead of replacing or extending components of the original LDA model with elements that are compatible with word vectors (Das et al., 2015; Nguyen et al., 2015), we propose a variational approximation $\boldsymbol{q}$ to the posterior of the LDA model that considers word embeddings as one of its parameters. In the following subsections, we will introduce the formulation of $\boldsymbol{q}$, the derivation of the ELBO as the objective function and the translation of $\boldsymbol{q}$ to a neural network architecture. We will denote V, K, D as the vocabulary size, the number of topics and the word embedding size, respectively.

5.1 | The proposed distribution q

To design the variational distribution $\boldsymbol{q}$, we make two assumptions:

- A coherent topic should be a group of words whose vectors are close to each other in the embedding space.
- From the embedding and the context of a word, we could decide which topic that the word belongs to.

To realize both assumptions, we parameterize the word-to-topic distribution – a categorical distribution – using word embeddings and a set of vector representations of the topic (one vector for each topic). More specifically, for word i, topic t and document d, the log probability of the

unnormalized word-to-topic distribution is a sum of two factors: the first one, denoted $s_{d,i}^t$, is the dot product between the word embedding ω_i and the topic embedding ρ_t; the second one, denoted c_d^t, is a function g_t that receives the context representation of that word in document d and outputs the coefficient of topic t:

$$\begin{aligned} s_{d,i}^t &= \omega_i^d \cdot \rho_t \\ c_d^t &= g_t\left(\overline{\omega^d}\right) \\ \log\tilde{q}\left(z_{d,i}=t|\rho_t,\omega^d\right) &= \pi_{d,i}^t = s_{d,i}^t + c_d^t \\ q\left(z_{d,i}=t|\rho_t,\omega^d\right) &= \frac{\exp\left(\pi_{d,i}^t\right)}{\sum_t \exp\left(\pi_{d,i}^t\right)} \end{aligned} \tag{7}$$

where $z_{d,i}$ is the topic assigned to word i in document d, and ω^d is the matrix of all the word embeddings within document d, $\overline{\omega^d}$ is the mean of all the word vectors in document d. Here, we represent the context of a word in a document as a the mean of word vectors in that document since we want the representation to be agnostic to the document's length. This strategy works well in practice.

The intuition behind this formulation is as follows:

- For $s_{d,i}^t$, since a topic should be a group of neighbouring word embeddings, the dot product between each word vector in the topic and the topic vector should be large. Therefore, this term will put a high probability on a topic whose vector is near the vector of the current word. This property is illustrated in Figure 7.
- To account for homonyms, we introduce the term c_d^t, since these words have different meanings in different contexts and thus will belong to different topics. For example, the word *Jaguar* in Figure 7 could indicate an animal or a car manufacturer depending on the current context.

In practice, we use the Gumbel-Softmax estimator to approximate the categorical distribution. Therefore, Equation (7) is rewritten as:

$$q\left(z_{d,i}=t|\rho_t,\omega^d\right) = \frac{\exp\left(\pi_{d,i}^t/\tau\right)}{\sum_t \exp\left(\pi_{d,i}^t/\tau\right)} = \mu_{d,i}^t \tag{8}$$

where τ is called the temperature. This term controls how much the continuous approximation resembles the real categorical distribution, with τ approaching 0 resulting in a more discrete-like sample. When training, we anneal the temperature from 1 to a predefined minimum value.

With each word in a document now assigned a topic based on the distribution defined in Equation (7), we calculate the topic proportion of document d as $q(\theta_d|z_d)$ = Dirichlet(ν_d), where ν_d is defined as:

$$\nu_d = \texttt{softplus}(\eta_d * a + b) \tag{9}$$

where

$$\eta_d = \left[\eta_{d,0}, \eta_{d,1}, \ldots, \eta_{d,K-1}\right] \tag{10}$$

is a K-dimension vector and $\eta_{d,i}$ is the number of words in document d assigned the i-th topic, a and b are scalars that map η_d to the appropriate values for the parameters of the Dirichlet distribution, and $\texttt{softplus} = \log(1+\exp(x))$. The use of `softplus` is to make sure that ν_d is always positive, a requirement for Dirichlet parameters.

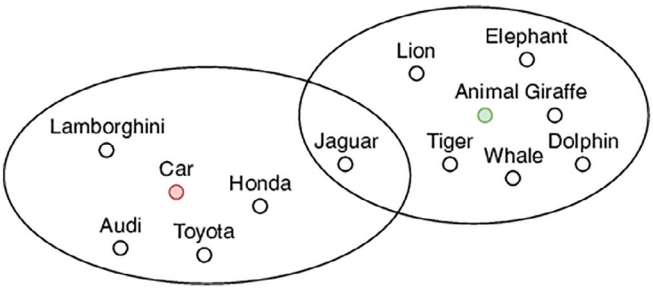


FIGURE 7 An example of word embeddings with two topic vectors: Car and Animal. Here, the word Jaguar is a homonym which could mean an animal or a name of a car manufacturer

Overall, the variational posterior $\boldsymbol{q}$ is defined as:

$$\begin{aligned} q(\theta, z|\rho, \omega) &= \prod_d q(\theta_d|z_d) q\left(z_d|\rho, \omega^d\right) \\ &= \prod_d q(\theta_d|z_d) \prod_i q\left(z_{d,i} = t|\rho_t, \omega^d\right) \end{aligned} \tag{11}$$

5.2 | Variational objective

With the definition of the variational posterior in Equation (11), we can write the variational objective function. Since all the documents in a corpus are generated independently of each other, we derive the objective function for one document and the sum of all such functions as the final objective function of the corpus. The ELBO of document d is defined as:

$$\begin{aligned} \mathscr{L}_d &= \mathbb{E}_{q(\theta_d, z_d|\rho, \omega^d)}\left[\log p(w_d, \theta_d, z_d)|\alpha, \beta) - \log q\left(\theta_d, z_d|\rho, \omega^d\right)\right] \\ &= \mathbb{E}_{q(\theta_d, z_d|\rho, \omega^d)}\left[\log p(\theta_d|\alpha) + \log p(z_d|\theta_d) + \log p(w_d|z_d, \beta) - \log q(\theta_d|z_d) - \log q\left(z_d|\rho, \omega^d\right)\right] \\ &= \mathbb{E}_{q(z_d|\rho, \omega^d)}\left[-\log q\left(z_d|\rho, \omega^d\right) - KL_{Dir}(q(\theta_d|z_d)||p(\theta_d|\alpha)) + \log p(w_d|z_d, \beta) + \mathbb{E}_{q(\theta_d|z_d)}[\log p(z_d|\theta_d)]\right] \end{aligned} \tag{12}$$

Combining Equations (8), (10) and (12) we have:

$$\mathscr{L}_d = -\sum_i w_{d,i}\left(\sum_t \mu_{d,i}^t \log \mu_{d,i}^t\right) + \mathbb{E}_{q(z_d|\rho, \omega^d)}\left[-KL_{Dir}(q(\theta_d|z_d)||p(\theta_d|\alpha)) + \sum_i w_{d,i} \log \beta_i^{z_{d,j}} + \mathbb{E}_{q(\theta_d|z_d)}\left[\sum_t \eta_{d,t} \log \theta_{d,t}\right]\right] \tag{13}$$

where:

- Each $w_{d,i}$ indicates the count of word i in document d.
- Each $z_{d,i}$ denotes the topic assignment of word i in document d and $z_d = [z_{d,0}, z_{d,1}, \ldots, z_{d,V-1}]$.
- $\omega^d \in \mathbb{R}^{V \times D}$ is the matrix of all word embeddings of document d.
- $\rho \in \mathbb{R}^{K \times D}$ is the matrix of all the topic vectors.
- α represents the parameters of the Dirichlet prior in the LDA model.
- $\beta \in \mathbb{R}^{K \times V}$ is the matrix containing the parameters of all the topic-to-word distributions.
- Each β_i^t is the probability that topic t generates word i.
- KL_{Dir} is the KL-divergence between two Dirichlet distributions, which can be calculated analytically.

Here, we treat each document d as a bag-of-words following the standard practice in training LDA using variational inference.

To optimize Equation (13), we have to use the RT twice: one for the word-to-topic distribution $q(z_d|\rho, \omega^d)$ and one for the document-to-topic distribution $q(\theta_d|z_d)$. For the word-to-topic distribution, a categorical distribution, we use the RT introduced in Jang et al. (2016); Maddison et al. (2016), and for the document-to-topic distribution, a Dirichlet distribution, we use the RT introduced in Figurnov et al. (2018).

5.3 | Neural network architecture

From the equations presented in Section 5.1, it is straightforward to translate the variational distribution to a neural network representation. An overview of the architecture is illustrated in Figure 8, and the direct mapping between the formula and the architecture is depicted in Figure 9 by t-Distributed Stochastic Neighbour Embedding. As seen in the figure, our model receives the embeddings of words within a document as its inputs. In practice, these embeddings are parts of the model's parameters, which gives us the ability to further fine-tune the word embeddings to suit the dataset, should we choose to do so.

Similar to Srivastava and Sutton (2017), wby t-Distributed Stochastic Neighbour Embedding e found the usage of batch normalization (BN) to be essential to achieving the desire performance of the model, as training without BN frequently leads to *component collapsing*. This phenomenon happens because early in the training process, the KL-divergence between two Dirichlet distributions dominates the loss term, which leads to the model converging to a local minimum where most of the components in the posterior are inactive; hence, the resulting topics are all similar to each other. Moreover, in addition to adding BN between fully connected layers as in Figure 9, we also utilize BN in the calculation of the β term in Equation (13):

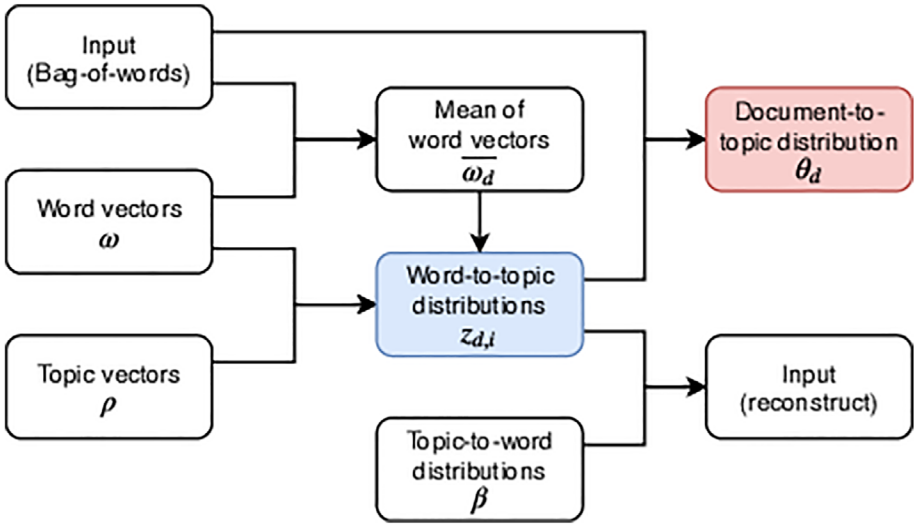


FIGURE 8 An overview of the neural network architecture of the *nested variational autoencoder*

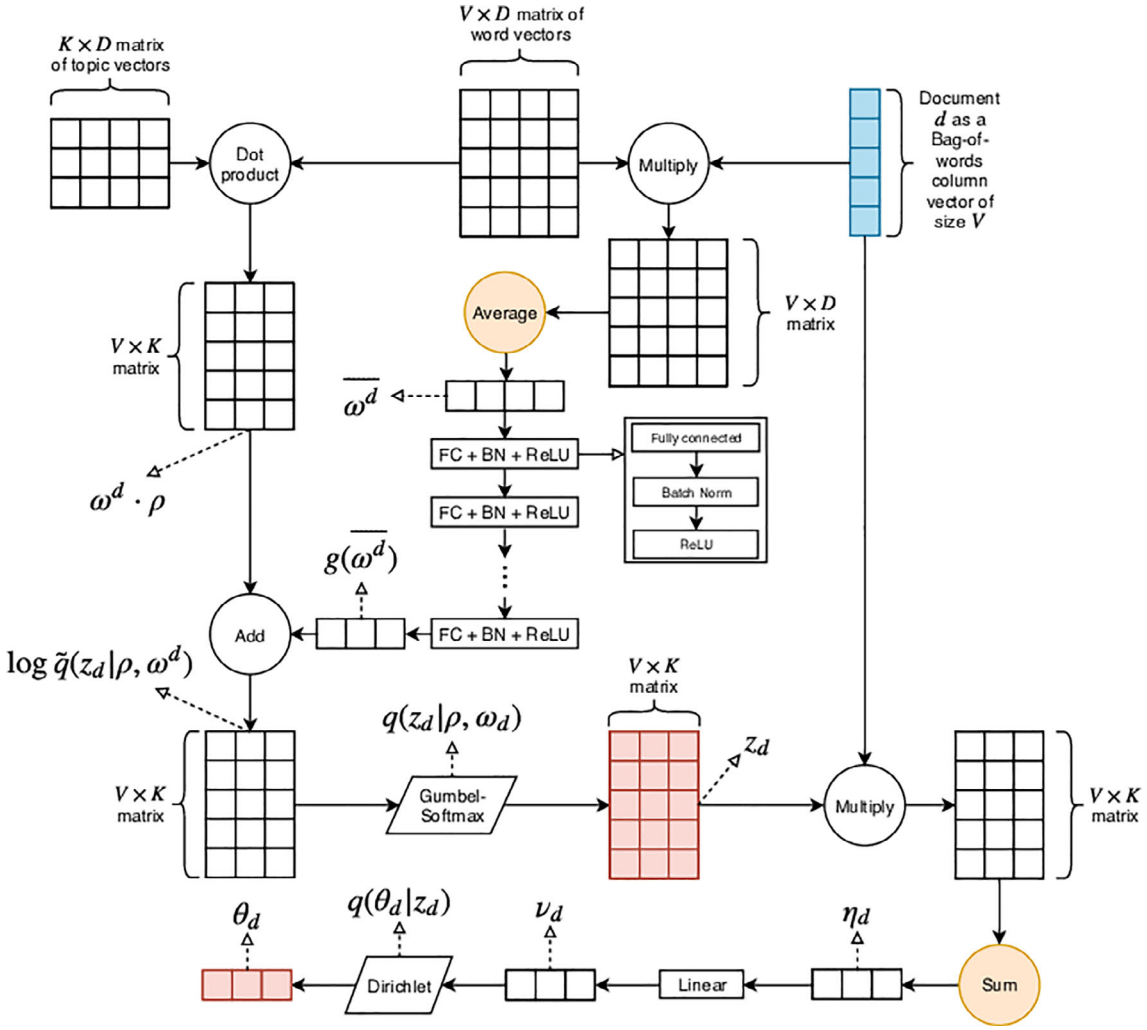


FIGURE 9 The neural network representation of the posterior distribution q(θ,z|p,ω). Each dotted line originated from a vector and points to the corresponding factor in the formulas introduced in Section 5.1. Each circle depicts a simple operation, and those with an orange background are operations performed on elements across the first dimension of a matrix. Red vectors are samples drawn from a distribution. ReLU is the function f(x) = max(0,x). Linear is the function f(x) = axe+b where a and b are scalars. A parallelogram depicts a distribution from which we draw samples. K is the number of topics, D is the size of the word embeddings and V is the vocabulary size. Here, K = 3, D = 4 and V = 5. Blue vectors indicate the input of the model

$$\beta = \texttt{Softmax}(\texttt{Transpose}(\texttt{Batch}-\texttt{Norm}(\texttt{Transpose}(\tilde{\beta})))) \quad (14)$$

where $\beta \in \mathbb{R}^{K \times V}$. Here, we apply BN so that for each topic, the non-normalized probability $\tilde{\beta}_t$ will have a stable mean and variance, which we found necessary for convergence. We hypothesize that such stability provided by BN helps all the topic-to-word distributions to learn at the same rate, that is, to have gradients with similar magnitudes, which makes the model converge to a more favourable local minimum and avoid component collapsing. Indeed, when measuring the gradients of the topic-to-word distributions, we found the model without BN at the β matrix experiencing a divergence – where dominating topics have larger gradients, while other topics' gradients approach zero – and the model with BN having its corresponding gradients stay close to each other. A detailed study will be included in Section 6.2.

6 | EXPERIMENTS AND RESULTS

In this section, we evaluate the *N-VAE* by measuring its topic coherence and its performance on the document clustering task. Topic coherence indicates how related the words that are assigned to each topic are, which closely resembles how humans evaluate a topic model. The document clustering task directly compares the clustering results to the ground truth labels by considering each class as a cluster. To demonstrate the advantages of our model, we compare its performance and runtime against the original LDA model and other topic models that utilize word embeddings. To study the significance of word embeddings with respect to the topic model's quality, we compare the results on different sets of pretrained word embeddings. Since our focus is on performing a topic model on a corpus of microtexts, we conduct our experiments mostly on datasets with the average text length smaller than 20 words. We also present the hyperparameters and the training process required for the *N-VAE* to reach its optimal performance. Finally, we present a detailed study regarding the importance of BN to the model's performance. Note that for each experiment in the subsequent sections, we ran it 10 times and reported the average score.

6.1 | Experimental setup

6.1.1 | Pretrained word embeddings

We use two state-of-the-art sets of pretrained word embeddings in our experiments:

- Google word vectors[1]: these are 300-dimensional embeddings trained on a subset of the Google News corpus that contains approximately 100 billions words using the Word2Vec framework Mikolov et al. (2013a, 2013b). We denote this set of vectors as **w2v**.
- Stanford word vectors[2]: these are 300-dimensional embeddings trained on the Common Crawl dataset that contains 42 billions tokens using the GloVe method (Pennington et al., 2014). We denote this set of vectors as **glove**.

We have experienced N-VAE with both methods of **glove** and **word2vec**. However, the experimental result described in Table 3 shows that embedding **glove** method returns distinctly better result than that of **word2vec**. This result leads us to a conclusion that with short text, **word2vec** has not captured sufficient information about words co-occurrence in the same document, so that it could not achieve high accuracy. In the meantime, the **glove** method is trained with wide information of the corpus, which makes it has more contextual information than **word2vec**. Therefore, we only use the **glove** method for all baseline method described in the following section.[3]

6.1.2 | Datasets

We use four datasets: the 20-Newsgroups dataset, the TagMyNews dataset (Vitale, Ferragina, & Scaiella, 2012), the Sander Twitter corpus and the Web Snippets dataset (Phan et al., 2011) to evaluate our model.

For the 20-Newsgroups and TagMyNews datasets, we use the preprocessed version and their derivation provided by Nguyen et al. (2015). These include the full version of the 20-Newsgroups dataset and the TagMyNews dataset, denoted **N20** and **TMN** respectively; the **N20short** dataset contains all the documents from the N20 dataset having less than 21 words; the **N20small** dataset is balanced and contains 400 randomly selected documents from the original N20 dataset; and the **TMNtitle** dataset consists of only news titles from the TMN dataset.

The topic classification results can be illustrated in Figure 10. We conducted the experiment on the TagMyNews dataset as a news data set, in which each news included title and short description, divided into seven categories of *business*, *health*, *sci-tech*, *sport*, *the US*, *world*, *entertainment*. Figure 10 visualizes topics modelled by *t-Distributed Stochastic Neighbour Embedding (t-SNE)* (Maaten & Hinton, 2008). This visualization demonstrates two significant characteristics which should be carefully taken into account once considering topic modelling results.

When the topic is modelled, there should be of *coherence*. That is, the words with thigh scores in the topic should be related to each other, and this case should be highly relevant to the predefined categories they should belong to.

Once treated as clusters, the modelled topics should demonstrate the high quality of the generated clusters, which means different clusters should be well distinguished to each other.

For the Sander Twitter corpus,[4] we download the 5,513 tweets using their Tweet IDs. There are 400 non-downloadable tweets. We closely follow the preprocessing method for this dataset as presented in Nguyen et al. (2015). After the preprocessing, there are 2,546 remaining tweets.

For the Web Snippets[5] dataset, we remove stop words using the list of stop words from the Stanford CoreNLP,[6] as well as any words that are not contained in the Stanford and Google pretrained vectors. We also eliminate words that appear less than three times in the corpus. Finally, we remove any document whose length is zero after the preprocessing.

Table 1 summarizes the statistics of all the datasets.

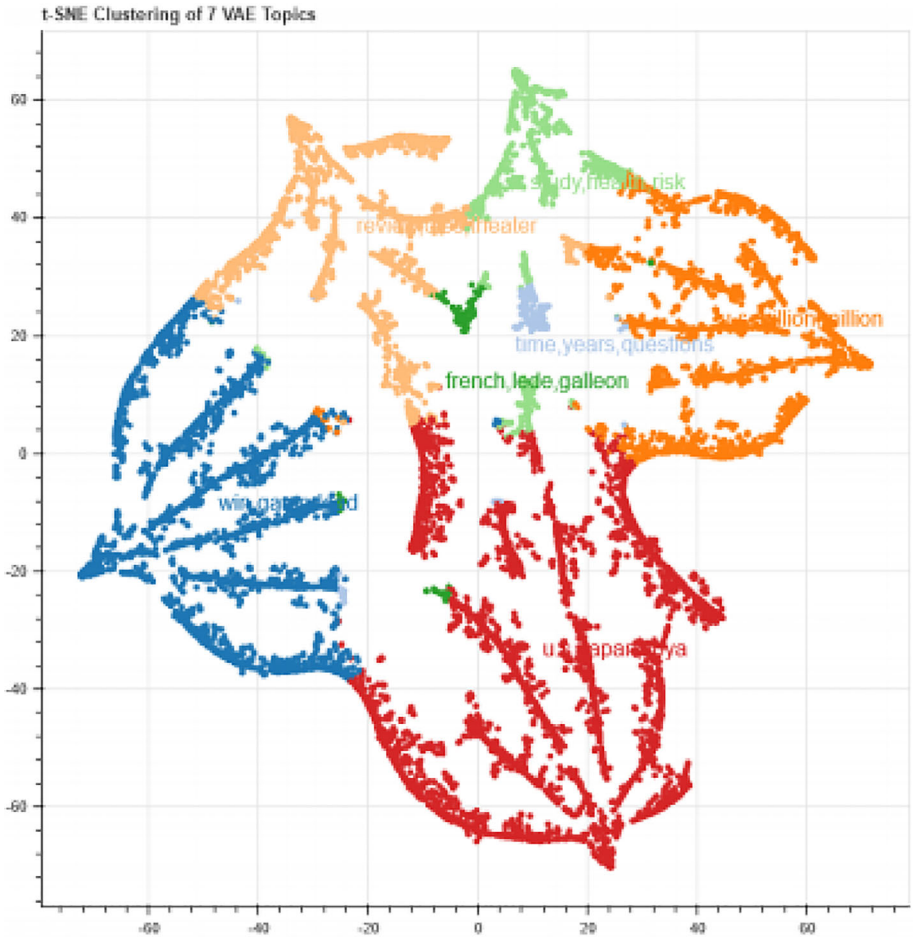


FIGURE 10 t-SNE Clustering of seven variational autoencoder topics

TABLE 1 Statistics of the datasets

Dataset	#g	#d	#w/d	V
N20	20	18,820	103.3	19,572
N20short	20	1,794	13.6	6,377
N20small	20	400	88	8,157
TMN	7	32,597	18.3	13,428
TMNtitle	7	32,503	4.9	6,347
Web snippets	8	12,335	14.5	7,314
Twitter	4	2,546	4.9	1,402

Note: #g: number of ground truth labels. #d: number of documents. #w/d: the average length of a document. V: the size of the vocabulary.

TABLE 2 The hyperparameters for each dataset

Dataset	#epochs	BS	#burn-in epochs	Min τ	Layers	Train word embedding
N20	64	256	32	0.5	128, 128	Yes
N20short	256	256	128	0.7	128, 128	No
N20small	256	200	128	0.7	128	No
TMN	128	256	64	0.7	128, 128	No
TMNtitle	128	256	64	0.7	128, 128	No
Web snippets	64	256	32	0.7	128, 128	No
Twitter	128	256	64	0.7	128, 128	No

Note: #epochs: number of training epochs. BS: batch size. #burn-in epochs: number of epochs before training α. Min τ: minimum temperature.

6.1.3 | Training process and hyperparameters

To train the *N-VAE*, we use the Adam Optimizer Kingma and Ba (2014) with $\beta_1 = 0$ and $\beta_2 = 0.99$. We use a low momentum value since it allows the topic placement of a word to change quickly during the training process. From our experiments, we are able to confirm that using a low momentum allows our model to reach a lower perplexity as well as improve both topic coherence scores and document clustering results. We set the learning rate to $8e-3$ since it allows our model to converge quickly.

When training, we slowly increase the learning rate from 0 to $8e-3$ while decreasing the temperature τ of the Gumbel-Softmax from 1 to a minimum value – which is 0.7 for small datasets or datasets of microtexts and 0.5 for large datasets of long texts. This process takes place in the first epoch, which acts as a warm-up period for the model.

We set the initial value of the Dirichlet prior α to 0.1, and optimize this value after a certain number of epochs, which is usually one-half the total amount of epochs. We find this greatly improves our model performance, especially on the document clustering task, which is not surprising since the study in Wallach et al. (2009), confirmed the importance of the Dirichlet prior to the quality of the topic model.

We use a batch size of 200 for the N20small dataset and of 256 for other datasets. We choose the number of epochs for each dataset that guarantees the convergence of our model. We use one fully connected layer of 128 units for the N20small dataset and two fully connected layers of 128 units for other datasets.

On large datasets of long texts, such as the N20 dataset, we also allow the training of word embeddings, which resulted in an additional boost in the model performance.

Table 2 presents the hyperparameters for each dataset.

6.2 | Effect of batch normalization

To provide a deeper understanding of the training process of our model, we investigate the effect of BN on the *N-VAE*'s performance. In the model's architecture, we use BN at two components: between the fully connected (FC) layers and at the β matrix. We carry out the experiment on the N20short dataset with six topics, where we alternately add and remove the BN at each component such that there are ultimately four possible combinations. We plot the value of the Dirichlet parameters α, the gradients of the topic-to-word distribution β and the gradients of the weights of the last FC layer for each experiment.

Figures 11–14 depict the graphs of each combination of the four experiments. From the graph, we can observe that the removal of BN at the β matrix leads to the divergence of the gradients between topics, where topics are divided into two groups: one group enjoys a high gradient while the other group obtains an increasingly smaller gradient. Therefore, the effect of BN at the β matrix is to maintain a similar learning speed between topics in order to prevent the component collapsing phenomenon from happening. We can also witness the component collapsing phenomenon by observing the α: its values diverge in Figures 11 and 13 and remain similar in Figures 12 and 14.

On the other hand, the BN between the FC layers has the effect of allowing a larger gradient update at the weights of these layers. Figures 11 and 12 show that without the BN at these layers, their gradient becomes extremely small, which means that the model fails to learn anything at all. By contrast, the gradients in Figures 13 and 14 have a much larger magnitude throughout the training process.

We hypothesize that these phenomena occur due to the saturation of the softmax function during the training period. At the β matrix, we use softmax to transform the original weights into a legitimate probability distribution. The training process may leads to some topics unfortunately having their softmax at the β matrix saturated quickly, resulting in a diminishing gradient. The same event could happen with

TABLE 3 NPMI scores (higher is better) for the N20, N20small, N20short, TMN, TMNtitle, Twitter and Web Snippets dataset

Dataset	Number of topics	N-VAE word2vec	N-VAE glove	GPU-DMM	NVI	Was-A	AVITM	LDA	AATM	RATM	ARMN
N20	6	0.513	0.519	0.517	0.513	**0.533**	0.529	0.516	0.377	0.353	0.253
	20	0.62	0.632	0.61	0.62	**0.65**	0.63	0.582	0.452	0.460	0.361
	40	0.617	0.624	0.647	**0.667**	0.607	**0.667**	0.557	0.486	0.522	0.367
	80	0.6	0.613	0.59	0.6	0.63	**0.64**	0.515	0.585	0.582	0.890
N20small	6	0.403	0.437	0.438	0.439	**0.443**	0.423	0.376	0.362	0.351	0.063
	20	0.492	0.353	0.482	**0.542**	0.502	0.492	0.474	0.382	0.382	0.067
	40	0.514	0.56	0.534	**0.564**	0.534	0.514	0.513	0.494	0.491	0.094
	80	0.554	0.585	0.574	0.584	0.574	**0.604**	0.563	0.454	0.596	0.099
N20short	6	0.252	0.301	0.292	0.262	**0.302**	0.282	0.224	0.448	0.232	0.202
	20	0.302	**0.353**	0.342	0.312	0.292	0.322	0.248	0.198	0.242	0.212
	40	0.311	**0.375**	0.341	0.331	0.331	0.341	0.268	0.285	0.245	0.231
	80	0.367	0.367	**0.407**	0.357	0.397	0.377	0.308	0.199	0.297	0.297
TMN	7	0.467	**0.487**	0.467	0.477	0.477	0.457	0.429	0.327	0.237	0.337
	20	0.457	0.467	**0.507**	0.487	0.477	0.497	0.4	0.367	0.470	0.337
	40	0.452	**0.464**	0.442	0.442	0.442	0.442	0.354	0.372	0.372	0.382
	80	0.447	**0.474**	0.437	0.457	0.447	0.467	0.319	0.383	0.387	0.397
TMNtitle	7	0.367	0.4	**0.407**	0.357	0.387	0.357	0.317	0.333	0.247	0.347
	20	0.361	**0.395**	0.361	0.381	0.391	0.361	0.269	0.268	0.251	0.171
	40	0.362	0.393	**0.412**	0.402	0.372	0.372	0.227	0.279	0.182	0.192
	80	0.341	**0.372**	0.371	0.381	0.391	0.341	0.196	0.278	0.191	0.171
Twitter	4	0.274	**0.324**	0.304	0.264	0.304	0.262	0.234	0.197	0.184	0.114
	20	0.251	0.28	**0.301**	0.261	0.281	0.249	0.178	0.143	0.158	0.121
	40	0.276	0.291	**0.306**	0.266	0.296	0.266	0.162	0.164	0.156	0.136
	80	0.267	0.277	**0.287**	0.244	0.257	0.267	0.156	0.172	0.137	0.144
Snippet	8	0.63	0.628	0.64	**0.67**	0.62	0.63	0.49	0.178	0.294	0.372
	20	0.597	0.59	**0.647**	0.597	0.637	0.597	0.424	0.184	0.354	0.387
	40	0.574	0.573	**0.604**	0.584	0.584	0.574	0.364	0.194	0.297	0.288
	80	0.579	**0.629**	0.589	0.579	0.58	0.599	0.336	0.194	0.299	0.292

Note: Bold entries imply the best result.

the Gumbel-Softmax distribution, where the saturation of the softmax means that the FC layers have very small gradients during backpropagation. Both of these catastrophic phenomena could be remedied by BN. Finally, BN is shown to smooth out the loss landscape (Santurkar, Tsipras, Ilyas, & Madry, 2018), which allows the usage of a large learning rate and makes the model more robust against changes in hyperparameters.

6.3 | Baseline models

We use the following models as baselines in our experiments:

- LDA: the original LDA model introduced in Blei et al. (2003). Here, we consider using Gibbs sampling (Griffiths & Steyvers, 2004) as the inference method.
- LFLDA: a model introduced in Nguyen et al. (2015) that incorporates word vectors to improve the topic model result on short text by using a mixture of the original Dirichlet multinomial and a latent feature component as the topic-to-word distribution. For the experiments, we use the code provided by the authors[7] and the recommended settings in the paper, which includes setting the mixture weight λ to 0.6, the Dirichlet

prior α and β to 0.1 and 0.01, respectively, the number of initial sampling iterations to 1,500 and the number of iterations involving word vectors to 500.

- GPU-DMM: the combination between auxiliary word embedding and Dirichlet process, as presented in Li et al. (2016).
- NVI: the VAE-based approach introduced in Miao et al. (2016).
- Was-A: The Wasserstein Autoencoder model presented in Nan et al. (2019).
- AVITM: This technique is published in Srivastava and Sutton (2017), using VAE to approximate the LDA process.
- AATM:The model uses the attention-based autoencoder technique for topic modelling in (Tian & Fang, 2019).
- RATM: The model uses the attention-enhance Bayes-based topic modelling for sequence-treated documents in (Li et al., 2017).
- ARNN: The model is similar to RATM, but employing RNN with attention for topic modelling in (Mei et al., 2017).

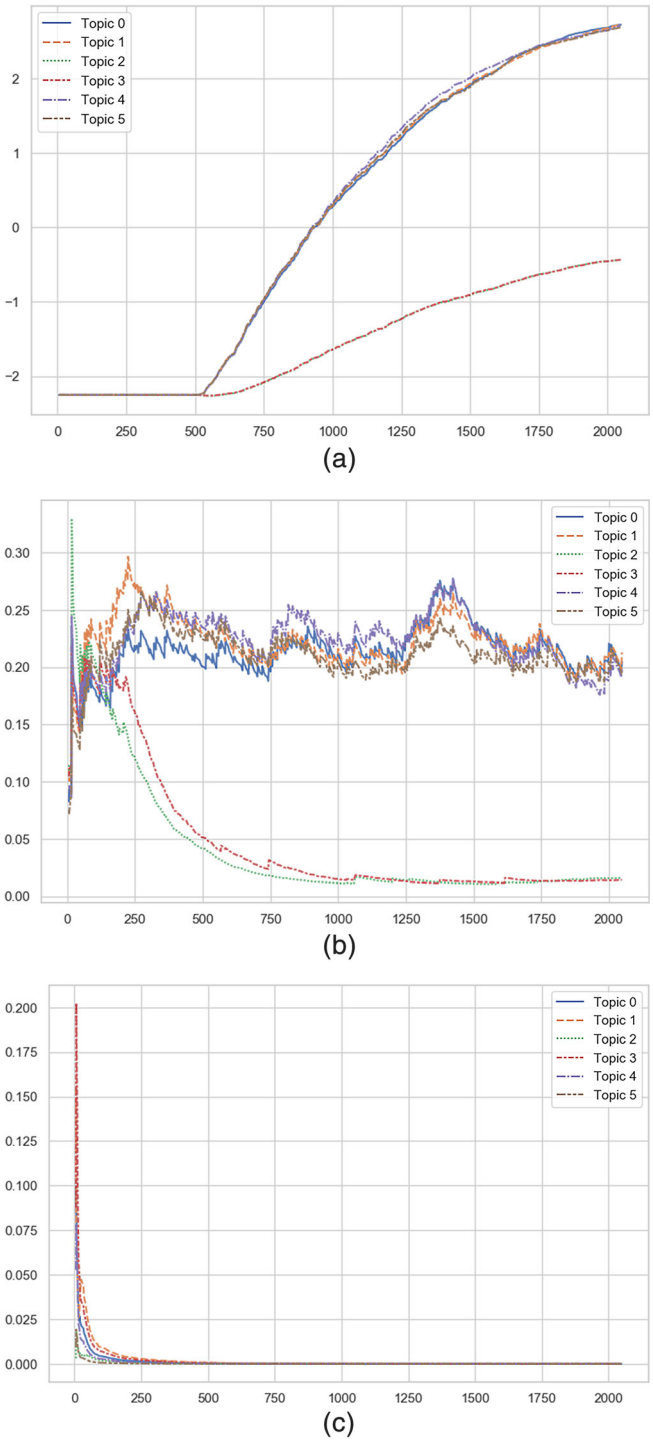
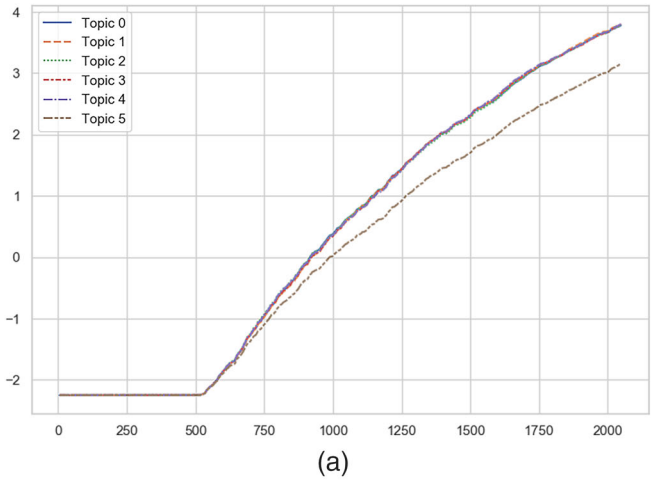
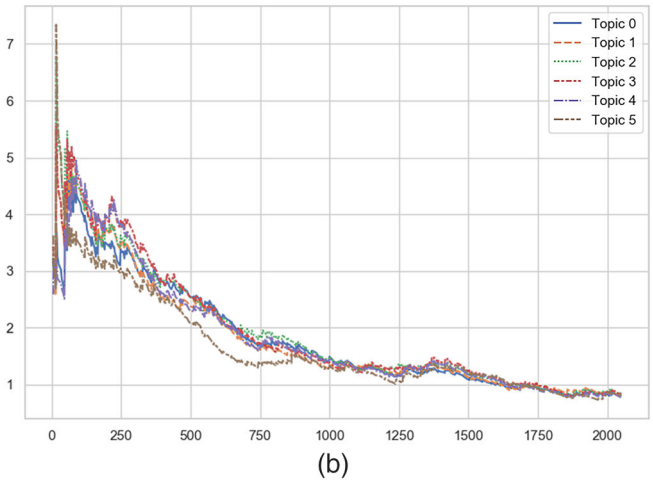


FIGURE 11 No BN between the fully connected layers and at the β matrix





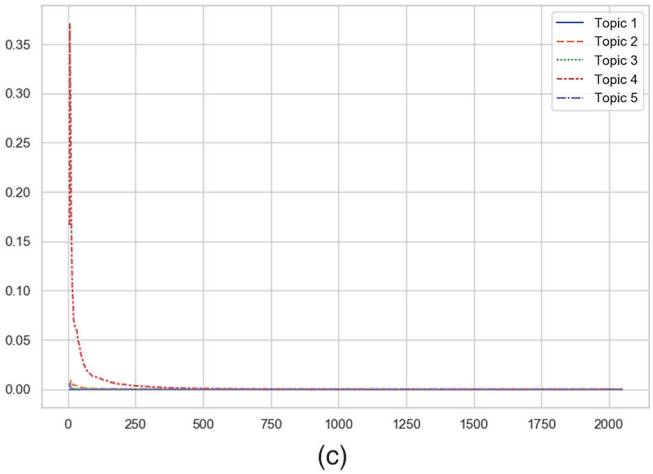


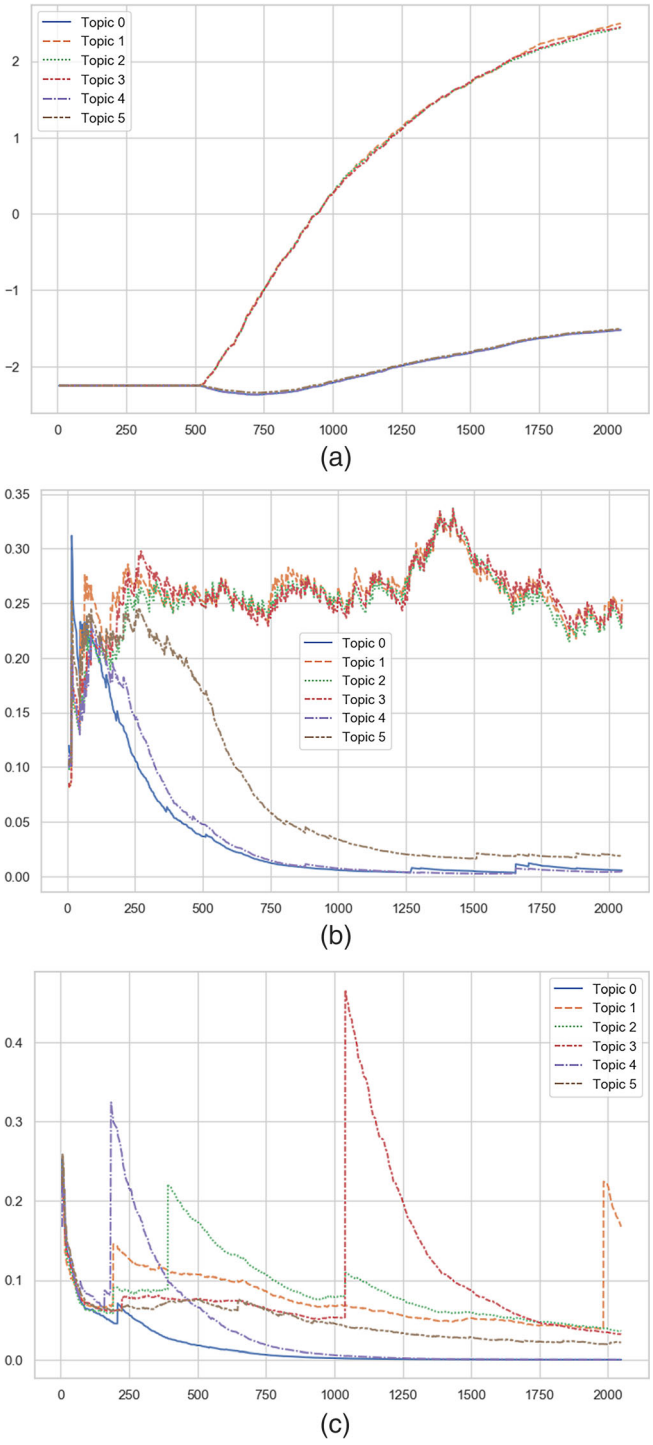
FIGURE 12 BN only at the β matrix and none between the fully connected layers

6.4 | Topic coherence

We examine the quality of each topic produced by our model via measuring how semantically coherent its top words are.

Based on the survey done in Röder, Both, and Hinneburg (2015), we use normalized pointwise mutual information (NPMI) as the metric for quantitative analysis of topic coherence. This metric is introduced in Bouma (2009) and is proven empirically to have a strong correlation with human evaluation. The NPMI is defined as:

FIGURE 13 BN only between the fully connected layers and none at the β matrix



$$\mathrm{NPMI}(w_i, w_j) = \sum_{i=0}^{N-2} \sum_{j=i+1}^{N-1} -\frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{\log P(w_i, w_j)} \tag{15}$$

where the probabilities are collected using a sliding window of 10 words on an external corpus. For each topic, its NPMI score is calculated using its top-15 words. We use *Palmetto*,[8] the tool provided by the authors of Röder et al. (2015) to measure topic coherence using Wikipedia as the external corpus. For each dataset, we calculate the coherence of each topic and use the average score of all the topics as the model's coherence.

Table 3 presents the NPMI scores from various models with different numbers of topics. The table shows that the methods based on variational autoencoder produce topics with relatively higher coherence than the ones based on LDA. On one hand, those results illustrate that the

methods based on variational autoencoder have successfully approximated Dirichlet distribution, which is essential to ensure good quality of the topic modelling process. On the other hand, this is also indicates the advantage of using word vectors, because the embedding process helps word vectors properly encode semantic information, so that topics created form these words have better coherence than the mere LDA-based processes, which assume input as one-hot vectors. However, when processing short text corpora (N20short, TMNtitle, Twitter and Snippet), our N-VAE and GPU-DMM methods enjoy better coherence due their better capability of handling short text processing. In our N-VAE method, we have combined bag-of-word model with *word* × *document* distribution when approximating the distribution with VAE. Meanwhile, the GPU-DMM method uses auxiliary words to enhance word vectors with more semantics information when embedding. However, the process of

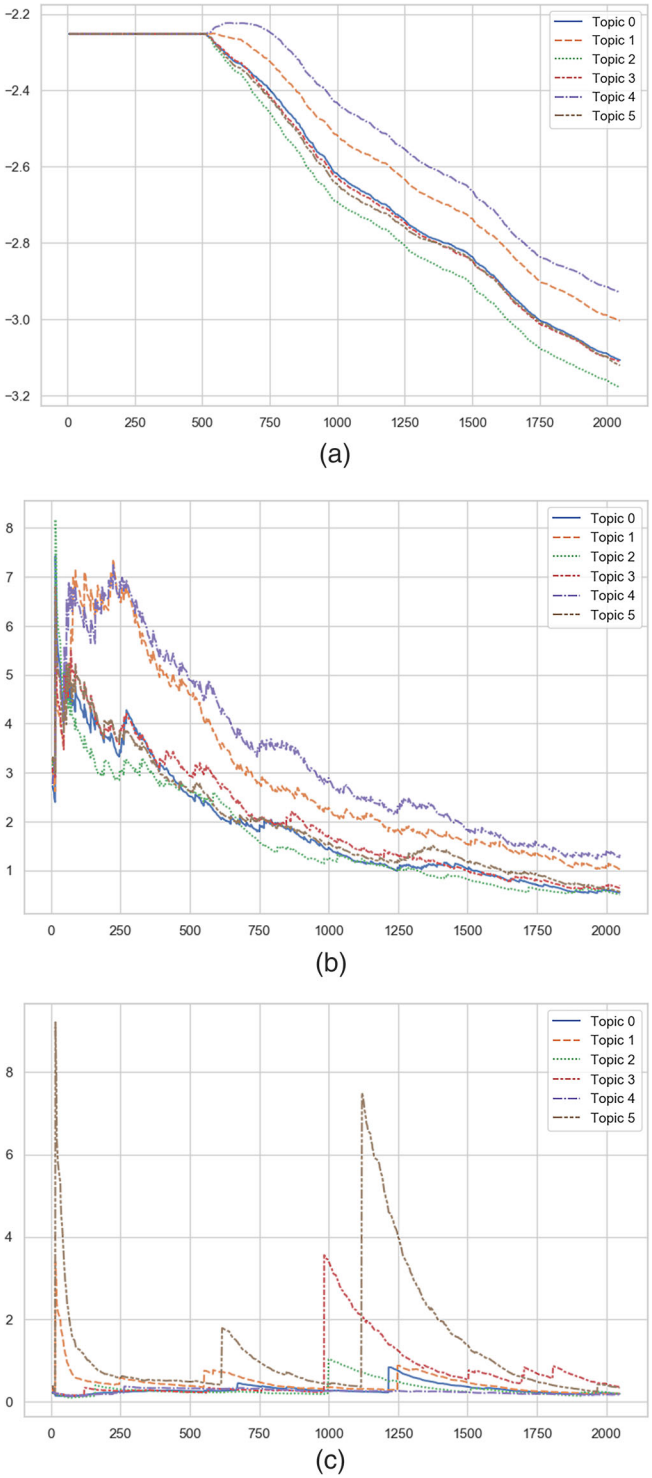


FIGURE 14 BN between the fully connected layers and at the β matrix

TABLE 4 Purity scores (higher is better) for the N20, N20small, N20short, TMN, TMNtitle, Twitter and Web Snippets dataset

Dataset	Number of topics	N-VAE glove	GPU-DMM	NVI	Was-A	LFLDA	AATM	RATM	ARMN
N20	6	0.293	0.298	0.298	**0.308**	0.291	0.248	0.303	0.221
	20	**0.602**	0.584	0.564	0.594	0.569	0.454	0.463	0.479
	40	**0.652**	0.624	0.614	0.634	0.616	0.564	0.569	0.576
	80	**0.664**	0.632	0.632	0.632	0.638	0.572	0.576	0.588
N20small	6	**0.252**	0.247	0.237	0.237	0.229	0.163	0.188	0.126
	20	**0.451**	0.408	0.418	0.428	0.439	0.372	0.362	0.351
	40	**0.524**	0.445	0.475	0.465	0.516	0.375	0.373	0.366
	80	0.586	0.543	0.563	0.553	**0.595**	0.493	0.479	0.495
N20short	6	**0.297**	0.249	0.269	0.269	0.278	0.27	0.185	0.232
	20	**0.375**	0.32	0.34	0.33	0.366	0.356	0.217	0.244
	40	**0.4**	0.355	0.325	0.335	0.395	0.388	0.237	0.285
	80	0.373	0.326	0.306	0.326	**0.429**	0.392	0.295	0.291
TMN	7	0.634	**0.672**	0.652	0.652	0.658	0.462	0.467	0.468
	20	0.754	0.748	0.738	**0.758**	0.716	0.678	0.575	0.57
	40	**0.766**	0.735	0.735	0.745	0.72	0.595	0.589	0.683
	80	**0.741**	0.681	0.681	0.711	0.725	0.699	0.696	0.694
TMNtitle	7	**0.634**	0.603	0.603	0.613	0.579	0.443	0.561	0.579
	20	**0.662**	0.659	0.649	0.649	0.619	0.489	0.572	0.579
	40	**0.654**	0.622	0.642	0.622	0.611	0.582	0.583	0.581
	80	**0.632**	0.588	0.588	0.608	0.598	0.498	0.592	0.588
Twitter	4	**0.615**	0.613	0.623	0.613	0.594	0.463	0.433	0.404
	20	0.637	**0.691**	0.651	0.671	0.665	0.471	0.561	0.561
	40	0.661	**0.689**	0.649	0.679	0.661	0.589	0.579	0.579
	80	0.654	0.669	0.649	**0.679**	0.654	0.589	0.599	0.591
Snippet	8	0.79	**0.816**	**0.816**	0.796	0.73	0.656	0.636	0.636
	20	0.849	**0.873**	**0.873**	0.863	0.724	0.673	0.663	0.693
	40	0.855	**0.875**	**0.875**	**0.875**	0.709	0.675	0.675	0.685
	80	**0.866**	0.837	0.837	0.857	0.703	0.677	0.697	0.697

Note: Bold entries imply the best result.

acquisition of those auxiliary words is often not trivial and hardly automatically proceeded, whereas the N-VAE method can be performed in a fully automatic manner.

As compared to the attention-based methods (AATM, RATM and ARNN), it shown that those methods achieve good performance when dealing with documents whose sizes are sufficiently long. It is because that those methods treat documents as sequences and use attention mechanism to capture context information of the whole documents. Thus, they required the sequences should be of considerably lengths to gain the context information. When handling short documents (the datasets of N20short, TMNtitle, Twitter and Snippet), the methods of RATM and ARNN suffer from poor performance. Meanwhile, the ATTM method still maintains good performance when processing the datasets of N20short, TMNtitle; since this method using Wordnet as an auxiliary source to infer the context of the short document. However, ATTM does not obtain good performance with the datasets of Twitter and Snippet, since Wordnet is developed in an academic manner, not appropriate to reflect the vocabulary used in social media channels.

6.5 | Document clustering

We measure to what extent the clustering result of the topic model agrees with the ground truth label. After calculating the topic proportion for each document using a topic model, we consider a document to belong to the topic with the highest probability in that document. We then calculate the similarity between clusters produced by the topic model and the ground truth label using two metrics: *cluster purity* (Manning, Raghavan, & Schütze, 2008), and *Silhouette index* (Rousseeuw, 1987). Whereas the *purity* metric indicates the fact that the items in a cluster should be as close

TABLE 5 Sihoulette scores (higher is better) for the N20, N20small, N20short, TMN, TMNtitle, Twitter and Web Snippets dataset

Dataset	Number of topis	N-VAE glove	GPU-DMM	NVI	Was-A	AVITM	LDA	LFLDA	AATM	RATM	ARMN
N20	6	**0.104**	0.09	0.077	0.077	0.08	0.104	0.101	0.094	0.087	0.095
	20	**0.53**	0.489	0.489	0.489	0.52	0.49	0.485	0.493	0.289	0.389
	40	**0.599**	0.572	0.558	0.558	0.56	0.581	0.55	0.579	0.472	0.458
	80	**0.616**	0.602	0.602	0.574	0.57	0.591	0.58	0.61	0.481	0.496
N20small	6	**0.048**	0.02	0.02	0.006	0.01	0.02	0.016	0.035	0.018	0.014
	20	**0.322**	0.294	0.308	0.294	0.31	0.263	0.306	0.317	0.194	0.208
	40	**0.423**	0.381	0.395	0.395	0.4	0.358	0.412	0.393	0.281	0.395
	80	**0.508**	0.494	0.481	0.494	0.49	0.471	0.521	0.503	0.394	0.481
N20short	6	**0.11**	0.082	0.082	0.068	0.07	0.066	0.083	0.102	0.058	0.078
	20	**0.217**	0.19	0.203	0.203	0.19	0.141	0.205	0.206	0.094	0.103
	40	**0.252**	0.224	0.21	0.224	0.22	0.187	0.245	0.241	0.152	0.188
	80	**0.214**	0.187	0.201	0.187	0.2	0.245	0.292	0.196	0.213	0.251
TMN	7	0.574	0.547	0.547	0.561	0.56	0.59	**0.608**	0.554	0.447	0.547
	20	**0.74**	0.726	0.726	0.726	0.73	0.7	0.688	0.734	0.626	0.526
	40	**0.757**	0.729	0.743	0.715	0.74	0.692	0.693	0.745	0.552	0.543
	80	**0.722**	0.708	0.708	0.694	0.71	0.688	0.7	0.718	0.575	0.608
TMNtitle	7	**0.574**	0.561	0.561	0.561	0.53	0.501	0.499	0.569	0.536	0.361
	20	**0.613**	0.586	0.586	0.572	0.57	0.541	0.554	0.594	0.538	0.456
	40	**0.602**	0.561	0.574	0.588	0.57	0.518	0.543	0.582	0.461	0.344
	80	**0.572**	0.558	0.558	0.558	0.56	0.503	0.525	0.565	0.488	0.458
Twitter	4	**0.548**	0.534	0.507	0.507	0.51	0.517	0.519	0.544	0.434	0.427
	20	0.579	0.551	0.565	0.565	0.54	0.543	**0.617**	0.567	0.451	0.465
	40	**0.62**	0.584	0.598	0.584	0.57	0.55	0.612	0.598	0.484	0.578
	80	**0.612**	0.561	0.561	0.588	0.57	0.561	0.602	0.574	0.491	0.381
Snippet	8	**0.79**	0.762	0.748	0.762	0.75	0.675	0.707	0.773	0.562	0.408
	20	**0.871**	0.857	0.843	0.83	0.84	0.694	0.699	0.865	0.565	0.443
	40	**0.879**	0.838	0.852	0.866	0.85	0.646	0.678	0.857	0.568	0.452
	80	**0.894**	0.853	0.881	0.867	0.87	0.646	0.67	0.879	0.585	0.481

Note: Bold entries imply the best result.

TABLE 6 Runtime of the latent Dirichlet allocation and the N-VAE on the N20short dataset with various number of topics

Dataset	Number of topics	N-VAE glove	GPU-DMM	NVI	Was-A	AVITM	LDA	LFLDA	AATM	RATM	ARMN
N20short	6	40.12	38.916	50.551	62.587	35.31	1,266.09	2,000.422	39.12	1,225.551	1,932.587
	20	40.26	44.882	60.808	73.355	41.02	2,007.49	3,071.46	41.26	1,826.808	2,838.355
	40	61.75	56.81	77.805	96.948	54.34	3,247.57	4,936.306	59.75	2,928.805	4,838.948
	80	73.12	68.733	91.4	112.605	62.15	4,068.06	6,386.854	72.12	3,129.4	5,839.605

to each other as possible, the *Silhouette* index additionally takes into account the distinction between clusters (i.e., the generated clusters should be as distinct as possible), in order to prevent the clustering process from generating clusters that are too similar to each other.

Tables 4 and 5, respectively present the purity scores and Silhouette values from various models with different numbers of topics. Table 4 shows that methods based on variational autoencoder can generate clusters with better purity than those based on LDA. This can be expected since it is predictable that the purity of a cluster has relative correlation with the coherence of the topic represented by this cluster. This also explains why our method of N-VAE and the GPU-DMM method have better purity score when handling corpora of short text documents.

Table 5 shows that N-VAE outperforms other models most of the time, in terms of Silhouette value. Even on the N20 dataset, the additional information from the word vector is still useful and allows our model to reach a much better score than the original LDA model. Compared to other VAE-based baseline approaches, N-VAE is the only method incorporates the categorical distribution of word-to-topic relations to the N-VAE process, so that topics, that is, clusters, have categorically distinguished to each other and thus enjoying higher Silouette values.

It is noted that when topics are treated as clusters of documents and corresponding metrics are measured, high scores can be attained if one maintains the Dirichlet distributions of keywords among topics. The attention-based methods (AATM, RATM and ARNN) also enjoy good purity scores on long documents datasets as the attention technique helps those methods capture contexts of long sequence documents. However, for the generic short text corpus (N20short, TMNtitle, Twitter and Snippet, only RATM and ARNN maintain good performance since they try to infuse the Dirichlet distribution to their processes by means of Gibbs sampling, whereas the Wordnet-based approach of AATM seems not very successful. All attention-based approaches did not perform well in terms of Purity and Silhouette scores on the specific datasets of Twitter and Snippet, explained by the naturally unconventional language used on the dataset and the short lengths of documents, which cause difficulty to construct informative context vectors required by the attention mechanism.

6.6 | Runtime

We compare the runtime between the LDA-based and VAE-based methods. We ran each model on the N20short dataset using a machine with 24 vCPUs and 90 GB of RAM rented on the Google Compute Engine8. Even though a GPU could be utilized to speed up the training process of our model, we only use the CPU for fairness. The results are shown in Table 6.

As mentioned, the N20short dataset consists around 1,800 documents and approximately 6,400 words in vocabulary size. In this table, the execution time is given in seconds. We did not measure the preprocessing time of each method.

Overall, the VAE-based approach requires approximately 1/30th the training time of the LDA to reach the desired performance. This is a significant advantage of this approach and also our model, enabling training on a large corpus while requiring only a small amount of time. Our N-VAE approach enjoys fast execution time since it does not need to load the corpus for Gibb sampling process, as required by the LDA method. Meanwhile, by precisely approximating the categorical distribution and Dirichlet distribution by two VAEs in a nested mechanism, we keep maintaining good accuracy performance as analysed in the previous sections.

Once the attention mechanism is employed, the AATM still enjoys fast processing speed, which is comparable to that of N-VAE. Meanwhile, similar to other LDA-based approaches, RATM and ARNN rely on Gibbs sampling to estimate the Dirichlet distribution and suffer from long execution time. This illustrates the fact that whereas neural-network-based approaches can be scalable to handle large dataset, integrating Dirichlet distribution to an end-to-end deep learning architecture in order to obtain good quality of the modelled topics is a non-trivial task, which we successfully handle by our proposed N-VAE architecture.

6.7 | Discussion

Overall, the *N-VAE* frequently returns better results than the original LDA model as well as the LFLDA on both the topic coherence evaluation and document clustering task. In terms of the significance of the word embedding, we found Word2Vec helps the model to reach better topic coherence scores than GloVe, while GloVe is evidently more beneficial to the document clustering task. This is actually quite logical, since the NPMI metric used in the topic coherence evaluation has a window-based nature similar to the Word2Vec. The NPMI employs a sliding window over tokens in an external corpus (Wikipedia) to calculate the cooccurrences between pairs of words, while the Word2Vec training method emphasizes capturing the relationships between a word and its neighbours, that is, local co-occurrences. This is not to posit that the NPMI is a well-rounded metric for topic coherence, since it fails to take into account words that obviously come from the same topic but rarely occur near each other, such as names of competing brands, for example, Lamborghini and Toyota. Conversely, the GloVe training method considers the cooccurrences within the entire document (global cooccurrences) and the indirect relationships between words, which is better in identifying words that belong to the same topic and grouping them together. In fact, table 3 in Pennington et al. (2014) suggests that GloVe is better than Word2Vec in word similarity tasks even when it is trained on a smaller corpus. This, in turn, leads to more words being included in one topic, which means more documents having the same label are assigned to the same cluster, reflected through a high agreement between the clustering result of the *N-VAE* and the ground truth label.

We consider how BN assists in the training process of our model. We conclude that the effect of BN is twofold: first, it helps solve the component collapsing problem of the model caused by the rapid saturation of the softmax function; and second, it permits the employment of high learning rates which speeds up convergence.

Finally, the major advantage of our model is its speed. Our model can extract topics from a corpus using only a fraction of the time required by the LFLDA, and the time gap increases with a larger number of topics. Our model not only succeeds in securing a time advantage but also

consumes less memory since it is trained using mini batches instead of loading the entire corpus into the memory, which makes it extremely useful for inducing topics of a large dataset of microtexts, for example, posts on social media.

Other baseline methods that rely on the VAE techniques also gain the same advantages of topic coherence (due to the embedding techniques of the word vectors) and execution speed (due to the neural network nature of VAE model). However our approach of N-VAE has outperformed most of those approaches on two extents. The first is the capability of processing short text, since our model incorporates the combination of bag-of-word model with *word* × *document* distribution of the whole corpus once producing topics for short text. The second is quality of the generated clusters, thanks to the incorporation of the categorical distribution of *word* × *topic* in our models.

7 | CONCLUSION

In this paper, we propose *N-VAE*, a model that can be combined with pretrained word vectors to induce latent topics from a dataset of microtexts. We present a distribution q that takes into account word embeddings to approximate the joint distribution of LDA using the variational inference method. We express this distribution using a neural network for a faster convergence speed and a smaller memory footprint so that our model can operate on large datasets. We also conducted experiments to compare our methods with other similar works using LDA and VAE. Experiments show the improved performance and runtime of our model in deriving latent topics from microtexts compared to other methods. These advantages make our model suitable for real-world use cases, such as categorizing a large collection of comments on social networks.

It is also noted that even though our approach is good when detecting topics in short texts, it cannot "memorize" the topics mentioned across multiple sequentially related documents, which is a common request in many domains, such an e-commerce. In Jin et al. (2018), LSTM is used to model topics among user reviews for recommendation in e-commerce systems. Since LSTM can "remember" information mentioned previously in the long documents, this technique can help to analyse long discussion between multiple reviews of users to mine the discussed topics. This characteristic can be used for more advanced task, such as sentiment analysis, as we already reported in Vo et al. (2019). However, as reviews in such social media channels are usually short, our approach of N-VAE may potentially improve further the analysed results once combined with RNN-based techniques like LSTM or GRU. It is also the direction that we pursue in the future.

ACKNOWLEDGEMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number: 06/2018/TN.

ORCID

Tho Quan https://orcid.org/0000-0003-0467-6254

ENDNOTES

[1] https://code.google.com/archive/p/word2vec/

[2] https://nlp.stanford.edu/projects/glove/

[3] This is also a finding of this paper, but we consider that it is not substantial enough to be claimed as a proper contribution of the paper.

[4] http://www.sananalytics.com/lab/index.php

[5] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz

[6] https://github.com/stanfordnlp/CoreNLP

[7] https://github.com/datquocnguyen/LFTM

[8] https://github.com/dice-group/Palmetto

REFERENCES

Bansal, T., Belanger, D., & McCallum, A. (2016). Ask the gru: Multi-task learning for deep text recommendations. 107–114.

Batmanghelich, K., Saeedi, A., Narasimhan, K., & Gershman, S. (2016). Nonparametric spherical topic modeling with word embeddings. Paper presented at: Proceedings of the conference. Association for computational linguistics. Meeting. (Vol. *2016*, p. 537).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. Paper presented at: Proceedings of GSCL, 31–40.

Costa, F. A, Yamaguchi, Y., Traina, A. J. M., Traina, C., & Faloutsos, C. (2015). RSC: Mining and Modeling Temporal Activity in Social Media. Paper presented at: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 269–278). New York, NY, USA: ACM. https://doi.org/10.1145/2783258.2783294

Das, R., Zaheer, M., & Dyer, C. (2015, July). Gaussian LDA for Topic Models with Word Embeddings. Paper presented at: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 795–804). Beijing, China: Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-1077

Fellbaum, C. (2012). Wordnet. The encyclopedia of applied linguistics.
Figurnov, M., Mohamed, S., & Mnih, A.. (2018, May). Implicit Reparameterization Gradients. arXiv:1805.08498 [cs, stat].
Goldstone, A., & Underwood, T. (2014). The quiet transformations of literary studies: What thirteen thousand scholars could tell us. *New Literary History*, *45*, 359–384. https://doi.org/10.1353/nlh.2014.0025
Griffiths, T. L., & Steyvers, M. (2004). Finding scientihc topics. *Proceedings of the National Academy of Sciences*, *101*(suppl 1), 5228–5235. https://doi.org/10.1073/pnas.0307752101
Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. Paper presented at: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 50–57). New York, NY, USA: ACM. event-place: Berkeley, California, USA. https://doi.org/10.1145/312624.312649
Hong, L., & Davison, B. D. (2010). Empirical Study of Topic Modeling in Twitter. Paper presented at: Proceedings of the First Workshop on Social Media Analytics (pp. 80–88). New York, NY, USA: ACM. https://doi.org/10.1145/1964858.1964870
Jang, E., Gu, S., & Poole, B. (2016, November). Categorical Reparameterization with Gumbel-Softmax. arXiv:1611.01144 [cs, stat].
Jansson, P., & Liu, S. (2017). Topic modelling enriched LSTM models for the detection of novel and emerging named entities from social media. 4329–4336.
Jin, M., Luo, X., Zhu, H., & Hankui, H. (2018). Combining deep learning and topic modeling for review understanding in context-aware recommendation. 1605–1614.
Jin, O., Liu, N. N., Zhao, K., Yu, Y., & Yang, Q. (2011). Transferring Topical Knowledge from Auxiliary Long Texts for Short Text Clustering. Paper presented at: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (pp. 775–784). New York, NY, USA: ACM. event-place: Glasgow, Scotland, UK. https://doi.org/10.1145/2063576.2063689
Jo, Y., Lee, L., & Palaskar, S. (2017). Combining lstm and latent topic modeling for mortality prediction. arXiv preprint arXiv:1709.02842.
Kennedy, H., & Moss, G. (2015). Known or knowing publics? Social media data mining and the question of public agency. *Big Data & Society*, *2*(2), 2053951715611145. https://doi.org/10.1177/2053951715611145
Kingma, D. P., & Ba, J. (2014, December). Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].
Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. CoRR, abs/1312.6114.
Lazard, A., Scheinfeld, E., Bernhardt, J. M., Wilcox, G. B., & Suran, M. (2015). Detecting themes of public concern: A text mining analysis of the centers for disease control and prevention's Ebola live twitter chat. *American Journal of Infection Control*, *43*(10), 1109–1111.
Li, C., Wang, H., Zhang, Z., Sun, A., & Ma, Z. (2016). Topic modeling for short texts with auxiliary word embeddings. Paper presented at: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval (pp. 165–174).
Li, S., Zhang, Y., Pan, R., Mao, M., & Yang, Y. (2017). Recurrent attentional topic model. Paper presented at: Thirty-first AAAI conference on artificial intelligence.
Liu, L., Tang, L., Dong, W., Yao, S., & Zhou, W. (2016, September). An overview of topic modeling and its current applications in bioinformatics. *Springerplus*, *5*(1), 1608. https://doi.org/10.1186/s40064-016-3252-8
Lu, H.-Y., Xie, L.-Y., Kang, N., Wang, C.-J., & Xie, J.-Y. (2017). Don't forget the quantihable relationship between words: Using recurrent neural network for short text topic discovery.
Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.
Maddison, C. J., Mnih, A., & Teh, Y. W. (2016, November). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. arXiv:1611.00712 [cs, stat].
Manning, C. D., Raghavan, S., & Schütze, H. (2008). *Introduction to information retrieval*. England: Cambridge University Press.
Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013). Improving LDA Topic Models for Microblogs via Tweet Pooling and Automatic Labeling. Paper presented at: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 889–892). New York, NY, USA: ACM. https://doi.org/10.1145/2484028.2484166
Mei, H., Bansal, M., & Walter, R. M. (2017). Coherent dialogue with attention-based language models. Paper presented at: Thirty-first AAAI conference on artificial intelligence.
Miao, Y., Yu, L., & Blunsom, P. (2016). Neural variational inference for text processing. Paper presented at: International conference on machine learning (pp. 1727–1736).
Mikolov, T., Chen, K., & Corrado, G., Jeffrey, D. (2013a). Efficient estimation of word representations in vector space. arXiv Preprint arXiv:1301.3781.
Mimno, D. (2012, April). Computational historiography: Data mining in a century of classics journals. *Journal on Computing and Cultural Heritage*, *5*(1), 3: 1–3:19. https://doi.org/10.1145/2160165.2160168
Nan, F., Ding, R., Nallapati, R., & Xiang, B. (2019). Topic modeling with wasserstein autoencoders. arXiv Preprint arXiv:1907.12374.
Newman, D. J., & Block, S. (2006). Probabilistic topic decomposition of an eighteenth-century American newspaper. *Journal of the American Society for Information Science and Technology*, *57*(6), 753–767. https://doi.org/10.1002/asi.20342
Nguyen, D. Q., Billingsley, R., Du, L., & Johnson, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, *3*, 299–313.
Pandey, R., & Purohit, H. (2018). CitizenHelper-Adaptive: Expert-Augmented Streaming Analytics System for Emergency Services and Human-itarian Organizations. Paper presented at: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 630-633). https://doi.org/10.1109/ASONAM.2018.8508374
Paul, M. J., Sarker, A., Brownstein, J. S., Nikfarjam, A., Scotch, M., Smith, K. L., & Gonzalez, G. (2015). Social media mining for public health monitoring and surveillance. Paper presented at: Biocomputing 2016 (pp. 468–479). World Scientific. https://doi.org/10.1142/9789814749411_0043
Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. event-place: Doha, Qatar. https://doi.org/10.3115/v1/D14-1162
Petterson, J., Buntine, W., Narayanamurthy, S. M., Caetano, T. S., & Smola, A. J. (2010). Word features for latent Dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems 23* (pp. 1921–1929). Vancouver, British Columbia, Canada: Curran Associates, Inc.
Phan, X., Nguyen, C., Le, D., Nguyen, L., Horiguchi, S., & Ha, Q. (2011). A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 961–976. https://doi.org/10.1109/TKDE.2010.27

Qiang, J., Chen, P., Wang, T., & Wu, X. (2017). Topic modeling over short texts by incorporating word Embeddings. In J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, & Y.-S. Moon (Eds.), *Advances in knowledge discovery and data mining* (pp. 363–374). Cham, Switzerland: Springer International Publishing.

Quan, X., Kit, C., Ge, Y., & Pan, S. J. (2015). Short and Sparse Text Topic Modeling via Self-aggregation. Paper presented at: Proceedings of the 24th International Conference on Artificial Intelligence (pp. 2270–2276). AAAI Press.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. Paper presented at: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (pp. II–1278–II–1286). JMLR.org. event-place: Beijing, China.

Riddell, A. (2014). How to read 22,198 journal articles: studying the history of German studies with topic models. In K. Belgum, T. Boes, L. Koepnick, T. Kontje, K. Mellmann, N. Pethes, et al. (Authors) & M. Erlin & L. Tatlock (Eds.), *Distant readings: Topologies of German culture in the long nineteenth century* (pp. 91–114). Boydell & Brewer.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. Paper presented at: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (pp. 399–408). New York, NY, USA: ACM. event-place: Shanghai, China. https://doi.org/10.1145/2684822.2685324

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Sadilek, A., Kautz, H., DiPrete, L., Labus, B., Portman, E., Teitel, J., & Silenzio, V. (2017). Deploying nEmesis: Preventing foodborne illness by data mining social media. *AI Magazine*, *38*(1), 37–48. https://doi.org/10.1609/aimag.v38i1.2711

Sahami, M., & Heilman, T. D. (2006). A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. Paper presented at: Proceedings of the 15th International Conference on World Wide Web (pp. 377–386). New York, NY, USA: ACM. event-place: Edinburgh, Scotland. https://doi.org/10.1145/1135777.1135834

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31* (pp. 2483–2493). Montreal, Canada: Curran Associates, Inc.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. https://doi.org/10.1145/3137597.3137600

Song, Y., Wang, H., Wang, Z., Li, H., & Chen, W. (2011, June). Short Text Conceptualization Using a Probabilistic Knowledgebase. Paper presented at: Twenty-Second International Joint Conference on Artificial Intelligence.

Srivastava, A., & Sutton, C. (2017). Autoencoding variational inference for topic models. arXiv:1703.01488 [stat].

Tamimi, I., Lamrani, E. K., & Kamili, M. E. (2017). Community detection through topic modeling in social networks. In E. Sabir, A. García Armada, M. Ghogho, & M. Debbah (Eds.), *Ubiquitous networking* (pp. 70–80). Cham, Switzerland: Springer International Publishing.

Teh, Y. W., Newman, D., & Welling, M. (2006). A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. Paper presented at: Proceedings of the 19th International Conference on Neural Information Processing Systems (pp. 1353–1360). Cambridge, MA, USA: MIT Press.

Tian, T., & Fang, Z. F. (2019). Attention-based autoencoder topic model for short texts. *Procedia Computer Science*, *151*, 1134–1139.

Vitale, D., Ferragina, P., & Scaiella, U. (2012). Classification of short texts by deploying topical annotations. In R. Baeza-Yate, A. P. de Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, & F. Silvestri (Eds.), *Advances in information retrieval* (pp. 376–387). Barcelona, Spain: Springer Berlin Heidelberg.

Vo, K., Nguyen, T., Pham, D., Nguyen, M., Truong, M., Mai, T., & Quan, T. T. (2019). Combination of domain knowledge and deep learning for sentiment analysis of short and informal messages on social media. *International Journal of Computational Vision and Robotics*, *9*(5), 458–485.

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems 22* (pp. 1973–1981). Vancouver, British Columbia, Canada: Curran Associates, Inc.

Weng, J., Lim, E.-P., Jiang, J., & He, Q. (2010). Twitterrank: Finding Topic-sensitive Inhuential Twitterers. Paper presented at: Proceedings of the Third ACM International Conference on Web Search and Data Mining (pp. 261–270). New York, NY, USA: ACM. https://doi.org/10.1145/1718487.1718520

Xun, G., Li, Y., Zhao, W. X., Jing, G., & Aidong, Z. (2017). A correlated topic model using word embeddings. Paper presented at: *Ijcai* (pp. 4207–4213).

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013, 05). A biterm topic model for short texts. Paper presented at: Www 2013 - Proceedings of the 22nd International Conference on World Wide Web (p. 1445–1456). https://doi.org/10.1145/2488388.2488514

Yang, T.-I., Torget, A. J., & Mihalcea, R. (2011). Topic Modeling on Historical Newspapers. Paper presented at: Proceedings of the 5th ACL-HLT Workshop on Lan-guage Technology for Cultural Heritage, Social Sciences, and Humanities (pp. 96–104). Stroudsburg, PA, USA: Association for Computational Linguistics.

Yin, J., & Wang, J. (2014). A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering. Paper presented at: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 233–242). New York, NY, USA: ACM. https://doi.org/10.1145/2623330.2623715

Yin, Z., Cao, L., Han, J., Zhai, C., & Huang, T. (2011). Geographical Topic Discovery and Comparison. Paper presented at: Proceedings of the 20th International Conference on World Wide Web (pp. 247–256). New York, NY, USA: ACM. https://doi.org/10.1145/1963405.1963443

Zhang, W., Li, Y., & Wang, S. (2019). Learning document representation via topic-enhanced lstm model. *Knowledge-Based Systems*, *174*, 194–204.

Zhu, Q., Feng, Z., & Li, X.. (2018). Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. Paper presented at: Proceedings of the 2018 conference on empirical methods in natural language processing (pp. 4663–4672).

AUTHOR BIOGRAPHIES

Trung Tring received his B.Eng. (Honors) degree in Computer Science from Ho Chi Minh City University of Technology in 2018. His current research interests incvolve natural language processing, image processing and deep learning.

Dr. Tho Quan is an Associate Professor in the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam. He received his B.Eng. degree in Information Technology from HCMUT in 1998 and received Ph.D degree in 2006 from Nanyang Technological University, Singapore. His current research interests include formal methods, program analysis/verification, the Semantic Web, machine learning/data mining and intelligent systems. Currently, he is the Vice Dean for Academic Affairs of the Faculty.

Trung Mai is currently working as a researcher in the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology. He received his M.Eng (2015) in Computer Science from James Cook University, Australia and B.Eng (2013) in Computer Engineering from Auckland University of Technology, New Zealand. His current research interests involve machine learning and deep learning.

How to cite this article: Trinh T, Quan T, Mai T. Nested variational autoencoder for topic modelling on microtexts with word vectors. *Expert Systems*. 2021;38:e12639. https://doi.org/10.1111/exsy.12639