Mixture Variational Autoencoder of Boltzmann Machines for Text Processing

Bruno Guilherme Gomes[1], Fabricio Murai[1], Olga Goussevskaia[1], Ana Paula Couto da Silva[1]

Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
{brunoguilherme, murai, olga, ana.coutosilva}@dcc.ufmg.br

Abstract. Variational autoencoders (VAEs) have been successfully used to learn good representations in unsupervised settings, especially for image data. More recently, mixture variational autoencoders (MVAEs) have been proposed to enhance the representation capabilities of VAEs by assuming that data can come from a mixture distribution. In this work, we adapt MVAEs for text processing by modeling each component's joint distribution of latent variables and document's bag-of-words as a graphical model known as the Boltzmann Machine, popular in natural language processing for performing well in a number of tasks. The proposed model, MVAE-BM, can learn text representations from unlabeled data without requiring pre-trained word embeddings. We evaluate the representations obtained by MVAE-BM on six corpora w.r.t. the perplexity metric and accuracy on binary and multi-class text classification. Despite its simplicity, our results show that MVAE-BM's performance is on par with or superior to that of modern deep learning techniques such as BERT and RoBERTa. Last, we show that the mapping to mixture components learned by the model lends itself naturally to document clustering.

1 Introduction

Digital libraries and online social networks are current examples of ecosystems where large volumes of textual data are generated by users at every instant. On average, it is estimated that 500 million tweets are posted daily on Twitter[1], while 600 articles are created every day on Wikipedia.[2]. Similar figures also hold for other digital platforms such as Amazon Review, Yahoo Answers and Yelp Reviews. This explains in part the ever increasing importance of analyzing user-generated patterns in large textual data sets for Natural Language Processing (NLP) research.

In the last two decades, probabilistic graphical models (PGMs) have underpinned many successful applications in NLP [15,12]. Many popular word embeddings methods, such as word2vec [13] and GloVe [16], are based on simple Bayesian networks, which are PGMs defined over directed acyclic graphs. The

[1] http://www.tweetstats.com/
[2] https://en.wikipedia.org/wiki/Wikipedia:Statistics

success of PGMs in NLP stems from their ability to use unlabeled samples effectively for learning complex patterns in the data by allowing to explicitly specify dependencies among variables. For some more complex PGMs, exact inference is intractable due to the calculation of high-dimensional integrals. In these cases, variational inference techniques for approximating conditional distributions have been proposed and successfully applied to address the computational complexity issues [8,10].

In computer vision, similar approximations have been used in non-deterministic neural network models for learning compact image representations in unsupervised settings. These models, called variational autoencoders [8], typically consist of two networks respectively called encoder and decoder. The role of the encoder is to obtain a compact representation – an encoding – of an input image through non-linear transformations. This encoding is combined with some noise, i.e., a random variable sampled from a Gaussian distribution, and passed onto the decoder, whose role is to recover the original images through more non-linear transformations. More recently, a mixture variational autoencoder (MVAE) was proposed to make better use of the latent representation space [7].

In this paper, we propose a novel framework based on MVAE for text processing. Each mixture component models the joint distribution of the latent variables and the bag-of-words vector that represents a document. This distribution is represented as the graphical model known as the Boltzmann Machine, popular in NLP for performing well in a number of tasks and for being efficiently trained with variational learning due to its simple structure [2]. Despite the current trends in deep learning, we show that a shallow network can be effectively used as an encoder.

Our model, named MVAE-BM[3], can learn text representations from unlabeled data without requiring pre-trained word embeddings. MVAE-BM takes as input the bag-of-word vector representing a document and outputs its latent representation. We evaluate the representations obtained by MVAE-BM using six corpora w.r.t. the perplexity metric and accuracy on text classification. In spite of its simplicity, our results demonstrate that MVAE-BM's performance is on par with or superior to that of sophisticated deep learning techniques such as BERT [4] and RoBERTa [9]. Last, we show that the association between text and mixture component learned by the model lends itself naturally to document clustering.

2 Related work

The task of learning patterns in large textual data sets has received significant interest in the last two decades. Here we discuss the main fronts of research related to our work.

Probabilistic Graphical Models (PGMs): PGMs provide a declarative language for blueprinting prior knowledge and valuable relationships in complex

[3] https://github.com/brunoguilherme1/MVAE-BM/

datasets. They contributed to fundamental advances in NLP, such as Topic Modeling [20] and word embedding [13,16]. A simple, yet effective graphical model used for language modeling is the Boltzmann Machine. This technique represents texts as bags-of-words and aims to learn their latent representation [2]. While these models represent documents using vectors of binary latent variables (since they are based on the Restricted Boltzmann Machine), MVAE-BM employs dense continuous document representations that are both expressive and easy to train.

Variational Autoencoder (VAE): VAE is a generative model that can be seen as an improved version of a standard autoencoder. VAE models are able to learn meaningful representations from the data in an unsupervised fashion. Variational inference with the re-parameterization trick was initially proposed in [8] and thereafter VAE has been widely adopted as a generative model for images [7]. Our MVAE-BM builds its encoder networks based on the VAE strategy [8] for the estimation of the latent variables present in the Boltzmann Machine and the Gumbel-Softmax strategy [6] to efficiently estimate the latent indicator variable of the mixture model.

Recently, several studies have presented efficient ways of combining PGM and VAE to solve NLP problems, with similar outcomes to MVAE-BM. In [23] an approach is presented for text modeling with latent information explicitly modeled as a Dirichlet variable. [12] and [11] introduced a generic variational inference framework for generative and conditional models of text, as well as alternative neural approaches for topic modeling. More recently, [15] combined non-parametric distribution models with VAE for text modeling.

Even though a mixture model using VAE has already shown promising results [7], MVAE-BM differs from the techniques listed above because it uses two neural networks to encode its latent variables and, in this way, it provides an estimation of the Boltzmann Machine as well as its mixture.

3 The MVAE-BM model: Mixture Variational Autoencoder of Boltzmann Machines

In this section, we present MVAE-BM, an unsupervised model for document representation, based on mixture variational autoencoders. We first briefly introduce how variational autoencoders are used to estimate latent representations.

3.1 Background on variational autoencoders

A variational autoencoder (VAE) is a generative model which combines the encoder-decoder architecture for unsupervised learning with variational inference. In a VAE, the latent variables are sampled from a distribution (typically Gaussian) whose parameters are computed by passing the input through the encoder. VAE modifies the autoencoder network by replacing the latent variable h of an input $\boldsymbol{x}$ with a learned posterior recognition model $p_\theta(h|\boldsymbol{x})$. Let

$X = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ be a dataset comprised of N i.i.d. samples from a random variable $\boldsymbol{x}$, and h be an unobserved continuous random variable, assuming that $\boldsymbol{x}$ is dependent on h. The marginal distribution of $\boldsymbol{x}$ is defined as:

$$p(\boldsymbol{x};\theta) = \int p(h;\theta)p(\boldsymbol{x}|h;\theta)dh. \tag{1}$$

In practice, the integral in Eq. (1) is intractable [8]. Hence, VAE uses a recognition model $q_\phi(h|\boldsymbol{x})$ to approximate the true posterior $p_\theta(h|\boldsymbol{x})$. So, instead of maximizing the marginal likelihood directly, the objective function becomes the variation lower bound, a.k.a. the evidence lower bound (ELBO) of the marginal:

$$\mathcal{L}(\boldsymbol{x};\theta,\phi) = \mathbb{E}_{q_\phi(h|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|h)] - \mathrm{KL}(q_\phi(h|\boldsymbol{x})||p_\theta(h)),$$

where $q_\phi(h|\boldsymbol{x})$ is the approximation distribution variational for the true posterior $p_\theta(h|\boldsymbol{x})$. In the VAE model, $q_\phi(h|\boldsymbol{x})$ is known as the recognition (encoder) model, and $p_\theta(\boldsymbol{x}|h)$, the decoder model. Both encoder and decoder models are implemented via neural architectures. As discussed in [8], optimizing the marginal log-likelihood is essentially equivalent to maximizing $\mathcal{L}(\boldsymbol{x};\theta,\phi)$, i.e., the ELBO, which consists of two terms. The first term is the expected reconstruction error, indicating how well the model can reconstruct data, given a latent variable. The second term is the KL divergence between the approximate posterior and the prior, acting as a regularization term that forces the learned posterior to be as close to the prior as possible. The prior $p_\theta(h)$ and the variational posterior $q_\phi(h|\boldsymbol{x})$ are frequently chosen from conjugate distribution families, allowing the KL divergence to be calculated analytically [8,6].

3.2 Proposed Model

MVAE-BM is an unsupervised learning model where two vectors of hidden variables, $\boldsymbol{h} \in \mathbb{R}^H$ and $\boldsymbol{c} \in \mathbb{R}^K$, are used for representing documents. Let V be the vocabulary and $\boldsymbol{x} \in \mathbb{R}^{|V|}$ be the bag-of-words representation of a document. We consider the generative model $p(\boldsymbol{x},\boldsymbol{h},\boldsymbol{c}) = p_{\boldsymbol{\pi}}(\boldsymbol{c})p(\boldsymbol{h})p_{\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{h},\boldsymbol{c})$, in which the latent variable $\boldsymbol{h}$ is generated from a centered multivariate Gaussian $\mathcal{N}(\mathbf{0},\mathbf{I})$, and the latent indicator $\boldsymbol{c}$ is generated from a categorical distribution Multinomial($\boldsymbol{\pi}$). The latent indicator $\boldsymbol{c} = [c_1, c_2, \ldots, c_K]$ satisfies the conditions $c_i \in \{0,1\}$, $\sum_{i=1}^{K} c_i = 1$. Each $\boldsymbol{x}$ is associated with an unique sample of $\boldsymbol{h}$, and is generated from a single component in the mixture model $p_{\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{h},\boldsymbol{c})$. The generative process is given by:

$$\begin{aligned} \boldsymbol{c} &\sim \prod_{k=1}^{K} \boldsymbol{\pi}_k^{c_k}, \\ \boldsymbol{h} &\sim \mathcal{N}(\mathbf{0},\mathbf{I}), \\ \boldsymbol{x}|\boldsymbol{h},\boldsymbol{c} &\sim \prod_{k=1}^{K} p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h})^{c_k}, \end{aligned} \tag{2}$$

where K is the predefined number of components in the mixture, and each component $p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h})$ is an energy function based on the Boltzmann machine [14] parameterized by $\boldsymbol{\Theta}^{(k)}$. For $K = 1$, it reduces to a VAE. In a VAE, an encoder network is used for learning a function $q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})$ that compresses documents' original representation into a low-dimensional continuous space. In a MVAE, an additional encoder network is needed to learn the function $q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})$ that clusters documents into specific groups. We found that using a simple Multi-Layer Perceptron (MLP) with two hidden layers for each of MVAE-BM's encoders works well in practice. For the decoder model $p_{\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{h},\boldsymbol{c}) = \prod_{k=1}^{K} p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h})^{c_k}$, MVAE-BM uses a simple softmax decoder to reconstruct the document by independently generating words given $\boldsymbol{c}$ and $\boldsymbol{h}$.

To maximize the log-likelihood of a document $\boldsymbol{x}$, we derive the ELBO of $\mathcal{L}(\boldsymbol{x};\boldsymbol{\Theta},\boldsymbol{\phi},\boldsymbol{\eta})$:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})}\left[\sum_{k=1}^{K} c_k \log p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})||p(\boldsymbol{h})) - \mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})||p(\boldsymbol{c})). \tag{3}$$

The conditional probability over words in a document $p_{\boldsymbol{\Theta}_k}(\boldsymbol{x}|\boldsymbol{h})$ is modeled by the multinomial logistic regression energy with parameters $\boldsymbol{\Theta}^{(k)} = (\boldsymbol{R}^{(k)}, \boldsymbol{b}^{(k)})$:

$$\begin{aligned} p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h}) &= \frac{1}{Z}\exp(-E(\boldsymbol{x};\boldsymbol{h},\boldsymbol{\Theta}^{(k)})), \\ E(\boldsymbol{x};\boldsymbol{h},\boldsymbol{\Theta}^{(k)}) &= -\boldsymbol{h}^{\top}\boldsymbol{R}^{(k)}\boldsymbol{x} - (\boldsymbol{b}^{(k)})^{\top}\boldsymbol{x}, \end{aligned}$$

where Z is the partition function, $\boldsymbol{R}^{(k)} \in \mathbb{R}^{H\times|V|}$ is the semantic word embedding and $\boldsymbol{b}^{(k)} \in \mathbb{R}^{|V|}$ is the bias term for the k-th mixture component. Figure 1 depicts the complete architecture for the recognition and generative models. A vector $\boldsymbol{x}$ representing a document passes through two neural networks (encoders) in parallel to obtain the latent representations $\boldsymbol{c}$ and $\boldsymbol{h}$ used by the mixture of Boltzmann machines.

The posterior approximation $q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})$ is conditioned on the current document $\boldsymbol{x}$. The inference network $q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})$ is modeled as:

$$\begin{aligned} q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x}) &\sim \mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2(\boldsymbol{x}))), \\ \boldsymbol{l} &= \boldsymbol{g}(\boldsymbol{f}_{A_2}(\boldsymbol{g}(\boldsymbol{f}_{A_1}(\boldsymbol{x})))), \\ \boldsymbol{\mu} &= \boldsymbol{f}_{A_3}(\boldsymbol{l}), \\ \log\boldsymbol{\sigma} &= \boldsymbol{f}_{A_4}(\boldsymbol{l}), \end{aligned}$$

where $\boldsymbol{f}_{A_i}(.)$ is the function represented by a linear layer A_i, $i = 1,\ldots,4$, and $\boldsymbol{g}(.)$ is an activation function. For each document $\boldsymbol{x}$, the neural network computes the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ that parameterize the distribution of the latent variable $\boldsymbol{h}$. Since the prior $p(\boldsymbol{h})$ is a standard Gaussian, the KL-Divergence $\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})||p(\boldsymbol{h}))$ can be computed analytically [8].

For $q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})$ we use a Gumbel-softmax as a proxy for the true posterior. The Gumbel-softmax [6,10] is a continuous approximation for sampling from a

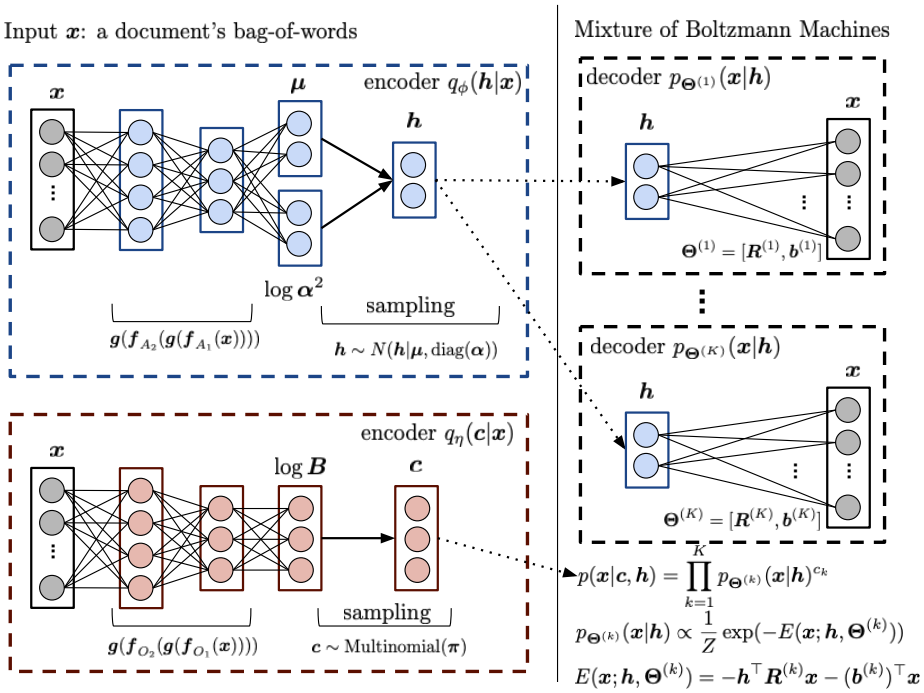

Fig. 1: MVAE-BM encoders $q_\phi(\boldsymbol{h}|\boldsymbol{x})$ and $q_\eta(\boldsymbol{c}|\boldsymbol{x})$ compress document $\boldsymbol{x}$ into latent representations $\boldsymbol{h}$ and $\boldsymbol{c}$. Each of the K decoders is a Boltzmann machine that computes $p_{\boldsymbol{\Theta}^{(k)}}(\boldsymbol{x}|\boldsymbol{h})$ through the energy function $E(\boldsymbol{x}; \boldsymbol{h}, \boldsymbol{\Theta}^{(k)})$. The mixture is controlled by the latent indicator vector $\boldsymbol{c}$.

categorical distribution. More specifically, the recognition $q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})$ is given by:

$$q_{\boldsymbol{\eta}}(c_i = 1|\boldsymbol{x}) \sim \frac{\exp((\log(B_i) + \epsilon_i)/\tau)}{\sum_{j=1}^{K} \exp((\log(B_j) + \epsilon_j)/\tau)},$$
$$\log(\boldsymbol{B}) = \boldsymbol{g}(\boldsymbol{f}_{O_2}(\boldsymbol{g}(\boldsymbol{f}_{O_1}(\boldsymbol{x})))),$$

where $\epsilon_i \sim \text{Gumbel}(0, 1)$ and $\boldsymbol{f}_{O_1}(.)$ and $\boldsymbol{f}_{O_2}(.)$ represent linear layers. The approximation is accurate for a discrete distribution when the hyperparameter τ (known as 'temperature') goes to 0 and smooth for $\tau > 0$. Hence, using this approach, the KL-Divergence $\text{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})||p(\boldsymbol{c}))$ can be easily evaluated [6].

Finally, to compute the expectation term in Eq. (3), we use the "re-parameterization trick" proposed in [8] (for $q_{\boldsymbol{\phi}}(\boldsymbol{h}|\boldsymbol{x})$) and in [6] (for $q_{\boldsymbol{\eta}}(\boldsymbol{c}|\boldsymbol{x})$).

4 Experimental results

In this section we describe the datasets used in our experiments, study MVAE-BM's hyperparameters and analyze MVAE-BM's performance on three different learning tasks: topic modeling, text classification, and document clustering.

Table 1: Properties of the datasets used in the experiments.

Dataset	Training set	Test set	Vocabulary	#Classes
20NewsGroups	11,314	7,531	2,000	20
Reuters (RCV1-v2)	794,414	10,000	10,000	90
Yelp Reviews	100,000	10,000	90,000	5
Yahoo Answers	100,000	10,000	20,000	10
TwitterHate	19,500	5,512	15,334	3
Subjectivity	9,756	3,323	5,563	2

4.1 Datasets and experimental setup

In our experiments, we leverage six corpora previously used in the literature for analyzing text representation models. Table 1 lists the number of samples in the training and test sets, vocabulary size and number of classes of each dataset. To make a direct comparison with the prior work, we reproduce the experiments in [20] (*20NewsGroups* and *RCV1-v2* datasets) and in [25] (*Yelp Reviews* and *Yahoo Answers* datasets), following the same pre-processing procedures and using the same training and test sets. Moreover, we compare the performance of MVAE-BM to the performance values reported in [20,11,5,22,15] and [25,23] for several baseline models, listed in Tables 2 and 3. For the *Subjectivity* and *TwitterHate* datasets, on the other hand, we created our own train-test splits, given that this information was not available from the related work.

Hyperparameter configuration: For each dataset, the MVAE-BM's hyperparameters were chosen by grid search in the training set. The search was performed over the values 50, 200, 1,000, 2,000 for the number of neurons in each layer A_1, A_2, A_3, A_4 and O_1, respectively. Moreover, the search covered the values $1, 2, 4, 6, 8$ for the parameter O_2, which determines the number of components K in the mixture, and values $0.1, 0.5, 1$ for τ to obtain approximate categorical samples [6]. For the activation function $\mathbf{g}$, we experimented with the *tanh* and *sigmoid* functions. The final hyperparameters can be found at.[4]

All of our experiments were executed on Google Colab[5]. Unlike more computationally expensive techniques, such as BERT and XLM-RoBERTa, MVAE-BM can be trained within a few minutes on platforms that provide public virtual machines. Its implementation, based on neural networks, is also suitable for parallelization via GPU/TPU.

4.2 Document modeling

Here we evaluate the likelihood of documents left-out of the training set according to the model, using the perplexity metric. Perplexity measures how poorly

[4] https://github.com/brunoguilherme1/MVAE-BM/tree/main/hyperparameters
[5] https://colab.research.google.com

Table 2: Document modeling: perplexity values. (The latent dimension is indicated in parenthesis, and results not available in the original papers by dashes)

	20NewsGroups		*RCV1*	
Model	(50)	(200)	(50)	(200)
LDA	1,091	1,058	1,437	1,142
RSM	953	836	988	—
DocNade	836	—	742	—
GSM	787	829	717	602
fDARN	917	—	724	598
NVDLA	1,073	993	791	797
NVDM	836	852	563	550
NTM-R	775	763	—	—
NB-NTM	740	—	—	—
iTM-VAE-Prod	—	779	—	508
MVAE-BM	**730**	**740**	**550**	**504**

a probability model predicts a sample (lower is better), and is widely used with language models to measure their capacity to represent documents. Perplexity is defined as $\exp(-\frac{1}{D}\sum_{i=1}^{D}\frac{\log p(x_i)}{|x_i|})$, where D is the number of documents, and $|x_i|$ is the number of words in the document x_i. Following previous approaches, the variational lower bound (ELBO) is used to estimate $p(x_i)$ (which is actually an upper bound on perplexity[20]). A low perplexity indicates the model is good at predicting a given corpora.

Table 2 presents the perplexity metric of document modeling in *20NewsGroups* and *RCV1-v2*, for latent variable dimensions 50 and 200 (shown as separate columns), for MVAE-BM and for 10 baselines: LDA [12], NVLDA [19], GSB [11], NVDM [12], NB-NTM [22], RSM [20], DocNADE [12], fDARN [20], SBN [12], NTM-R [5] and iTM-VAE-Prod [15]. These baselines represent a variety of techniques for topic modeling, some based on graphical models (LDA and RSM) and some based on *belief networks* and on deep networks (DocNADE, SBN, fDARN, NVDM).

MVAE-BM achieves the lowest perplexity values among all baselines in both datasets. Compared to the graphical models, MVAE-BM with a latent variable of dimension $H = 50$ in *RCV1-v2* performs even better than some baselines with 200 dimensions, which is likely due to the interaction between $\boldsymbol{c}$ and $\boldsymbol{h}$, indicating that using $\boldsymbol{c}$ as an additional latent representation is more effective than increasing H.

4.3 Classification based on learned representations

We now turn our attention to the task of text classification, using the representations learned by different models. In this *supervised* experiment the performance of MVAE-BM is compared against baselines based on VAE models: CNN-VAE

Table 3: Document classification: models' accuracy (%). Baselines' results were transcribed from reference papers (dashes denote absent values).

Model	*Yahoo Answers*	*Yelp Reviews*	*Subjectivity*	*TwitterHate*
SCNN-VAE-Semi	65.0	52.0	—	—
CVAE	18.7	29.2	—	—
CVAE BoW	58.5	45.5	—	—
Dirichlet VAE	51.5	39.2	—	—
Dirichlet VAE BoW	59.0	46.3	—	—
BERT	67.6	52.5	87.7	78.2
RoBERTa	66.6	53.0	86.5	77.5
XLM-RoBERTa	69.2	52.5	76.2	74.2
DistilBERT	**70.1**	52.3	88.2	80.3
MVAE-BM	66.5	**55.3**	**89.2**	**82.3**

[25] and Dirichlet-VAE [24] and on deep learning (Transformer) architectures: BERT [4], RoBERTa [9], XLM-RoBERTa [1], and DistilBERT [17].

The experiment consists of a document classification task on the test set of each dataset, performed by classifiers that were trained with the representations learned by MVAE-BM and the baseline models. We train a *logistic regression classifier* for the classification task. Since our main goal is to develop and evaluate text representations for classification tasks, we used the classifier standard implementation[6] without any optimizations.

Table 3 displays the classification accuracy obtaine by each baseline and by MVAE-BM. For *Yelp Reviews* and *Yahoo Answers*, we transcribed the results of VAE and deep learning baselines from the original papers. For *Subjectivity* and *Twitter*, only the results for Transformers were found.

In *Yelp Reviews*, MVAE-BM has the highest accuracy. In Yahoo Answers, although the Transformer models and, in particular, DistilBERT, perform best, MVAE-BM outperforms the VAE baselines and its accuracy is on par with RoBERTa's. In *Subjectivity* and *TwitterHate* datasets, MVAE-BM achieves the highest accuracy among all the baselines, even though the deep learning models require significantly more computation power.

4.4 Document clustering

In this section we evaluate how MVAE-BM performs at document clustering tasks. In general, automatic labeling can be done by applying any unsupervised method (e.g., K-means[7]) to the embeddings obtained for the documents. MVAE-BM, however, already includes a labeling of the data by means of the latent indicator vector $\boldsymbol{c} = [c_1, c_2, \ldots, c_K]$, defined in Eq. (2). Since $\boldsymbol{c}$ is approximately a one-hot vector, it can be interpreted as a clustering of the input into K groups.

[6] www.sklearn.com
[7] https://github.com/UKPLab/sentence-transformers#clustering

We aim to compare the quality of the clusters defined by MVAE-BM against those found by applying K-means to the text representations obtained using Transformer models.

Table 4 exhibits the results measured w.r.t. the Silhouette [18], the Calinski-Harabasz [21] and the Davies-Bouldin [3] clustering quality measures. We set MVAE-BM's and the baselines's hidden dimension h to 1024 and the number of clusters K in MVAE-BM and in K-means to the number of classes of each dataset (Table 1). The proposed model achieves the highest quality scores in almost all of the combinations (dataset, measure). In particular, in some cases (*Yahoo Answers*, *Yelp Reviews* and *TwitterHate*), the Silhouette score for MVAE-BM is one or two orders of magnitude higher than the baselines'.

Table 4: Clustering Score: SI (Silhouette), DB (Davies-Bouldin) and CA (Calinski-Harabasz). K-means used to cluster BERT variants' embeddings.

	TwitterHate			Subjectivity			20News			Yahoo Answers			Yelp Reviews		
	SI	DB	CA	SI	DB	CA	SI	DB	CA	SI	DB	CA	SI	DB	CA
BERT	0.03	3.23	378	0.005	5.4	110	0.11	6.3	623	0.01	10.34	654	0.05	11.69	781
DistilBERT	0.02	3.45	**367**	0.06	4.8	113	0.012	6.1	**589**	0.02	10.89	689	0.02	11.98	769
RoBERTa	0.01	3.43	378	0.05	5.2	112	0.01	6.5	650	0.04	10.45	623	0.018	11.67	720
XLM-RoBERTa	0.06	3.89	389	0.01	5.4	114	0.10	6.3	677	0.07	10.33	**656**	0.07	11.34	754
MVAE-BM	**0.23**	**3.12**	372	**0.15**	**4.1**	**98**	**0.23**	**6.0**	687	**0.33**	**10.01**	698	**0.46**	**11.23**	**712**

5 Conclusion

In this work, we presented MVAE-BM, a mixture of unsupervised latent models for language modeling. MVAE-BM is inspired by the Boltzmann machine and uses modern neural inference techniques to estimate the intractable latent distributions that appear in the model. In our experiments, we compared to more than 15 different baselines. In these tasks , our model outperformed all baselines in 5 of the 6 datasets used in this work. Apart from the performance gains, our model also has the advantage of learning text representations from unlabeled data without requiring pre-trained word embeddings. Those text representations can be applied with success in various learning tasks, including clustering.

References

1. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: ACL (2020)
2. Dahl, G.E., Adams, R.P., Larochelle, H.: Training restricted boltzmann machines on word observations. In: ICML (2012)
3. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-1**(2), 224–227 (1979). https://doi.org/10.1109/TPAMI.1979.4766909

4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
5. Ding, R., Nallapati, R., Xiang, B.: Coherence-aware neural topic modeling. In: EMNLP (2018)
6. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (2017)
7. Jiang, S., Chen, Y., Yang, J., Zhang, C., Zhao, T.: Mixture variational autoencoders. Pattern Recognit. Lett. **128** (2019)
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR (2019), http://arxiv.org/abs/1907.11692
10. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. In: ICLR (2017)
11. Miao, Y., Grefenstette, E., Blunsom, P.: Discovering discrete latent topics with neural variational inference. In: ICML (2017)
12. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: ICML (2015)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS (2013)
14. Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: ICML (2014)
15. Ning, X., Zheng, Y., Jiang, Z., Wang, Y., Yang, H., Huang, J., Zhao, P.: Nonparametric topic modeling with neural inference. vol. 399 (2020)
16. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: EMNLP (2014)
17. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv preprint arXiv:2004.09813 (04 2020)
18. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20** (1987)
19. Srivastava, A., Sutton, C.: Neural variational inference for topic models. In: NeurIPS (2016)
20. Srivastava, N., Salakhutdinov, R., Hinton, G.: Modeling documents with a deep boltzmann machine. In: Conference on Uncertainty in Artificial Intelligence (2013)
21. Sugar, C.A., James, G.M.: Finding the number of clusters in a dataset. Journal of the American Statistical Association **98**(463) (2003)
22. Wu, J., Rao, Y., Zhang, Z., Xie, H., Li, Q., Wang, F.L., Chen, Z.: Neural mixed counting models for dispersed topic discovery. In: Annual Meeting of the Association for Computational Linguistics (Jul 2020)
23. Xiao, Y., Zhao, T., Wang, W.Y.: Dirichlet variational autoencoder for text modeling. CoRR (2018)
24. Xu, J., Durrett, G.: Spherical latent spaces for stable variational autoencoders. In: EMNLP (2018)
25. Yang, Z., Hu, Z., Salakhutdinov, R., Berg-Kirkpatrick, T.: Improved variational autoencoders for text modeling using dilated convolutions. In: ICML (2017)