Undergraduate Research Opportunity Program
(UROP) Project Report

Deep Learning in Topic Modelling

By

Jethro Kuan

Department of Computer Science

School of Computing

National University of Singapore

2018/19

Undergraduate Research Opportunity Program
(UROP) Project Report

Deep Learning in Topic Modelling

By

Jethro Kuan

Department of Computer Science

School of Computing

National University of Singapore

2018/19

Project No:
Advisor: Prof. Kan Min-Yen
Deliverables:
    Report: 1 Volume

Abstract

Traditional approaches of topic modelling suffer from several drawbacks. In this report, we investigate the weaknesses of traditional topic modelling, and look towards deep learning for help. We implement and evaluate lda2vec, a model that learns word vectors jointly with document-level mixtures of topic vectors.

Acknowledgement

I would like to WING NUS for the insightful discussions and feedback throughout my research journey, and for the compute resources necessary to run my experiments.

List of Figures

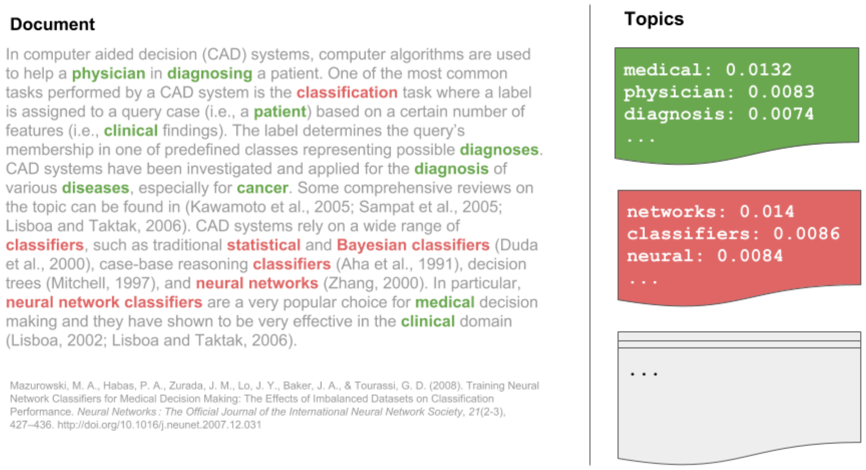
List of Tables

Table of Contents

Chapter 1

Introduction

Topic modelling is a popular statistical modelling technique in the field of Information Retrieval. Topic models are typically trained on an unlabelled corpus of text. The goal of a topic model is to discover abstract topics within in a collection of documents. This facilitates discovery and navigation in a large, unlabelled corpora of documents. Topic models have been successfully employed on historical documents, scientific documents [Boyd-Graber et al., 2017], and also as components of recommendation systems [McAuley and Leskovec, 2013].



Document

In computer aided decision (CAD) systems, computer algorithms are used to help a **physician** in **diagnosing** a patient. One of the most common tasks performed by a CAD system is the **classification** task where a label is assigned to a query case (i.e., a **patient**) based on a certain number of features (i.e., **clinical** findings). The label determines the query's membership in one of predefined classes representing possible **diagnoses**. CAD systems have been investigated and applied for the **diagnosis** of various **diseases**, especially for **cancer**. Some comprehensive reviews on the topic can be found in (Kawamoto et al., 2005; Sampat et al., 2005; Lisboa and Taktak, 2006). CAD systems rely on a wide range of **classifiers**, such as traditional **statistical** and **Bayesian classifiers** (Duda et al., 2000), case-base reasoning **classifiers** (Aha et al., 1991), decision trees (Mitchell, 1997), and **neural networks** (Zhang, 2000). In particular, **neural network classifiers** are a very popular choice for **medical** decision making and they have shown to be very effective in the **clinical** domain (Lisboa, 2002; Lisboa and Taktak, 2006).

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance. *Neural Networks : The Official Journal of the International Neural Network Society*, *21*(2-3), 427–436. http://doi.org/10.1016/j.neunet.2007.12.031

Figure 1.1: According to the trained topic model, topics like "Medicine" and "Neural Networks" generate the colored words of the document

The most popular topic modelling technique today is a simple probabilistic topic model called the Latent Dirichlet Allocation (LDA). Deep learning has revolutionized many information retrieval tasks such as image retrieval and question answering, but advancements in the field has not changed topic modelling approaches significantly.

This motivates our research direction: we seek to investigate the areas where current topic models fall short, and look to deep learning to attempt to fill in the gaps.

1.1 Our Contributions

We survey traditional topic modelling approaches, present their weaknesses. Following which, we survey neural approaches to topic modelling. We study one of them in particular: lda2vec [Moody, 2016].

Moody proposes lda2vec from a neural perspective, while we consider the motivations for the lda2vec model from the standpoint of improving on traditional approaches to topic modelling. We implemented lda2vec, and trained it on a subsample of the 20 Newsgroups dataset. Finally, we discover some of the difficulties in training the model, and suggest methods to improve the model.

1.2 Report Organization

In chapter 2, we will briefly explain how the design and training process for probabilistic topic models. Next, in chapter 3 we summarize the research done in neural approaches to topic modelling. This leads us to lda2vec, which we detail in chapter 4. We implement lda2vec, and present our results in chapter 6. Finally, in chapter 7, we conclude and discuss possible future work in this direction.

Chapter 2

Analysing Probabilistic Topic Model Approaches

Since probabilistic topic models are the more popular approach, we focus on this class of topic models for our analysis. In this section, we will give some background on the process of building a probabilistic topic model. For a more comprehensive survey, refer to our literature review[1].

We shall discuss probabilistic topic models using Latent Dirichlet Allocation (LDA) [Blei et al., 2003] as a specific example, but the conclusions we draw will be general.

2.1 Designing a Topic Model

In probabilistic topic models, documents are a distribution over topics, and topics are a distribution over words. Designing a probabilistic topic model involves designing the generative process of for each document in the corpus. The generative process for Latent Dirichlet Allocation (LDA) is shown in Algorithm 1.

The generative process can be represented as probabilistic graphical models, in plate notation (Figure 2.1). We note that in the generative process, the words in the document are observed, while the topic and document distributions are both latent.

The Dirichlet distribution encourages sparsity, encoding the belief that the document-topic

[1] `https://github.com/jethrokuan/lda-survey`

Algorithm 1 Generative Process of LDA

```
for each document $d$ in $D$ do
    Draw topic distribution $\theta_d \sim Dir(\alpha)$
    for each word $n_d$ in $N_d$ do
        Sample topic $z_{d,n} \sim Multinomial(\theta_d)$
        Sample word $w_{d,n} \sim Multinomial(\beta_{z_{d,n}})$
```

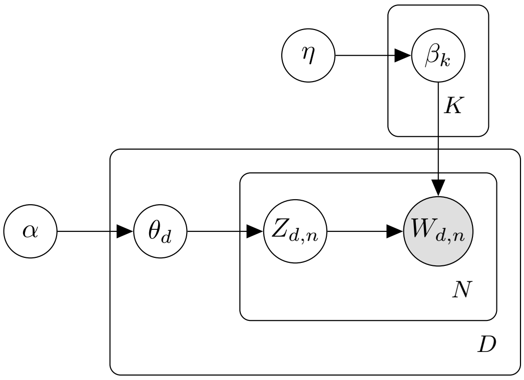


Figure 2.1: Plate notation for LDA.

distribution has few topics per document, and the topic-word distribution has few words per topic. These two beliefs work against each other, and LDA discovers this sparsity balance, which gives rise to the structure of the textual data.

Some models are also capable of utilising side-information, in addition to the text in the corpora. These models have to incorporate these information in the generative process. An example of such a model would be the author-topic model (ATM) [Rosen-Zvi et al., 2004], where each author is also associated with a distribution of topics (Algorithm 2). Dynamic Topic Modelling (DTM) attempted to model the time evolution of topics by allowing the topic distribution parameters and topic proportions to evolve over time [Blei and Lafferty, 2006b]. Each of these models have a larger number of latent variables to learn.

2.2 Learning a Topic Model

In probabilistic topic modelling, we wish to solve for the hidden variables in the generative process. We do so through *posterior inference* – computing the posterior distribution of the

Algorithm 2 Generative Process of ATM

```
for each document $d$ in $D$ do
    for each word $n_d$ in $N_d$ do
        Draw author $x$ uniformly among document authors $a_d$
        Draw topic distribution $z$ from author's distribution $\theta_x$
        Sample word $w_{d,n} \sim Multinomial(\beta_{z_{d,n}})$
```

hidden variables given the observed words in a document:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{2.1}$$

The posterior is computationally intractable, hence approximate posterior inference methods must be used. The two main families of inference methods are Markov Chain Monte Carlo (MCMC) methods and Variational Inference.

Variational inference provides an analytical approximation for the posterior distribution. In variational inference, the posterior distribution over a set of unobserved variables $p(Z|H)$ is approximated by a distribution $q(Z)$, selected to be in a family that can approximately model the true posterior. Inference is performed by minimizing the distance between $p(Z|H)$ and $q(Z)$. One common metric used is the KL-divergence. It is also often able to find approximate solutions quicker than MCMC methods. However, deriving the equations for parameter updates is often much more difficult, and requires appropriate choices of distributions for the generative process. For LDA, mean-field variational inference is simple, due to the model simplicity, and that the prior and posterior distributions are conjugate distributions.

On the other hand, MCMC methods provide numerical approximations for the posterior distribution. In MCMC, we construct an ergodic Markov chain on the latent variable z, whose stationary distribution is the posterior $P(z|x)$. Samples can be treated as drawn from the stationary distribution, and used to approximate the posterior empirically.

Gibbs sampling is a popular sampling method adopted in probabilistic topic models. In Gibbs sampling, the space of the Markov Chain is the space of the configurations of the hidden variables. The next state is reached by sequentially sampling all variables from the distribution, conditioned on all the current sampled values. After a "burn-in" period, the samples would be drawn from the posterior distribution.

Algorithm 3 Gibbs Sampling

```
$x^0 \leftarrow q(x)$
for $i = 1, 2, 3, \ldots$ do
    for $d = 1, 2, 3, \ldots, D$ do
        $x_d^i \sim P(X_1 = x_1 | X_k = x_k^{i-1}$ for $k = \{1..n \setminus d\})$
```

Gibbs sampling suffers from several drawbacks. First, it is possible that samples drawn are very correlated, and it might take a long time to reach its stationary state (long "burn-in" period). MCMC methods are often used when deriving the equations for variational inference is difficult. MCMC methods will also exactly approximate the target distribution in the long run.

Hence, variational inference is suited to large datasets and models which have simple equations for parameter updates, and MCMC methods are suited for smaller datasets where the distributions need to be approximated with greater precision, at a computational cost.

2.3 Preparation of Corpus

After designing a topic model and choosing an inference method, we need to prepare our corpus. Below is a list of common pre-processing techniques used in topic modelling.

Stopword Removal

Words that are expected not to contribute to any meaningful topics are removed. Stopword lists can be constructed manually to suit a dataset, or can be generic lists that include words such as "I" and "we", found in popular text processing libraries.

Stemming

Related words generally map to the same stem. Hence, stemming is seen as a method to reduce vocabulary size, while minimizing the effects on topic inference.

Bag-of-words

> The bag-of-words processing simplifies the representation of a document, ignoring word order.

The combinations of text-preprocessing methods to use have not been standardized across the NLP community. While some amount of research is being done to recommend standard practices for text pre-processing techniques Schofield et al. [2017], it is often the case that the best combinations are subject to the choice of topic model and the dataset used.

2.4 Weaknesses of Traditional Topic Model Approaches

Armed with a background knowledge of probabilistic topic models, we can now begin to discuss their weaknesses.

2.4.1 Difficulty in incorporating side-information

It is often the case that various kinds of meta information are available at the document level. For example, scientific documents have author, conference and year metadata. These side-information provide useful information that can aid the inference of meaningful topics. Incorporating side-information requires their addition in the generative process, which quickly builds complexity. It is difficult to derive the update equations for variational inference in complex models, and MCMC sampling becomes increasingly computationally expensive. It is also difficult to jointly model various meta information in the same model. MetaLDA proposed a way to efficiently leverage arbitrary document and word meta information, but only in a binary form [Zhao et al., 2017].

2.4.2 Making Statistical Assumptions in Models

Embedded within the generative process are strong statistical assumptions. For example, in LDA, the choice of the Dirichlet distribution creates a strong and unrealistic model assumption that the presence of one topic is not correlated with the presence of another [Blei and Lafferty, 2006a]. Statistical assumptions significantly reduce the model hypothesis space. However, we must be careful in choosing and recognising the assumptions made, and whether the hypothesis space does indeed contain a plausible model for the given data.

2.4.3 Removal of useful information in pre-processing

Each pre-processing step removes information which could be useful in topic inference. This is especially harmful in already resource-scarce text corpora: topic models suffer from a large performance degradation over short-text corpora such as tweets and news headlines, because of insufficient word co-occurrence information.

Using the bag-of-words representation removes all word-level information. We have already observed in deep learning that word co-occurrence within a window can contain enough information to learn word vectors with semantic meaning [Mikolov et al., 2013]. Indeed, using word embeddings in topic models have proven useful in learning meaningful topics from short texts [Das et al., 2015] [Qiang et al., 2017].

Chapter 3

Neural Approaches to Topic Modelling

In the previous chapter, we described probabilistic topic models and some of their weaknesses. In this chapter, we describe how neural approaches to topic modelling solve the identified weaknesses.

3.1 Efficient Learning Algorithms

In subsection 2.4.1, we mentioned that a significant bottleneck to creating complex topic models is the intractability of training them. This is an issue that plagues probabilistic graphical models in general, and there is significant interest in developing efficient inference algorithms for directed probabilistic graphical models.

Auto-encoding Variational Bayes (AEVB) was introduced as an estimator of the variational lower bound [Kingma and Welling, 2013]. Fitting an approximate inference model to the intractable posterior using the lower bound estimator can be done through gradient descent, and using universal function approximators like neural networks allows this method to apply to many models. In addition, AEVB learns an inference method that maps a document to an approximate posterior distribution, without the need to run further updates. Neural networks are especially efficient at inference, unlike probabilistic topic models that have expensive inference steps.

AEVB has been found to be difficult to apply to topic models in practice because of 2 reasons. First, the Dirichlet prior is not a location scale family, which hinders reparameterisation, and second the inference network can get stuck in a bad local optimum in which all topics are identical [Srivastava and Sutton, 2017]. Autoencoded Variational Inference for Topic Models (AVITM) resolves these issues by collapsing variables in the posterior, using a laplace approximation to the Dirichlet prior, generating interpretable topics with a lower training time and is capable of fast inference [Srivastava and Sutton, 2017].

While this research shows promise, probabilistic graphical models still lack the capability of incorporating arbitrary side-information. Since the generative processes for these topic models are simple and still tractable through variational inference or MCMC methods, using these black-box inference methods show minimal improvement.

3.2 Removing Statistical Assumptions

In subsection 2.4.2, we've stated that the design process of graphical models involve making statistical assumptions. We've also noted that the Dirichlet prior is a source of difficulty in using black-box inference methods like AEVB. We return to a more fundamental question: why is the Dirichlet prior the prior to use? Neural approaches resolve this issue by considering a hypothesis space that contains all possible functions, since even small neural networks can be universal approximators. By switching to neural approaches, it is no longer necessary to design the generative process and inference methods around the chosen priors.

3.3 Learning from Word Sequences

In subsection 2.4.3, we noted that probabilistic topic models like LDA discard all word-local information by converting the documents into the bag-of-words representation. Neural approaches resolve this by using Recurrent Neural Networks (RNNs), neural network architectures that are particularly suited towards sequences. LSTM-Topic matrix factorization model (LTMF) uses

Long-Short Term Memory (LSTM) units in conjunction with Probabilistic Matrix Factorization, a simple topic modelling technique, to produce context-aware recommendation systems [Jin et al., 2018]. Latent LSTM allocation (LLA) [Zaheer et al., 2017] modifies the generative process to use an LSTM to include the index of the word in a document, effectively using the word sequence information to learn topics, as shown in Algorithm 4.

Algorithm 4 Generative Process of LLA

```
for each document $d$ in $D$ do
    Initialize LSTM $s_0 = 0$
    Draw topic distribution $\theta_d \sim Dir(\alpha)$
    for each word index $t$ from 1 to $N_d$ do
        Update $s_t = LSTM(z_{d,t-1}, s_{t-1})$
        Get topic proportions at time $t$ from LSTM $\theta = softmax_K(W_P s_t + b_p)$
        Sample topic $z_{d,n} \sim Multinomial(\theta)$
        Sample word $w_{d,n} \sim Multinomial(\beta_{z_{d,t}})$
```

LSTMs have a large number of parameters, and are unsuited where the corpora is relatively small. LTMF is trained on Amazon user reviews, and LLA is trained on document user history of Wikipedia. Both of these corpora are sufficiently large to learn the parameters for the LSTMs. Since our experiments involve a small dataset, using LSTM architectures would not be wise.

3.4 Incorporating Side-information

In subsection 2.4.1, we have argued for the need for models that can efficiently incorporate side-information. In the earlier section, we have shown research that incorporates word-sequence information. However, there is little research into being able to efficiently incorporate arbitrary meta information for learning meaningful topics. To this end, we've only seen lda2vec [Moody, 2016] and the very similar Topic2Vec [Niu and Dai, 2015]. lda2vec looks to be a promising model that resolves all the issues we've discussed thus far.

First, lda2vec can be easily extended to incorporate arbitrary side information, without designing a specific generative process and corresponding graphical model. This is achieved through a framework of combining embeddings for different meta information, and learning them via gradient descent. This is explained in further detail in subsection 5.0.2. Second, lda2vec has a comparatively small number of parameters, and should be effective with small datasets. Third, lda2vec also uses word-local information by training on word co-occurences within a sliding window, achieving the same effect as the LSTM architectures in section 3.3.

Finally, lda2vec opens up possibilities of multi-modal topic modelling. Probabilistic topic models have difficulty representing other modes of media, such as audio and images. On the other hand, Deep learning has been very successful in generating low-dimensional representations of images, audio and text. For example, one could use a typical convolutional neural network like Resnet to obtain representations of images with semantic meaning.

Chapter 4

lda2vec

In the previous chapter, we discussed several neural approaches to topic modelling, and concluded that lda2vec would be most suitable for our experiments. In this chapter, we describe lda2vec. To minimize repetition, we only provide a brief introduction to lda2vec. For a more detailed treatment, refer to Moody [2016].

4.0.1 word2vec

lda2vec is an extension of the word2vec model. word2vec is a model that learns word embeddings using a shallow 2-layer neural network [Mikolov et al., 2013]. The skip-gram word2vec model is given the task of predicting context words given a target word. Training examples are produced by generating word pairs within a sliding window.

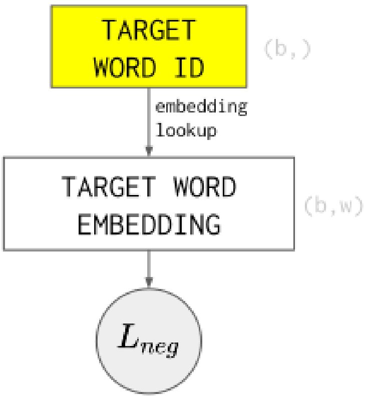


Figure 4.1: The word2vec model trained with batch size b, and embedding size of w.

For illustration, we look at the sentence: “the quick brown fox jumps over the lazy dog”, and look at the training examples produced. Suppose we choose a window size of 2.

We begin with the first target word “the”, and we want to predict words that occur at most 2 words away from the target. Hence, we form the training examples “(the, quick)” and “(the brown)”. We then slide the window forward, looking at the target word “quick”. Similarly, we produce the training examples “(quick, the)”, “(quick, brown)”, and “(quick, fox)”. This process repeats until all possible target words are processed.

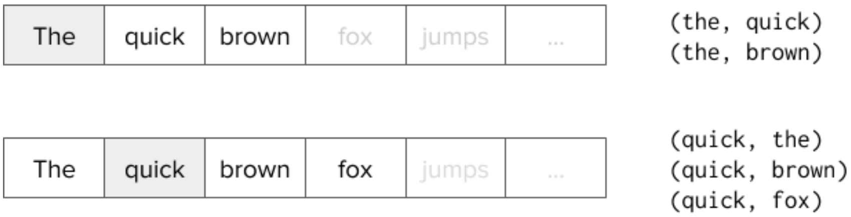


Figure 4.2: The first 5 training examples generated from the sentence “The quick brown fox jumps over the lazy dog.” Here, the window size is 2.

The objective of the word2vec network is the negative sampling loss:

$$L_{ij}^{neg} = \log \sigma(w_j \cdot w_i) + \sum_{l=0}^{n} \log \sigma(-w_j \cdot w_l) \tag{4.1}$$

For each target word w_j, context word w_i, and the negatively sampled words w_l.

These word embeddings exhibit arithmetic properties: where words with small cosine distance are similar.

Chapter 5

Extending word2vec

In lda2vec, document and topic embeddings are jointly learnt with word embeddings. Figure 5.1 illustrates how the portion of the lda2vec neural network architecture containing the document and topic embedding matrices.

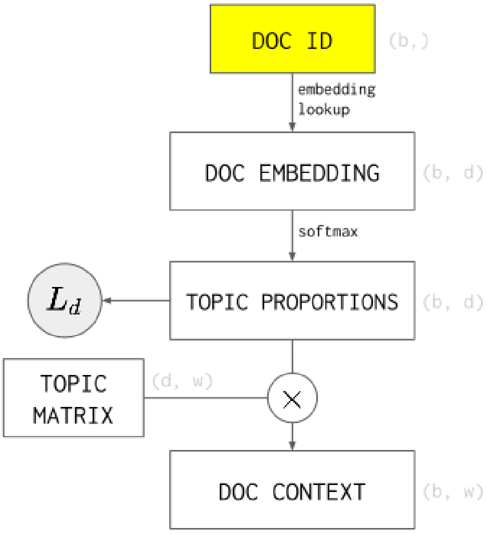


Figure 5.1: The portion of the lda2vec neural network architecture containing the document and topic embedding matrices. The document matrix has shape (b, t), where t is the number of topics. The topic matrix has shape (t, w), so the the resultant document context has the same shape as the word embedding.

Each training example is obtained from a document, with a corresponding document id. We use this document id to obtain the document embedding. The softmax function transforms the document embedding into topic proportions, such that the sum of the weights in each embedding dimension would sum to one. To obtain the final embedding representing the document, we perform a matrix multiplication of the topic proportions with the topic matrix.

Similar to LDA, we want to encourage sparsity in the topic proportions. This is achieved through an objective inspired by the Dirichlet:

$$L_d = \sum_{jk} (\alpha - 1) \log p_{jk} \tag{5.1}$$

L_d measures the likelihood of document j in topic k summed over all available documents. A low concentration value α encourages the topic proportions coupling in each topic to be sparse. The topic proportions are initialized to be relatively homogeneous, and the model learns sparser, more concentrated topic proportions over time.

5.0.2 Learning document, topic and word representations

To jointly learn the document, topic and word representations, we combine the document context embedding and word embedding, and modify the negative sampling loss to use the context embedding $\vec{c_j}$ instead of the target word embedding $\vec{w_j}$.

$$L_{neg} = \log \sigma(\vec{c_j} \cdot \vec{w_i}) + \sum_{l=0}^{n} \log \sigma(-\vec{c_j} \cdot \vec{w_l}) \tag{5.2}$$

Hence, combined with the Dirichlet regularization term, we obtain the final loss:

$$L = L_d + \lambda L_{neg} \tag{5.3}$$

where λ is a term modulating the weight of the regularization term.

This describes the entire lda2vec model, which is depicted in Figure 5.2.

5.1 Parameters

There are several parameters that can be tweaked to alter the model behaviour, and here we summarise them.

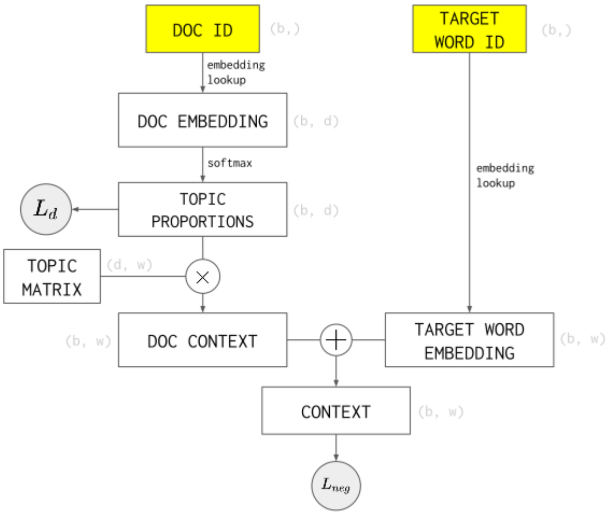


Figure 5.2: The lda2vec neural network architecture.

λ modulates the weightage of the Dirichlet regularization term. α is the Dirichlet parameterization term, and for topic models, one should set $\alpha < 1$. We also choose the number of negative samples l, when computing the sampled negative sampling loss L_{neg}. We choose the embedding size w for the word, topic and context vectors. Finally, we choose the mini-batch size b. To optimize GPU usage and offset the I/O time of transfer of data between the CPU and GPU, a large batch size should be used.

Chapter 6

Evaluation

Our first objective is to evaluate the quality of topic models generated by lda2vec. We want to evaluate if utilising word-local information would help infer more meaningful topics.

We had also planned to train lda2vec on a corpus of scientific documents, including author information, jointly learning author embeddings in addition to word, document and topic embeddings. This would fully exploit the ease of incorporating side-information into lda2vec. Because of various difficulties faced, this work has been deferred and left as future work.

The code for reproducing the results of our experiments is available online [1].

6.1 Preparation of Data

We use the well-established, coherent 20 Newsgroups dataset. To keep experiments quick, we obtained a small sub-sample of the dataset by choosing 3 categories within the 20 Newsgroups dataset:

1. soc.religion.christian
2. sci.electronics
3. comp.windows.x

This subset contains 1783 documents.

[1] `https://github.com/jethrokuan/lda2vec/`

We use nltk [2] to tokenize the data. We perform no further text-preprocessing, as suggested by Schofield et al., we defer the text-treatment to post-processing, when we compute the top words for each topic.

To prepare the data for lda2vec, we need to produce the skipgram training examples. Similar to Mikolov et al., we extract target and context words using a sliding window. We chose the window size to be 5.

6.2 Implementation Details

To train a topic model for LDA, we use implementation from the scikit-learn [3]. This implementation uses an online variational Bayes method to update the variational parameters for the topic word distributions, described in Hoffman et al. [2010].

We implemented lda2vec using Tensorflow. The implementation of lda2vec differs slightly from the paper, where we've made iterative improvements based on observations of the generated topics.

6.3 Evaluation Methods

To evaluate our topic models, we compute topic coherence using the averaged Normalized Pointwise Mutual Information (C_v) as our metric.

Topic coherence is computed given by averaging the pairwise similarity between words in a topic. Newman et al. suggested that the higher the average pairwise similarity, the more coherent the topic.

$$Coherence(T) = \frac{\sum sim(w_i, w_j)}{\binom{n}{2}} \tag{6.1}$$

for each $w_i, w_j \in T$. Here, we choose our similarity measure between words to be the cosine similarity measure:

[2] https://www.nltk.org/

[3] http://scikit-learn.org/

$$sim_{cos}(w_i, w_j) = \frac{w_i \cdot w_j}{|w_i||w_j|} \tag{6.2}$$

A word vector space is constructed from the Wikipedia corpora using the NPMI measure as per Aletras and Stevenson, where:

$$PMI(w_i, w_j) = \frac{p(w_i, w_j)}{-\log(p(w_i, w_j))} \tag{6.3}$$

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log(p(w_i, w_j))} \tag{6.4}$$

We compute C_v, which averages the NPMI for each pair of words within a sliding window of size 110 on an external corpus [Röder et al., 2015].

Prior to this work, we spent a significant amount on time looking into evaluation methods for topic models, in an attempt to develop better intrinsic statistical evaluations for topic models. We have appended a short write-up on this topic in A.

6.4 Experiment Results

6.4.1 LDA

For each experiment, we trained LDA for 1000 iterations.

First, we trained LDA on the subsampled dataset. The results are shown in Table 6.1. The topics learned have poor topic coherence scores. We hypothesise that the subsampled dataset contained too few documents, and did not have enough information to learn meaningful topics.

Training LDA on the full dataset (Table 6.2), we see that the learned topics had a much higher topic coherence score, and appears to be much more cohesive.

6.4.2 lda2vec

For lda2vec, the training examples are skip-gram training examples (w_j, w_i), and these need to be obtained through further pre-processing. We use a window size of 5 on the sub-sampled data

Topic	0	1	2	3	4	5	6	7	8	9
	recently	hi	like	trying	need	looking	ve	don	just	does
	used	question	new	think	faq	want	try	know	let	know
	dear	use	article	yes	time	people	got	read	tell	hello
	sun	problem	com	god	apr	point	seen	sorry	sure	good
	having	th	christians	posted	archive	post	posting	xt	right	deleted
	subject	just	discussion	actually	ago	lot	motif	really	come	anybody
	comp	version	dec	write	93	true	reply	say	idea	folks
	simple	window	did	run	answer	send	wi	yo	running	way
	says	info	argument	wrong	probably	build	look	set	ok	wondering
	circuit	source	hp	code	help	suggest	heard	recent	asked	stuff
C_v	0.3680	0.3407	0.3448	0.2757	0.3060	0.3066	0.5533	0.3117	0.3821	0.3110
Average	0.3500									
Median	0.3262									

Table 6.1: The learned topics and their coherence scores with LDA on the subsampled dataset.

Topic	0	1	2	3	4	5	6	7	8	9
	edu	don	god	thanks	g9v	key	10	game	people	file
	space	just	people	use	b8f	use	00	year	government	drive
	com	like	does	window	a86	chip	15	team	mr	card
	information	know	jesus	windows	145	encryption	25	games	gun	disk
	available	think	believe	does	1d9	used	12	play	law	scsi
	mail	good	think	know	pl	keys	20	season	president	dos
	data	time	say	help	0t	clipper	11	hockey	armenian	mac
	ftp	people	don	like	cx	bit	16	league	said	pc
	pub	ve	just	program	34u	bike	14	players	state	memory
	send	going	know	using	2di	number	13	win	israel	use
C_v	0.3607	0.2991	0.2993	0.3505	0.2985	0.4479	0.4420	0.569	0.41310	0.4499
Average	0.3931									
Median	0.3869									

Table 6.2: The learned topics and their topic coherence scores with LDA on the full 20 Newsgroups dataset.

Topic 0	OOV, also, father, time, would, people
Topic 1	OOV, also, father, time, would, church
Topic 2	OOV, also, father, time, would, church
Topic 3	OOV, also, father, time, people
Topic 4	OOV, also, father, time, people
Topic 5	OOV, also, father, time, people, would
Topic 6	OOV, also, father, time, would, people
Topic 7	OOV, also, father, time, people
Topic 8	OOV, also, father, time, people
Topic 9	OOV, also, father, time, people

Table 6.3: The learned topics are homogeneous. Note that the number of items in the topic is low because we do post-processing stop-word removal.

set to obtain 48668 word-pairs. Each word-pair is annotated with a document id, identifying the source document. The tuple (w_j, w_i, d_k) is passed as training examples to lda2vec.

For our experiments, we choose $w = 200$, $\lambda = 200$, $\alpha = 0.7$, and $l = 15$. Each model is trained for 100000 steps on a batch-size of 512, which in the sub-sampled dataset corresponds to about 1000 epochs. We perform gradient descent using the Adam optimizer.

First, we trained a baseline model described in Moody [2016]. Multiple training runs of the baseline model revealed that:

1. The model learns the topics quickly after several steps, despite not having observed the entire document corpora.

2. The topics learnt were highly homogeneous Table 6.3.

Upon further investigation, we find that if we jointly learn word, topic and document embeddings on randomly initialized word embeddings, the model is unable to learn word embeddings with semantic meaning. The negative sampling loss L_{neg} encourages the topic embeddings to become homogeneous, such that it becomes a non-factor in the overall cost function.

Topic	0	1	2	3	4	5	6	7	8	9
	creator	correct	god	cool	sea	fall	whatever	eat	chance	human
	eat	say	choices	nobody	keep	grave	doubts	much	human	note
	asked	son	refuse	whereas	specific	refuse	work	known	afraid	rest
	maybe	unfortunate	maybe	teaches	believe	call	hand	race	detail	person
	name	perhaps	mind	doubts	another	much	directly	open	music	groups
	acting	meat	death	also	master	peace	concerned	new	hand	files
	rest	gods	forever	forever	fall	going	paul	serve	ever	forever
	eyes	sin	friends	worry	caused	though	least	ends	son	short
	romans	considerations	people	romans	day	anyway	jesus	choose	considerations	day
	control	anyway	grave	assume	judge	day	sabbath	opinion	content	others
C_v	0.4237	0.2890	0.3880	0.2877	0.4057	0.3739	0.3300	0.4876	0.3054	0.4646
Average	0.3697									
Median	0.3739									

Table 6.4: Topic coherence scores for lda2vec after tweaking and using pretrained embeddings.

We compare the results with the LDA model, and hypothesise that on this small sub-sampled dataset, there is not enough data to learn word representations with semantic meaning, especially given the additional joint task of learning topic embeddings and document proportions.

We achieve better results by adding regularization to the model. We add dropout on the word and document embeddings. We set the dropout probability to 0.5. We also added a loss-switch mechanism to allow word vectors to first learn semantic representations. This mechanism controls the loss as follows:

$$L = \begin{cases} L_{neg} & \text{if } global\_step \leq \text{n} \\ L_{neg} + \lambda L_d & \text{otherwise} \end{cases} \tag{6.5}$$

We set the switch loss step to 10000 (out of the total steps of 100000).

Finally, we utilise a technique common in deep learning: transfer learning. We load GloVe pre-trained word embeddings of size 200[4].

We summarize the topic coherence scores in Table 6.5. Scores were not calculated for the baseline lda2vec model because there were too few words in the topics generated after post-processing stop-word removal.

[4] The GloVe pretrained embeddings were obtained at `https://nlp.stanford.edu/projects/glove/`

Model	Average	Median
LDA	0.3500	0.3262
lda2vec improved + pretrained	**0.3697**	**0.3739**

Table 6.5: Summary of the C_v scores for the 2 models on the sub-sampled 20 Newsgroups dataset.

6.5 Comparing lda2vec to LDA

First, we note that the number of training examples (word-pairs) for lda2vec quickly increases with the number of documents and the length of the document. In addition, to learn good document proportions, it is natural that the model needs to train for several epochs. Hence, training lda2vec on large datasets takes a long time.

We also notice that LDA took a shorter time to infer meaningful topics as compared to lda2vec. This can be attributed the explicit modelling of topic and documents in the generative process by LDA. In contrast, lda2vec implicitly learns topics and document proportions through the auxiliary task of predicting the context word given the target word.

In addition, lda2vec is a small model that is difficult to optimize for the GPU. The bulk of the time is spent on embedding lookup operations, which are not GPU-optimized operations. Hence, if the model does not use much additional side-information, it would not benefit greatly from computing on GPUs.

On the other hand, the transfer learning we had done is unique to neural network architectures, and cannot be done in traditional probabilistic models. This shows promise for neural approaches to topic modelling, especially in resource-scarce corpora.

Chapter 7

Conclusion

In this paper, we analysed probabilistic topic models, and detail the weaknesses in these approaches. This motivates lda2vec, which we implement and evaluate against LDA. We were unable to train the lda2vec model presented in the paper, but was able to produce results with some tweaks. Finally, the use of pretrained embeddings allowed lda2vec to learn more meaningful topics compared to LDA.

7.1 Future Work

First, we've only used document id as additional side-information to learn topic proportions. Many text corpora come with more metadata that can be useful for learning meaningful topics. For example, scientific documents often come annotated with authorship, year of publication, as well as the conference they are published at. We can use lda2vec to achieve state-of-the-art automatic topic coherence scores on corpora with large amounts of additional metadata, or perform ablation studies on each metadata to explore their importance in learning meaningful topics.

Second, we can explore using the topic-enhanced word embeddings learned as a side-product of training a lda2vec model. The arithmetic properties of these embeddings can be directly exploited to produce recommendation systems. These pretrained embeddings can also be used in neural network architectures for other NLP tasks, such as language-modelling.

Finally, The Dirichlet regularization term L_d is a neat concept, that to our knowledge was first proposed by Moody. We think this is an interesting way of enforcing interpretability in deep learning models, and can be easily applied in other neural network architectures.

References

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22, 2013.

David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006a.

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006b.

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

Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296, 2017.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *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)*, volume 1, pages 795–804, 2015.

Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

Mingmin Jin, Xin Luo, Huiling Zhu, and Hankz Hankui Zhuo. Combining deep learning and topic modeling for review understanding in context-aware recommendatio. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1605–1614, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Christopher E Moody. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*, 2016.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

Liqiang Niu and Xin-Yu Dai. Topic2vec: Learning distributed representations of topics. *CoRR*, abs/1506.08422, 2015. URL `http://arxiv.org/abs/1506.08422`.

Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. Topic modeling over short texts by incorporating word embeddings. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 363–374. Springer, 2017.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

Alexandra Schofield, Måns Magnusson, and D Mimno. Understanding text pre-processing for latent dirichlet allocation. In *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, volume 2, pages 432–436, 2017.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112. ACM, 2009.

Manzil Zaheer, Amr Ahmed, and Alexander J Smola. Latent lstm allocation: Joint clustering and non-linear dynamic modeling of sequence data. In *International Conference on Machine Learning*, pages 3967–3976, 2017.

He Zhao, Lan Du, Wray Buntine, and Gang Liu. Metalda: a topic model that efficiently incorporates meta information. *arXiv preprint arXiv:1709.06365*, 2017.

Appendices

Appendix A

Evaluation Methods for Topic Models

The unsupervised nature of topic models makes model selection difficult, hence evaluating topic models becomes a tricky issue. In some cases, applications may have extrinsic tasks that utilize topic models, such as information retrieval or document classification, and these extrinsic tasks can provide a means of evaluation.

Since we do not have an extrinsic task for utilising the topic models obtained from LDA or lda2vec, we have to rely on an intrinsic measure that measures the generalization capability of a topic model. LDA is typically evaluated by measuring the probability of unseen held-out documents, or estimating the probability of the second half of a document, given the first half [Wallach et al., 2009]. Efficient and unbiased sampling methods for approximating these intrinsic measures have been thoroughly researched by Wallach et al..

However, statistical evaluation measures are not applicable to lda2vec. This is because unlike in LDA, where topics are probability distributions over words, in lda2vec topics are the nearest neighbours in their embedding latent space, and are not represented in probabilities. Hence, we need to find a different evaluation metric for comparing the two models.

Instead of using these model quality measures based on held-out likelihood, we turn to topic coherence, a measure of how semantically meaningful each topic is. Word intrusion and topic intrusion are tasks that test whether a topic model's decomposition of documents and topics are coherent [Chang et al., 2009]. However, these evaluation tasks presented by Chang et al. require human annotation.

Finally, Newman et al. presented several methods drawing on existing copora like WordNet, Wikipedia and Google search engine to evaluate topic coherence. Pointwise Mutual Information (PMI) using the Wikipedia corpora showed results that were most consistent with human annotation [Newman et al., 2010]. This inspired the exploration of automatic topic coherence measures [Aletras and Stevenson, 2013] [Röder et al., 2015]. Both papers show that using a normalized version of PMI (NPMI) to build a Topic Word Space, and compute topic coherence using symmetric word similarity measures (cosine, Jaccard, and Dice) to show state-of-the-art agreement with human judgement. For consistency with Moody, we choose C_v as our evaluation metric, as described in section 6.3.