Evaluation and Labelling in Topic Models

Shraey Bhatia (710795)

Department of Computing and Information System
The University of Melbourne

Credit Points: 75
COMP60002 - Research Project

Supervisors

Prof. Timothy Baldwin
Dr. Jey Han Lau

A thesis submitted for the degree of

Masters of Science

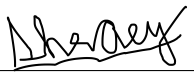
June 2017

Declaration

I certify that:

I. This thesis does not incorporate without acknowledgement any material previously submitted for degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

II. This thesis is fewer than 25000 words in length (excluding text in images, table, bibliographies and appendices).

Signed:  Date: 03/06/2017

Acknowledgements

I would like to thank a few people in helping me out in completion of my Masters' research. Firstly, I would like to express my sincere gratitude to my 2 supervisors; Prof. Timothy Baldwin and Dr. Jey Han Lau for their motivation, guidance and immense knowledge. They have been a pillar of support all along and the best supervisors I could have ever hoped for. Tim has been an inspirational figure, someone I always look upto and added that extra boost in my fascination for research. Jey has been very approachable and a perfect mentor even clearing up my low level or at times silly doubts. It has been a lot of fun having stimulating discussions with him. Thanks to my supervisors, my initial spark for research is now turning into a big flame.

Besides my supervisors, I would also like to thank IBM Research for the opportunity of working as a research intern. Working with fellow researchers was a fruitful experience and a big learning curve in my development. Additionally, I learnt how academic research and industry come together to build prototypes.

I thank my fellow classmates for the discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two and half years.

Lastly but most importantly I am really thankful to my family who stood with me. My dad who motivated me all along in difficult and frustrating times to push that extra mile and my mom who always has been close to me in stressful and challenging times. My gratitude to my younger sister who every now and then managed to bring a smile on my face with her antics. Lastly, a mention to a new introduction to the family my loving dog Ralph who though just joined me in the final month of my degree but nevertheless has been a perfect stress buster.

Abstract

This thesis is about a statistical model known as a topic model. The aim of topic models is to discover the set of topics that generated the documents in a large unstructured document collection. Topic models jointly learn "topics" and document-level topic distribution. By topics we mean an abstraction of general concept or theme in the document collection, with the idea of document-level topic distributions capture the top topics that summarize the document. For instance an article may talk about *education*, which we would hope to capture in a collection-level topic and set of topic weights for each document. This concept of *education* constitutes a topic. In the scope of the thesis, our problem statement will cover two domains: (1) evaluation of topic models, and (2) labelling of topics.

Extrinsic evaluation of topic models tend to focus exclusively on topic-level evaluation, e.g. by assessing the coherence of topics. The quality of topic model should not only based on how coherent the topics are but also on their association with documents. Here, we will demonstrate that there can be large discrepancies between topic- and document-level model quality, and that basing model-level evaluation on topic-level analysis can be highly misleading.

In the second part, we experiment with presentation of topics to humans so as to increase their interpretability. Topics generated by topic models are typically represented as a list of words. But to interpret the theme of the topic we need to analyse the list of terms, thus leading to a high cognitive overhead. To reduce the cognitive overhead of interpreting these topics for end-users, we propose an approach to label a topic with a succinct textual phrase such that it summarises its theme or idea.

The methodology used here to tackle both problems are novel and delivered promising results.

Citations to Previously Published Work

Large portions of Chapter 4 have appeared in the following paper:

> Bhatia, Shraey, Jey Han Lau and Timothy Baldwin (to appear) An Automatic Approach for Document-level Topic Model Evaluation, *In Proceedings of the Twenty First Conference on Computational Natural Language Learning (CoNLL 2017),* Vancouver, Canada.

Large portions of Chapter 5 have appeared in the following paper:

> Bhatia, Shraey, Jey Han Lau and Timothy Baldwin (2016) Automatic Labelling of Topics with Neural Embeddings, *In Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016),* Osaka, Japan, 953 - 963.

Contents

List of Figures

Chapter 1

Introduction

In this age of big data and the Internet there has been an exponential growth in the volume of unstructured text data that is readily accessible. With this enormous volume of data, there is an underlying challenge of being able to efficiently and intelligently process and understand the themes and ideas from a text collection and at the same time to be able to draw trends and conclusions from it (Wang and McCallum, 2006).

Topic models offer a way to extract a collection's themes and concepts (AlSumait et al., 2008). These themes and concepts in the lingo of topic models are known as "topics". For instance, there could be a document which talks about *agriculture* and *water scarcity* or *education*. These concepts of *agriculture*, *water scarcity* or *education* constitute the topics. Topic models jointly learn topics, which are represented as a list of words, and topic allocations to individual documents in the form of a probability distributions.

The input to a topic model is a collection of documents. These documents are pre-processed and then converted into a bag of words. The bag of words representation is concerned with only word frequencies and the ordering of the words is ignored. The output of this model is a list of learnt topics representative of the collection. The de-facto representation of topics is a list of top-n words sorted by descending probability score. For instance the topic $\langle$*school, student, university, college, teacher* $\rangle$ is represented by its top-5 words.

Topic models provide a powerful means for document collection navigation and visualisation (Newman et al., 2010a; Chaney and Blei, 2012; Smith et al., 2017). Topic models help us to categorise and classify data without needing manual annotations. Let us take an example of a news website. Topic modelling can be used to automatically categorise a news article into sports, entertainment, business or world, which results in ease of navigation for users searching for news in a particular area. Similarly in social media feeds like Twitter and Facebook, topic modelling can be employed to detect the current or hot trends.

One of the fundamental concerns of topic models is that we often come across "bad" topics. These bad topics can be a mixture of two or more themes together, or a repetition of similar topics multiple times, or just a set of words which do not collectively convey any idea. For instance, let us look at a few example topics generated automatically:

- **Topic 1** ⟨*investigation, fbi, official, department, federal, agent, investigator, charge, attorney, evidence* ⟩
- **Topic 2** ⟨*health care hospital services medical staff patients service child authority* ⟩
- **Topic 3** ⟨*taxation outpolled congratulates impracticable retook lifeboat scampered greets legroom scored* ⟩

We can see that Topic 1 and Topic 2 capture an idea or theme, e.g. Topic 1 is related to *criminal investigation* while Topic 2 is about *health and hospital*. In the case of Topic 3, it is quite difficult to think of one concept or idea associated with it, therefore can be interpreted as a bad topic (Lau et al., 2014). This quality of being a good or a bad topic is commonly referred to as coherence.

A topic model also generates topic distributions for each document in the collection. Each topic has an associated probability score that describes its significance to the document. The topics with the highest probability scores best capture the content of the document. For example let us look at the document:

"more than 2,000 attendees are expected to attend public funeral services for former nevada gov. kenny guinn . a catholic mass on tuesday morning will be followed by a memorial reception at palace station . the two-term governor who served from 1999 to 2007 died thursday after falling from the roof of his las vegas home while making repairs . he was 73 . guinn 's former chief of staff pete ernaut says attendance to the services will be limited only by the size of the venues . services start at 10 a.m. at st. joseph , husband of mary roman catholic church . ernaut says las vegas police will control traffic and security . ernaut says the moderate republican will be buried thursday in his childhood hometown of exeter , calif. , following private services."

If we look at the top -3 highest probability topics for this document:

- **Topic 1** ⟨*died family funeral honor memorial father death wife cemetery son* ⟩
- **Topic 2** ⟨*church gay marriage religious catholic pope couple pastor members bishop* ⟩

- **Topic 3** ⟨*casino las vegas nevada gambling slots payout police vehicles cars* ⟩

In this example we can observe that the document is about a Catholic funeral service in Las Vegas. Topic 1 talks about *funeral and death* and Topic 2 deals with *church and religion*. So both these topics are related in some way to the document and help us to summarize the document. Let us look at Topic 3. In its own right, Topic 3 looks like a good, coherent topic connected to the idea of *gambling and casinos*, but it does not directly capture anything about the document. Hence, Topic 1 and Topic 2 can be classified as good topics for this document, whereas Topic 3 even though being coherent can be given a tag of a bad topic for this document

One of the most popular topic model implementations has been Latent Dirichlet Allocation (" `lda` ") (Blei et al., 2003). The popularity of `lda`-style topic models has led to a wealth of variants being proposed, and hence the need for robust model evaluation strategies. Traditionally, perplexity or held-out likelihood on unseen documents has been used as an intrinsic metric to evaluate topic models (Wallach et al., 2009). Extrinsic evaluation tends to focus on topic-level evaluation, e.g. by assessing the coherence of topics (Newman et al., 2010b; Lau et al., 2014). But basing evaluation only on topic-level analysis can be highly misleading. The quality of a topic model is not only based on how coherent the topics are but also on their ability to generate good topic distributions for documents (i.e. document level evaluation).

Meanwhile, in recent years, deep learning has made significant progress for tacking problems in Natural Language Processing (" NLP "). Similar to topic models, the intermediate or hidden layers in the associated neural networks give us low dimensional representations of documents. Deep learning models are flexible and are not constrained by the bag of words assumption and so can preserve word ordering. Inspired by their flexibility and performance, there is considerable research interest to implement topic models using neural networks. One such example is a neural topic model (" `ntm` ") (Cao et al., 2015). For all the strengths of flexibility and performance, `ntm` suffers from the same shortcomings of conventional topic models of producing both good and bad topics. We will give a detailed account of this in later chapters.

In contexts where the output of the topic model is presented to a human user, a fundamental concern is the best way to present the rich information generated by the topic model, in particular, the topics themselves, which provide primary insights into the document collection. The usual representation is to express the topics as a list of words. But to interpret the theme of the topic we need to analyse the list of terms, which leads to a high cognitive overhead. This has led to interest in the task of generating labels or succinct phrases for

topics. For example we argue that a topic: ⟨*school, student, university, college, teacher, class, education, learn, high, program* ⟩ can be labelled and presented simply as EDUCATION.

An important point which we have not discussed is how we handle the topic model evaluation and topic labelling. One way forward is to have human judges perform manual annotations or judgements. But this is tedious and ill-suited for large scale evaluation. Hence, we ideally want to automate the process. We will introduce automatic evaluation and labelling approaches here, which emulate human assessment.

This concludes our brief introduction. To summarise, we introduced topic models and deep learning in NLP. We described the input and output for topic models. Additionally, we gave some examples of topics and topic distributions for documents. We then talked about topic model quality and how it is important to judge the quality of the topic model in terms of both topic and document-level model quality. Finally, we presented the idea of labelling of topics to reduce the cognitive overhead.

1.1 Contributions

The contributions of the thesis are as follows:

- We re-implement Neural Topic Model (`ntm`) (Cao et al., 2015) to reproduce its results, and then discuss its limitations.
- We empirically demonstrate that there can be large discrepancies between topic and document-level topic model evaluation.
- We demonstrate that previously-proposed document-level evaluation approaches can be misleading, and propose an alternative evaluation method.
- We propose an automatic approach to topic model evaluation based on analysis of document-level topic distributions, which we show to correlate strongly with manual annotations.
- We propose a label generation approach based on combined word and document embeddings, which is both considerably simpler than and empirically superior to the state-of-the-art generation method.
- We propose a simple label ranking approach that exploits character and lexical information, which is also superior to the state-of-the-art ranking approach.

- We release of an open source implementation of our method, including a new dataset for topic label ranking evaluation.[1]

1.2 Thesis Outline

The structure of this thesis is as follows:

- **Chapter 2**
  In this chapter, we first give an overview of topic models. Next, we describe deep learning techniques in NLP and their utility in the field of topic modelling. Finally, we present some past literature on topic model evaluation measures and automatic labelling of topics.

- **Chapter 3**
  In this chapter, we first describe Neural Topic Model (`ntm`) (Cao et al., 2015). Then, we reimplement it to reproduce the published results. Next, we discuss the implementation details, results on various datasets, and limitations associated with the topic model.

- **Chapter 4**
  In this chapter, we first evaluate topic model quality at the topic-level using topic coherence. We then demonstrate that topic-level evaluation approaches can be misleading and present document-level evaluation of topic-models. Finally, we propose an automatic approach to automate document level evaluation.

- **Chapter 5**
  In this chapter, we investigate the task of topic labelling as a means to reduce the cognitive overhead related to understanding topics. We propose a novel approach to topic labelling based on word and document embeddings, which both automatically generates label candidates given a topic input, and ranks the candidates in either an unsupervised or supervised manner, to produce the final topic label.

- **Chapter 6**
  In this chapter, we summarize the content of various chapters and propose future work.

[1] https://github.com/sb1992/NETL-Automatic-Topic-Labelling-

Chapter 2

Background

2.1 Topic Models

Topic models exist in different variants and are typically unsupervised. They generate topics t_i in the form of multinomial distributions over the terms w_j of the document collection ($\Pr(w_j|t_i)$), and topic distributions for each document d_k in the collection, in the form of a multinomial distribution over topics ($\Pr(t_i|d_k)$).

One of the first topic models was Latent Semantic Analysis (LSA) proposed by Deerwester et al. (1990). It simply constructs a term-document matrix and reduces it into a lower dimensionality by means of Singular Value Decomposition (SVD). The term document matrix is a $W \times D$ matrix where W corresponds to the size of vocabulary i.e unique terms in the document collection and D is the number of documents in the collection. The values of the matrix are weighted, traditionally using tf-idf (term frequency-inverse document frequency), a standard information retrieval weighting scheme.

The term document matrix is quite sparse, and decomposed using SVD. A is decomposed as:

$$A = U\Sigma V^T$$

where U is the new term matrix of dimensionality $W \times M$ and V^T is a new document matrix of dimensionality $M \times D$. Σ is a diagonal matrix of dimensionality $M \times M$ where $M = Rank(A)$, and contains singular values of A. Truncating U, V and Σ to K dimensions produces the topic model for K topics. Hence after truncating, U_K corresponds to the topics and V_K^T is the topic distribution for the documents.

One of the most widely used topic models as first introduced by Blei et al. (2003), is Latent Dirichlet Allocation (`lda`). `lda` is a generative model that generates observable data

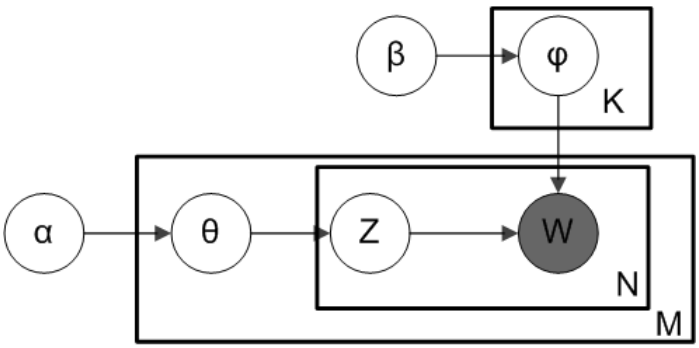


Figure 2.1: Plate notation of LDA

using some hidden structure. A visual representation of `lda` is given in Figure 2.1 in block notation.

The variables of `lda` given in Figure 2.1 are:

- K is the number of topics
- N is the total number of words in the document collection
- M is the total number of documents
- ϕ is the word distribution in the topics
- θ is the topic distribution in the documents
- Z is the topic assignment for the word
- W is the word in the document.
- α is the Dirichlet prior for topic distribution in documents
- β is the Dirichlet prior for word distribution in topics

`lda` uses Dirichlet priors (α and β) which are conjugate priors for the word-topic (ϕ) and topic-document (θ) multinomial distributions. Calculation of the posterior of `lda` is intractable meaning that methods like Collapsed Gibbs Sampling (Griffiths and Steyvers, 2004) are used to approximate it. Collapsed Gibbs Sampling takes advantage of the conjugacy property of Dirichlet distributions to simplify the integration hence modelling the posterior into the same form as prior.

The generative process of `lda` is as follows:

- Draw $\theta^{(d)}$ from Dirichlet(α) for each document.
- Draw $\phi^{(z)}$ from Dirichlet(β) for each topic.
- Draw a topic z from $\theta^{(d)}$.
- Draw a word from $\phi^{(z)}$.

One of the limitations of `lda` is that it makes an independence asssumption between topics. `lda` uses a Dirichlet prior on topic proportions and under the Dirichlet prior the components of the topic proportion are nearly independent. To address this, Blei and Lafferty (2006) proposed the Correlated Topic Model (`ctm`) an extension of `lda`. The goal behind `ctm` is the need to model correlations between different topics, and reduce overlap in topic content. `ctm` uses a logistic normal prior over topic proportions instead of a Dirichlet prior. The logistic normal is constructed from multivariate Gaussian Distribution which helps in inducing the dependencies in topics. But for all the positives of the logistic normal, its non-conjugacy makes it difficult to use Markov Chain Monte Carlo sampling techniques to approximate the posterior.

Although `lda` is the current state-of-the-art topic model, it is a parametric model and without any notion of hierarchy. Teh et al. (2004) introduced HDP-LDA, a non-parametric topic model which models the number of topics depending on a document collection. Their main work centred around improving the topic-word priors and also dynamically estimating "correct" number of topics. Taking inspiration from this work, Buntine and Mishra (2014) introduced a toolkit `hca` to train and test non parametric topic models. The authors proposed two extensions:

- To capture word burstiness (Doyle and Elkan, 2009).
- To place different priors on the document-topic and word-topic components.

The idea behind word burstiness is that there tends to be higher likelihood of seeing a word which has already been seen recently. In `lda` a symmetric prior i.e. Dirichlet prior, is used in both the document-topic and topic-word components. There has been a lot of experimentation with asymmetric priors of different types but most of them are computationally intensive. Recently, with Pitman Yor Processes (PYPs) also known as table indicator sampling, the authors have been able to make it computationally quicker with just a small space-time overhead in comparison to standard collapsed Gibbs Sampling. PYP is placed in the word-topic component and help us to model word-generations of the topic model. PYP has dynamic memory requirements and is able to reach convergence faster as it uses Collapsed Gibbs Sampling (Griffiths and Steyvers, 2004).

2.2 Deep Learning Embeddings

One of the recent breakthroughs in the field of deep learning has been the usage of neural network inspired embeddings as relevent pretraining methods. Mikolov et al. (2013b) proposed `word2vec` to learn word embeddings, which they found to perform strongly over a range of word similarity tasks, and also to be useful for initialising deep learning models. Two approaches are proposed in the paper: `cbow` and `skip-gram`. `cbow` combines neighbouring words to predict a target word, while `skip-gram` uses the target word to predict neighbouring words. Both approaches use a feedforward neural network with a non-linear hidden layer to maximize the objective function; to improve computational efficiency, the authors propose using negative sampling.

As an extension of `word2vec`, Le and Mikolov (2014) introduced `doc2vec` to learn embeddings for word sequences (e.g. paragraphs or documents). By treating each document as a word token, the same `word2vec` methodology can be used to learn document embeddings. The authors propose two implementations: `dbow`, which uses the document vector to predict its document words, and is the `doc2vec` equivalent of `skip-gram`; and `dmpv`, which uses a small window of words and concatenates them with the document vector to predict a document word, and is the `doc2vec` equivalent of `cbow`. Strictly speaking though, `cbow` combines word vectors by summing them, while `dmpv` combines word vectors and the document vector by concatenating them. Compared to `dbow`, `dmpv` takes into account the local word ordering, and has a higher number of parameters, since the input is a concatenation of vectors.

2.3 Evaluation of Topic Models

In recent years, with different variants of topic models it is important to have consistent evaluation metrics. Traditionally, due to the unsupervised nature of topic models, evaluation has been intrinsic. Perplexity or held-out likelihood is one such commonly used intrinsic evaluation metric (Wallach et al., 2009). The concept of perplexity is imported from language modelling, where we estimate the probability of words on unseen or test documents. The shortcoming with perplexity, though, is that it does not give us any measure of quality of topics or topic allocations for a document. On the contrary, Chang et al. (2009) showed that there is low or negative correlation between perplexity and direct human evaluations of topic model quality.

Chang et al. (2009) proposed two human judgement tasks, at the topic and document levels. The two tasks took the following form of "intruder" tasks, whereby subjects were

tasked with identifying an intruder topic word for a given topic, or an intruder topic for a given document. Specifically, in the word intrusion task, an intruder word was added to the top-5 topic words, and annotators were asked to identify the intruder word. Similarly in the topic intrusion task, each document was presented with 4 topics — the top 3 topics corresponding to the highest probability topics for the document and a random intruder topic — and subjects were asked to spot the intruder topic. The intuition behind both methods is that the higher the quality of the topic or topic allocation for a given document, the easier it should be to detect the intruder.

Chang et al. (2009) performed the analysis on 3 types of topic models:

- `pLSI` (Hofmann, 1999)
- `lda` (Blei et al., 2003)
- `ctm` (Blei and Lafferty, 2006)

In the word intrusion task, the authors calculated perplexity and the average precision of human judgements. The findings were that even though CTM performed best over perplexity, it was the worst in the word intrusion task, concluding that CTM produced less coherent topics. For the topic intrusion task, Chang et al. (2009) based their evaluation on perplexity and topic log odds. The authors defined topic log odds for a document–topic pair as the difference in the log-probability assigned to the intruder and the log-probability assigned to the topic chosen by a given annotator, which they then averaged across annotators to get a TLO score for a single document. As with the word intrusion task, `lda` and `pLSI` outperformed `ctm`, countering `ctm`'s superior performance on perplexity.

Newman et al. (2010b) proposed to measure topic coherence directly in the form of "observed coherence", in which human judges rated topics directly on an ordinal 3-point scale. The computed topic coherence score is calculated by the arithmetic mean of a similarity measure over all pairings of topic words. They experimented with a range of different similarity methods to automate the rating task:

- Association measures
- Wordnet Similarity.
- Wikipedia measures
- Search Engine based measures

They reported the best results were achieved by simply aggregating pointwise mutual information (pmi) scores for different pairings of topic words, based on a sliding window over English Wikipedia (Newman et al., 2010b).

In other work, Mimno et al. (2011) also suggested a similar approach of calculating coherence, but used conditional probability instead of pmi. They replaced English Wikipedia with the topic model training documents to calculate co-occurrence counts and showed that their method correlates with human annotations.

Building on the work of Newman et al. (2010b), Lau et al. (2014) proposed an improved method for estimating observed coherence based on normalised pmi (npmi). The authors further automated the word intruder detection task introduced by Chang et al. (2009). They computed a combination of word association features like pmi, npmi, CP1, and CP2 on the top N topic words of a topic. Lau et al. (2014) then combined the features in a learn to rank support vector regression model (Joachims, 2006) to learn the intruder word. Additionally, the authors showed a strong correlation between word intrusion and observed coherence, and suggested that it is possible to perform model-level evaluation based on aggregation of word intrusion or observed coherence scores across all topics.

2.4 Labelling of Topics

Presentation or Interpretation of generated topics is very significant for topic modelling. Conventionally, topics are presented as a list of top -n words. But this requires the words to be analysed to understand the theme or idea, therefore inducing cognitive overhead.

In terms of automatic labelling of topics, Mei et al. (2007) introduced the task of generating labels for LDA topics. They based it on first extracting bigram collocations or noun chunks from the topic-modelled document collection, known as the reference collection. The second step involved ranking the extracted candidates. Mei et al. (2007) used two scoring functions: the zero-order relevance and first-order relevance. The zero-order relevance function is based on the idea that the label should have high probability topic terms. In the case of first-order relevance, it takes the reference collection into account. The reference collection is used to calculate co-occurrence matrix for labels and topic terms. It computes the pmi score between the candidate label and topic terms and sums it up. The approach is completely unsupervised.

Lau et al. (2011) proposed using English Wikipedia to automatically label topics. First, they map the topic to a set of concepts by querying Wikipedia using the top-10 topic terms based on: (a) Wikipedia's native search API; and (b) Google's search API, with site restriction. The top-8 article titles from each of these two sources are pooled to generate the

primary candidate topic labels. Secondary labels are generated from component n-grams contained within the primary candidates, and filtering out incoherent and unrelated titles using the `RACO` measure (Grieser et al., 2011) to measure similarity with the primary labels, based on Wikipedia document categories. The combined set of primary and secondary label candidates is then ranked using a number of lexical association features, either directly in an unsupervised manner, or indirectly based on training a support vector regression model. The authors provide an extensive analysis of their method with that of Mei et al. (2007), and find their label generation and ranking methodology to be empirically superior (in both an unsupervised and supervised setting). In this thesis, we seek to improve upon the topic labelling benchmark set by Lau et al. (2011).

Hulpus et al. (2013) developed a graph-based method for topic labelling, leveraging the structure of DBpedia concepts. Their approach is styled around graph-based word sense disambiguation, and extracts a set of DBpedia concepts corresponding to the top-N terms of a topic. They then construct a graph centred around DBpedia concepts and filter noise based on graph connectivity (the hypothesis being that sense graphs of words from a topic should be connected). To find the best label for a topic, they experiment with a variety of graph centrality measures.

In work slightly further afield, Zhao et al. (2011) proposed topical keyphrase extraction for Twitter. Although, the work focuses mainly on Twitter, the methodology can be applied to other domains and to label topics. Zhao et al. (2011) follow a three-step process for keyphrase extraction: (1) keyword ranking; (2) candidate keyphrase generation (based on the individual keywords); and (3) keyphrase ranking. They use a novel topic context-sensitive *PageRank* method to regularise topic scores for keyword ranking, and a probabilistic scoring method that takes into account relevance, interestingness and keyphrase length for keyphrase ranking.

Building on `word2vec`, Kou et al. (2015) experimented with neural embeddings in the context of topic labelling. In addition to `skip-gram` and `cbow` word vectors, the authors also included letter trigram vectors of a word, with the rationale that it generalises over morphologically-related forms of the same word. Their methodology consists of first generating candidate labels for topics from topic-related documents using a chunk parser. By representing both topic words and topic labels using word embeddings and letter trigrams, they rank the labels using cosine similarity to obtain the best label for a topic. In their evaluation, they find that simple letter trigrams are ultimately the most reliable means of label ranking

Chapter 3

Neural Topic Model

3.1 Introduction

In recent years, deep learning has take played a major role in solving a NLP problems. Deep learning sidesteps the need for intensive feature engineering and automatically learns low-dimensional vectors for them. In Section 2.4 we discussed the deep learning methods like `word2vec` and `doc2vec` that map words and documents as low dimensional vector space. Inspired by these approaches Cao et al. (2015) introduced a neural network implementation of topic models. They propose Neural Topic Model (`ntm`), a topic model from the perspective of neural network.

3.2 Methodology

In this section we describe `ntm` in more detail. Topic models explain the appearance of words in document (i.e. $P(w|d)$) via topics, by computing topics t_i in the form of multinomial distribution of words w in the document collection ($P(w|t_i)$) and topic distribution of each document d in the collection in the form of multinomial distribution over topics ($P(t_i|d)$).

$$P(w|d) = \sum_{1 \leq i \leq K} P(w|t_i)P(t_i|d) \tag{3.1}$$

where K is the pre-defined number of topics.

We can re-write Equation 3.1 in vector form where we express the topic-word component by $\phi(w) = [p(w|t_1), ... p(w|t_K)]$ and the topic-document multinomial by $\theta(d) = [p(t_1|d), ... p(t_K)]$.

$$P(w|d) = \phi(w)\dot{\theta}^T(d) \tag{3.2}$$

Equation 3.2 can then be optimised using a simple neural network where $\phi(w)$ is a look up layer for words and $\theta(d)$ is an embedding layer for documents, with the output layer being the dot product between the two.

The architecture of the model is illustrated in Figure 3.1. Briefly, the left side of the network; (g) is modelling the words into distributed vector representations while the right side; (d_p and d_n) is trying to capture document representations. The addition of a parallel layer as shown in the right frame of the Figure 3.1 extends it into a supervised model. To better understand the dynamics of the model, we will explain each layer in more detail.

First, the input layer (g,d) is the pair of word g and a document id $d \in D$ where D is the document set. The embedding layer (le) contains the distributed representation of words. Here, we use pre-trained `word2vec` vectors (Mikolov et al., 2013a,c), which are trained on Google News,[1] where each word is represented by a 300 dimensional vector.

Next, Cao et al. (2015) proposed adding the topic layer (lt) to transform the word embeddings embeddings into K-dimensional topic vectors activated by the sigmoid activation function. Formally:

$$lt(g) = sigmoid(le(g) \times W_2) \tag{3.3}$$

where $W_2 \in \mathbb{R}^{300 \times K}$ is the weight matrix between the embedding layer and topic layer.

In parallel, there is a document layer (ld) given by a document look up matrix to model the topic-document component of a topic model similar to $\theta(d)$. It is given by:

$$ld(d) = softmax(W_1(d, :)) \tag{3.4}$$

where $W_1 \in \mathbb{R}^{D \times K}$ and $W_1(d, :)$ is the vector representation of document d similar to $\theta(d)$. Softmax is used to normalize the probabilities.

Lastly there is a scoring layer (ls) which give the dot product between document layer and topic layer to generate a score between 0 and 1.

$$ls(g, d) = lt(g) \times ld(d)^T \tag{3.5}$$

To extend the model to supervised learning Cao et al. (2015) included additional model parameters (right frame in Figure 3.1), (a) a classification layer where the label layer (ll) is introduced parallel to the scoring layer, and (b) a topic document layer which acts as

[1] Available from: `https://code.google.com/archive/word2vec`.

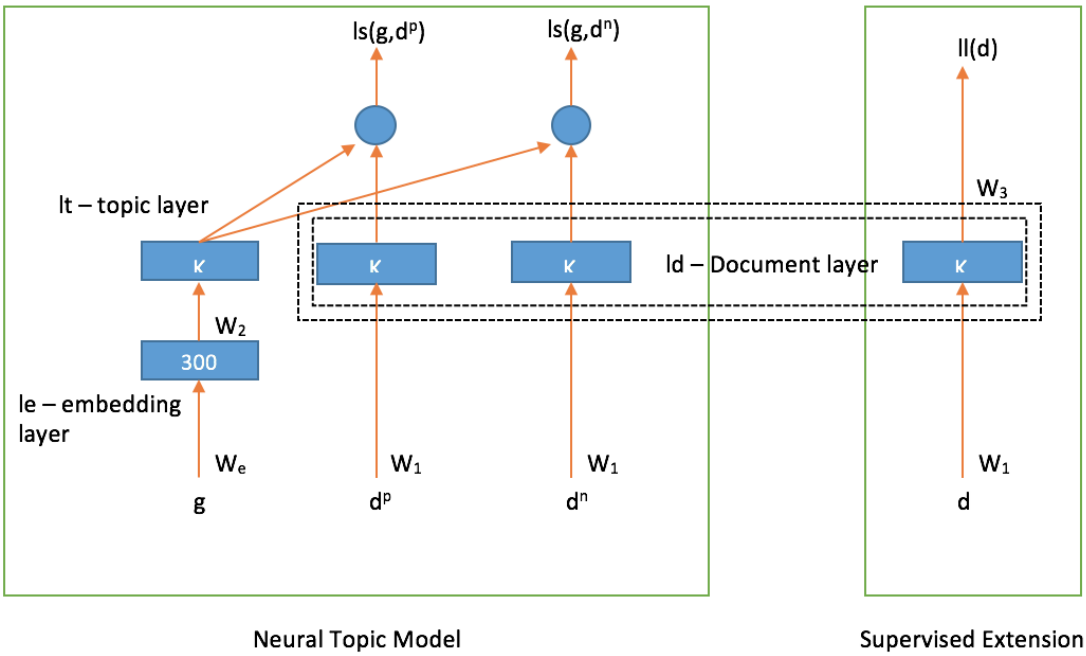


Figure 3.1: Neural topic Model and its supervised extension

the input for the multi-class classification task. Since the same W_1 weight is also used here it should help in tuning the document embedding matrix which will in turn result in improvements in the topic-word distribution. The labelling layer is given by Equation 3.6,

$$ll(d) = f(ld(d) \times W_3) \tag{3.6}$$

where $W_3 \in \mathbb{R}^{K \times n_c}$; n_c is the number of classes and f is the activation function.

Our training instances comprise of input word document pairs (g, d). We will train the model using mini-batches. In training, for each document d_p that contains word g, we will also randomly sample a negative document (d_n) which does not contain g. We will use max margin as the cost function.

$$c(g, d^p, d^n) = max(0, 0.5 + ls(g, d^n) - ls(g, d^p)) \tag{3.7}$$

For the supervised model, it has an additional label cross-entropy loss to optimise W_3 and W_1. For test or unseen documents, Cao et al. (2015) propose applying inference steps to update W_1 for the new documents but freeze all other parameters (W_e, W_2, W_3).

Dataset	#Docs	#Tokens
APNEWS	50K	15M
BNC	15K	18M

Table 3.1: Statistics for APNEWS and BNC

Partition	#Docs	#Tokens
Training	9314	2.6M
Development	2000	0.5M
Test	7532	1.7M

Table 3.2: 20NEWS preprocessed statistics.

3.3 Implementation

In Section 3.2 we discuss the structure of the neural network. In this section, we describe the implementation details of the neural network. To this end, we use Google's deep learning framework Tensorflow for our experiments.[2] As part of our experiments, we will first try to reproduce the supervised model performance of Cao et al. (2015) on 20NEWS dataset. Additionally, we evaluate the unsupervised version of the model on APNEWS and BNC datasets.

3.3.1 Datasets, Preprocessing and Setting

20NEWS is a dataset frequently used for text-classification as it is made up of forum like messages categorised into 20 categories. We use this dataset on the supervised model as it is a multi-class classification problem. We randomly sample 2K documents from the training data to construct the development set.

Additionally, we use two more document collections for unsupervised experiments: APNEWS and BNC. Documents in these datasets do not have any labels associated with them.

- APNEWS is a collection of Associated Press[3] news articles from 2009 to 2016.
- BNC is an amalgamation of extracts from different sources such as books, journals, letters and news.

We sample 50K and 15K documents from APNEWS and BNC respectively to create two datasets for our experiments

[2]Available from: https://www.tensorflow.org/
[3]https://www.ap.org/en-gb/

Topic No.	Accuracy
50	.649
100	.639
150	.628
200	.636

Table 3.3: 20NEWS classification accuracy.

Model	APNEWS	BNC
`lda`	0.16	0.14
`ntm`	0.10	0.08

Table 3.4: Topic coherence scores (`npmi`)

In the original paper, preprocessing details for the dataset were absent. As such, we follow standard preprocessing procedures where we tokenise words and sentences using Stanford Core NLP (Manning et al., 2014). We additionally remove stop words,[4] lower-case all word tokens, filter word types which occur less than 10 times and exclude the top 0.1% most frequent word types. Preprocessed statistics for the different document collections are given in Table 3.1 and Table 3.2.

Cao et al. (2015) provided the hyper-parameters in the paper but due to different preprocesing steps we use the development set to tune our hyper-parameters. We initialise the weights W_1, W_2 and in the case of supervised model W_3 randomly using a truncated random normal initializer. Additionally, we do not back-propagate on W_e and keep them fixed. Adam optimizer is used for learning and we fix our learning rate to 0.001 and batch size to 64. In the supervised model we fix the activation function f (Equation 3.6) to sigmoid. For negative sampling of documents, we sample them dynamically so that noise is not the same for every iteration. For consistency with the original paper we will use classification accuracy as the evaluation metric (Cao et al., 2015). We will test model over different numbers of topics: 50, 100, 150 and 200.

3.4 Results and Limitations

The classification accuracy for 20NEWS is given in Table 3.3. We can see that it fluctuates between 63-65% depending on the number of topics. This is lower than the numbers mentioned in the original paper but due to different and missing implementation details in the original paper, it is difficult to reproduce the exact numbers. Table 3.5 presents some

[4]We use Mallet's stop word list: https://github.com/mimno/Mallet/tree/master/stoplists

Dataset	Topics
APNEWS	shipments shipment exports imports exported shipped shipping stockpiles export deliveries
	burglary burglaries robbery burglarizing burglarized larceny shoplifting thefts robberies burglar
	travel airfare travelers fares traveling flights journeys trips visas commute
BNC	painters sculptors printmaking painter impressionists portraiture potters paintings carvers sculptor
	unhappy dissatisfied saddened disapproved disapprove irked distraught displeased incensed fuming
	threat threats danger menace alert dangers hazard scare peril impracticable

Table 3.5: Example APNEWS and BNC topics.

sample topics corresponding to APNEWS and BNC. Each topic is represented by its top-10 most probable words. Eyeballing the topics, we see that the topics do look coherent. In Section 2.3, we discussed about using Normalised pointwise mutual information (`npmi`) as a metric to measure topic coherence. We compute topic coherence for `lda` and `ntm` topic models over APNEWS and BNC using `npmi` (Lau et al., 2014) and present the results in Table 3.4.[5] We can see that though the scores for `ntm` are lower than that of state-of-the-art topic model `lda`, it is still good enough to paint a coherent picture for the topic model. But on a closer look at topics, we notice that the topics include different forms of the same word (e.g. *burglary*, *burglaries*) or near synonym words (e.g. *unhappy*, *saddened*) in the same topic. To better understand how well these topics describe documents, we present a number of documents and their top topics in Table 3.6. We can see that in general there is little association between the topics and the document content. This does give us some intuition into the poor quality of topic model at the document level, an idea which we explore at length in Section 4.

One of the limitations of the paper was that other than the classification accuracy there was no other evaluation on the topic model quality either at the topic-level or document-level. A high classification for the task does not directly translate to high quality topics. As we saw in Table 3.5 and Table 3.6, although the topics do look coherent but they are too fine grained to describe the document collection. These results form the kernel for Section 4 where we will study topic model quality in more detail.

[5]We use the following open source toolkit to compute topic coherence: https://github.com/jhlau/topic_interpretability.

3.5 Summary

In this chapter, we first reproduced `ntm` for 20NEWS dataset. Then we applied it to APNEWS and BNC to get an idea of generated topics. Next, we calculated topic-coherence for `ntm` using `npmi` and saw that generated topics are quite coherent. Finally, we gave an intuition on discrepancies between topic coherence and document–topic associations.

Dataset: APNEWS	**Document**	a judge in will county has approved further testing on the coat an oswego man was wearing when his wife and three children were found shot to death in 2007 . christopher vaughn is accused of killing his family inside their suv , which was parked on a frontage road along interstate 55 . authorities found kimberly vaughn shot to death , along with their children , 12-year-old abigayle , 11-year-old cassandra and 8-year-old blake . assistant state 's attorney mike fitzgerald on monday said prosecutors asked for more dna testing on the coat . the (joliet) herald-news reports the coat then will be sent to a ballistics expert for tests requested by vaughn 's attorney . vaughn told police that his wife shot him in the leg before killing the children and committing suicide . he 's charged with first-degree murder . ..
	Topics	0: arraigned burglarizing arrested bigamy detectives motorcyclist arraignment coroner accomplice fondled **probability: 0.92** 1: quarterly pretax dividend profit annualized earnings profits stockholders writedown premarket **probability: 0.03** 2: policies homeownership recessions income portfolios homebuyers premiums showrooms clientele lifestyles **probability: 0.009**
	Document	in the end , the american soldiers didn't get to burn down the wheat field . and that was a victory , much as some had anticipated the fiery show . the mine-strewn patch could have become a flash point of resentment for afghan villagers in the remote arghandab valley if the soldiers had gone ahead with their plan . instead , the painstaking efforts of the u.s. army 2nd battalion , 508th parachute infantry regiment , yielded a surprise. the villagers cleared the field themselves , a small victory of cooperation of the kind the international force must replicate if it hopes to turn around the war in afghanistan . for the past few weeks , the soldiers out of fort bragg , north carolina , have been talking to people in the area about the wheat patch nearly as large as a football field that , suspiciously , hadn't been harvested . this part of the arghandab valley , north of kandahar city ...
	Topics	0: dec autopsy pentobarbital stabbed executions beheading gunshot subjective wounding clemency **probability: 0.89** 1: rose shrank pct decliners quadrupled exhibitors parade spectrum index outperform **probability: 0.04** 2: asked asks questionnaire requested wondered quizzed declined begged questionnaires request **probability: 0.01**
Dataset: BNC	**Document**	belfast central library for your business information needs inquiry service to users in person and by telephone , telex or post . immediate access without formality to the largest collection of business literature in northern ireland . photocopying while you wait . trade directories , telephone directories , company information , official and legal publications and other material of business interest is available in the business library . ... the business library has the answers ... who makes beehives in britain ? ... or yachts in yugoslavia ? the answers to these and many other questions can be found by reference to our wide range of general and specialized directories which cover britain , europe and the rest of the world . whose trade name is transitray ? the uk trade names directory lists over 50,000 trade names of british manufacturers . what is the capital of capital gearing ? you will find this and much more information in such sources as the extel card services , mccarthy information services , kompass uk trade directory ...
	Topics	0: subjective commas tabular impracticable chromatin bibliographies felsic impedance password representativeness **probability: 0.93** 1:diet diets kingdom dieting dieter dietary calories calorie vitamin vitamins **probability: 0.03** 2: kingdom nutcracker palace mythological priestess noblemen pharaoh mystical courtier kingdoms **probability: 0.007**
	Document	unced environmental destruction " killing millions " says who an independent report prepared for the un 's world health organization (who) claims that environmental destruction resulting from over-population is killing millions of people every year , largely as a result of the contamination of water , soil and air . the report , our health , our planet , says that unless population growth can be drastically reduced , the resources needed to support the human race will be overwhelmed . the birth rate will only be cut , however , if the health prospects of poor families are improved . at present , 3.2 million children die every year from diarrhoeal diseases , a further two million from malaria , while hundreds of millions are infested with intestinal parasites . respiratory and other complaints triggered by air pollution affect hundreds of millions in both rich and poor countries , the report adds ...
	Topics	0: accuses cites disseminating reports disseminated ng memorandum investigative systematic resumes **probability: 0.90** 1: unhappy displeased defamatory mislead impracticable misleading dismayed dissatisfied incensed angers **probability: 0.05** 2: gardner gardener botanist gardeners florist bonsai naturalist potter jim robert **probability: 0.02**

Table 3.6: Documents and their top-3 topics

Chapter 4

Document Level Topic Model Evaluation

4.1 Introduction

In Section 3 we discussed on lack of topic quality evaluation measures to access the quality of `ntm`. Additionally, we also saw that `ntm` was producing very fine grained and different inflectional forms of the same word leading us to think that there may be some inconsistency in the topic distributions associated with documents from our collections. Topic models should not only produce coherent topics but should also capture general concepts from our document collection.

In Section 2.3 we saw that most of the research on topic model quality has focussed on primarily on evaluating the quality of individual topics and largely ignore the evaluation of topic allocations to individual documents. Hence, it has become widely accepted that topic-level evaluation is a reliable indicator of the intrinsic quality of the overall topic model (Lau et al., 2014). In this chapter we will not only compute coherence scores of different topic models as suggested by Lau et al. (2014) but will also challenge the above assumption, and demonstrate that topic model evaluation should operate at both the topic and document levels.

4.2 Datasets and Topic Models

In Section 3.3.1 we introduced two datasets APNEWS and BNC along with the preprocessing steps. We will use the same 2 preprocessed datasets in this chapter for our experiments.

Similarly to Chang et al. (2009), we base our analysis on a representative selection of topic models, each of which we train over APNEWS and BNC to generate 100 topics:

- **`lda`** as introduced in Section 2.1 uses a symmetric Dirichlet prior to model both document-level topic mixtures and topic-level word mixtures. It is one of the most

commonly used topic model implementations and serve as a benchmark for comparison. We use Mallet's implementation of `lda` for our experiments, noting that Mallet implements various enhancements to the basic LDA model, including the use of an asymmetric–symmetric prior: `http://mallet.cs.umass.edu/`.

- **`ctm`** (Blei and Lafferty, 2006) is an extension of `lda` that uses a logistic normal prior over topic proportions instead of a Dirichlet prior to model correlations between different topics and reduce overlap in topic content.

- **`hca`** (Buntine and Mishra, 2014) proposes an extension to capture word burstiness (Doyle and Elkan, 2009). The idea behind the model is that there tends to be higher likelihood of generating a word which has already been seen recently. Word generation is modelled by a Pitman–Yor process (Chen et al., 2011).

- **`ntm`** as discussed in Section 3, topic–word multinomials are modelled as a look-up layer of words and topic–document multinomials are modelled as a look-up layer of documents. The output layer of the network is given by the dot product of the two vectors. In this section we use only the unsupervised variant in our experiments.

- **`cluster`** is a baseline topic model, specifically designed to produce highly coherent topics but "bland" topic allocations. We represent word types in the documents with pre-trained `word2vec` vectors (Mikolov et al., 2013a,c), pre-trained on Google News,[1] and create word clusters using k-means clustering ($k = 100$) to generate the topics. We derive the multinomial distribution for each topic based on the cosine distance to the cluster centroid, and linear normalisation across all words.

To generate the topic allocation for a given document, we first calculate a document representation based on the mean of the `word2vec` vectors of its content words. For each cluster, we represent them by calculating the mean `word2vec` vectors of its top-10 words. Given the document vector and clusters/topics, we calculate the similarity of the document to each cluster based on cosine similarity, and finally (linearly) normalise the similarities to generate a probability distribution.

4.3 Topic-level Evaluation: Topic Coherence

Pointwise mutual information (and its normalised variant `npmi`) is a common association measure to estimate topic coherence (Newman et al., 2010b; Mimno et al., 2011; Aletras and

[1] Available from: `https://code.google.com/archive/word2vec`.

Model	APNEWS	BNC
lda	0.16	0.14
hca	0.14	0.08
ctm	0.07	0.09
ntm	0.10	0.08
cluster	0.18	0.17

Table 4.1: Topic coherence scores (npmi)

Model	Topics
lda	oil gas drilling gulf spill natural pipeline wells industry energy computer video screen program text disk windows electronic machine graphics health care hospital services medical staff patients service child authority
cluster	river creek lake rivers dam tributary lakes reservoir tributaries creeks prohibited forbid prohibiting prohibits violated prohibit contravened forbids violate barred terrace courtyard staircase staircases courtyards walls pergola walkway stairways walkways

Table 4.2: Example lda and cluster topics.

Stevenson, 2013b; Lau et al., 2014; Fang et al., 2016). Although the method is successful in assessing topic quality, it tells us little about the association between documents and topics. From other experiments we notice that a topic model can produce topics that are coherent — in terms of pmi association — but poor descriptors to represent the overall concepts in the document collection.

We first compute topic coherence for all 5 topic models over APNEWS and BNC using npmi (Lau et al., 2014) and present the results in Table 4.1.[2] We see that lda and cluster has a consistent and strong performance across both datasets. hca performs well in APNEWS but poorly in BNC. Both ctm and ntm topics appear to have low coherence in the two datasets.

Based on these results, one would conclude that cluster is a good topic model, as it produces very coherent topics. To better understand the nature and quality of the topics, we present a random sample of lda and cluster topics in Table 4.2.

Looking at the topics, we see that cluster tends to include different inflectional forms of the same word (e.g. *prohibited*, *prohibiting*) and near-synonyms/sister words (e.g. *river*, *lake*, *creeks*) in a single topic. This explains the strong npmi association of the cluster

[2]We use the following open source toolkit to compute topic coherence: https://github.com/jhlau/topic_interpretability.

topics. On the other hand, `lda` discovers related words that collectively describe concepts rather than just clustering (near) synonyms. This suggests that the topic coherence metric alone may not completely capture topic model quality, leading us to also investigate the topic distribution associated with documents from our collections.

4.4 Human Evaluation of Document-level Topic Allocations

In this section, we describe a series of manual evaluations of document-level topic allocations, in order to get a more holistic evaluation of the true quality of different topic models (in line with the original work of Chang et al. (2009)).

4.4.1 Topic Intrusion

The goal of the topic intrusion task is to examine whether the document–topic allocations from a given topic model accord with manual judgements. We formulate the task similarly to Chang et al. (2009), in presenting the human judges with a snippet from each document, along with four topics. The four topics comprise the top-3 highest probability topics related to document, and one intruder topic. Each annotator is required to pick the topic that is least representative of the document, with the expectation that the better the topic model, the more readily they should be able to pick the intruder topic. The intruder topic is selected randomly, subject to the following conditions: (1) it should be a low probability topic for the target document; and (2) it should be a high probability topic for at least one other document. The first constraint is intended to ensure that the intruder topic is unrelated to the target document, while the second constraint is intended to select a topic that is highly associated with some documents, and hence likely to be coherent/not a junk topic. Each topic is represented by its top-10 most probable words, and the target document is presented in the form of the first three sentences, with an option to view more of the document if further context is needed.

We used Amazon Mechanical Turk to collect the human judgements, with five document–topic combinations forming a single HIT, one of which acts as a quality control. The control items were sourced from an earlier annotation task where subjects were asked to score the top-5 topics for a target document on a scale of 0–3. The 50 top-scoring documents from this annotation task, with their top-3 topics, were chosen as controls. The intruder topic for the control was generated by randomly selecting 10 words from the corpus vocabulary. In order to pass quality control, each worker must correctly select the intruder topic for the control document–topic item over 60% of time (across all HITs they complete). Each

Topic Model	Mean Model Precision	
	APNEWS	BNC
lda	0.84	0.66
hca	0.60	0.44
ctm	0.64	0.66
ntm	0.26	0.17
cluster	0.39	0.48

Table 4.3: Mean model precision for human judgements

Topic Model	Mean Topic Log Odds	
	APNEWS	BNC
lda	-0.78	-1.84
hca	-2.09	-3.61
ctm	-1.04	-1.60
ntm	-7.16	-6.32
cluster	-0.12	-0.10

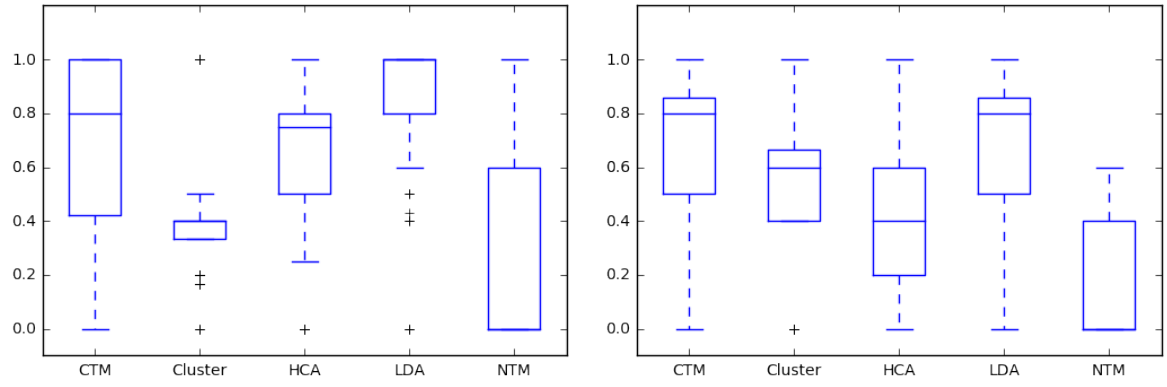
Table 4.4: Mean topic log odds for human judgements

document–topic pair was rated by 10 annotators initially, and for HITs where less than 3 annotations passed quality control, we reposted them for a second round of annotation.

For our annotation task, we randomly sampled 100 documents from each of our two datasets, for each of which we generate document–topic items based on the five different topic models. In total, therefore, we annotated 1000 (100 documents $\times$ 2 collections $\times$ 5 topic models) document–topic combinations. After quality control, the final dataset contains an average of 5.4 and 5.5 valid intruder topic annotations for APNEWS and BNC, respectively.

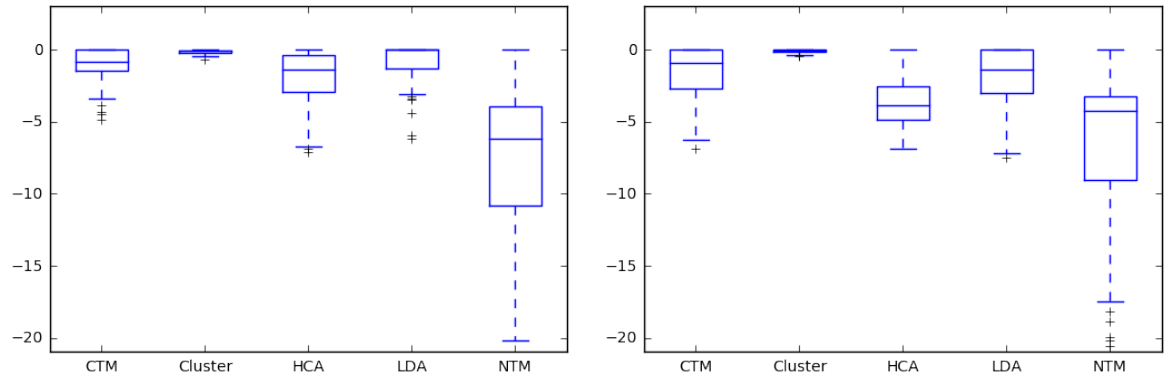
Chang et al. (2009) proposed topic log odds ("TLO") as a means of evaluating the topic intrusion task. The authors defined topic log odds for a document–topic pair as the difference in the log-probability assigned to the intruder and the log-probability assigned to the topic chosen by a given annotator, which they then averaged across annotators to get a TLO score for a single document. Separately, Chang et al. (2009) proposed model precision as a means of evaluating the word intrusion task, whereby they simply calculated the proportion of annotators who correctly selected the intruder word for a given topic. In addition to presenting results based on TLO, we apply the same methodology in our evaluation of the topic intrusion task, in calculating the proportion of annotators who correctly selected the intruder topic for a given document, which we then average across documents to derive a model-level score.

The results of the human annotation task are summarised in Tables 4.3 and 4.4. Looking at model precision for APNEWS first, we see that lda outperforms the other topic models.



(a) APNEWS (b) BNC

Figure 4.1: Boxplots of model precision



(a) APNEWS (b) BNC

Figure 4.2: Boxplots of topic log odds

`ctm` and `hca` perform credibly, whereas `ntm` and `cluster` are quite poor. Moving on to BNC, we see a drop in score for `lda`, to a level comparable with `ctm`. `cluster` improves slightly over BNC, whereas `hca` drops considerably (despite being designed specifically to deal with word burstiness in the longer documents characteristic of BNC). Figure 4.1 shows boxplots for topic-level model precision, and reflects a similar trend.

Looking next to TLO in Table 4.4, we see a totally different picture, with `cluster` being rated as the best topic model by a clear margin. This exposes a flaw in the TLO formulation, in the case of adversarial topic models such as `cluster` which assign near-uniform probabilities across all topics. This results in the difference in probability mass being very close to the upperbound of zero in all cases, meaning that even for random topic selection, TLO is near perfect. We can also see this in Figure 4.2, where the boxes for `cluster` have nearly zero range. Indeed, if we combined the results for TLO with those for topic coherence, we would (very wrongly!) conclude that `cluster` performs best over

Topic Model	Average rating	
	APNEWS	BNC
lda	1.26	1.01
hca	0.95	0.90
ctm	0.96	1.02
ntm	0.36	0.46
cluster	0.41	0.66

Table 4.5: Top-1 document–topic rating for each topic model

both document collections. More encouragingly, for the other four topic models, the results for TLO are much more consistent with those based on model precision.

4.4.2 Direct Annotation of Topic Assignments

Newman et al. (2010b) proposed a more direct approach to topic coherence, by asking people to rate topics based directly on the top-n words. Taking inspiration from their methodology, we propose to directly annotate each topic assigned to a target document. We present the human annotators with the target document and the top-ranked (highest probability) topic from each of the five topic models, and ask them to rate each topic on an ordinal scale of 0–3. At the model level, we take the mean rating over all document–topic pairings for that topic model (based, once again, on 100 documents per collection).[3] We summarise the findings in Table 4.5.

We observe that, in the case of APNEWS, lda does considerably better than ctm and hca, whereas for BNC, lda and ctm are quite close, with hca close behind. cluster and ntm do poorly across both datasets. The overall trend of APNEWS of lda $>$ ctm $>$ hca $>$ cluster $>$ ntm is consistent with the model precision results in Table 4.3. In the case of BNC, the observation of ctm $\approx$ lda $>$ hca $>$ cluster $>$ ntm is also broadly the same, except that hca does not do as well over the topic intrusion task. Here, we are more interested in the relative performance of topic models than absolute numbers, although the low absolute scores are an indication that does tell us that that it is a difficult problem to annotate.

Broadly combined across the two evaluation methodologies, lda and ctm are top-performing, hca gets mixed results, and cluster and ntm are the lowest performers. These results generally agree with the model precision findings, demonstrating that model precision is a more robust metric than TLO.

[3]The 100 documents used for this task were different to the ones used in Section 4.4.1.

4.5 Automatic Evaluation

A limitation of the topic intrusion task is that it requires manual annotation, making it ill-suited for large-scale or automatic evaluation. We present the first attempt to automate the prediction of the intruder topic, with the aim of developing an approach to topic model evaluation which complements topic coherence (as motivated in Sections 4.3 and 4.4).

4.5.1 Methodology

We build a support vector regression (SVR) model (Joachims, 2006) to rank topics given a document to select the intruder topic. We first explain an intuition of the features that are driving the SVR.

To rank topics for a document, we need to first compute the probability of a topic t given document d, i.e. $P(t|d)$. We can invert the condition using Bayes rule:

$$\begin{aligned} P(t|d) &= \frac{P(d|t)P(t)}{P(d)} \\ &\propto P(d|t)P(t) \end{aligned}$$

We can omit $P(d)$ as the probability of document d is constant for the topics that we are ranking.

Next we represent topic t using its top-N highest probability words, giving:

$$\begin{aligned} P(t|d) &\propto P(d|w_1, ..., w_N)P(w_1, ..., w_N) \\ &\propto \log P(d|w_1, ..., w_N) + \\ &\quad \log P(w_1, ..., w_N) \end{aligned}$$

The first term $\log P(d|w_1, ..., w_N)$ can be interpreted from an information retrieval perspective, where we are computing the relevance of document d given query terms w_1, w_2, ..., w_N. This term constitutes the first feature for the SVR. We use Indri[4] to index the document collection, and compute $\log P(d|w_1, ..., w_N)$ given a set of query words and a document.[5]

We estimate the second term, $\log P(w_1, ..., w_N)$, using the pairwise probability of the topic words:

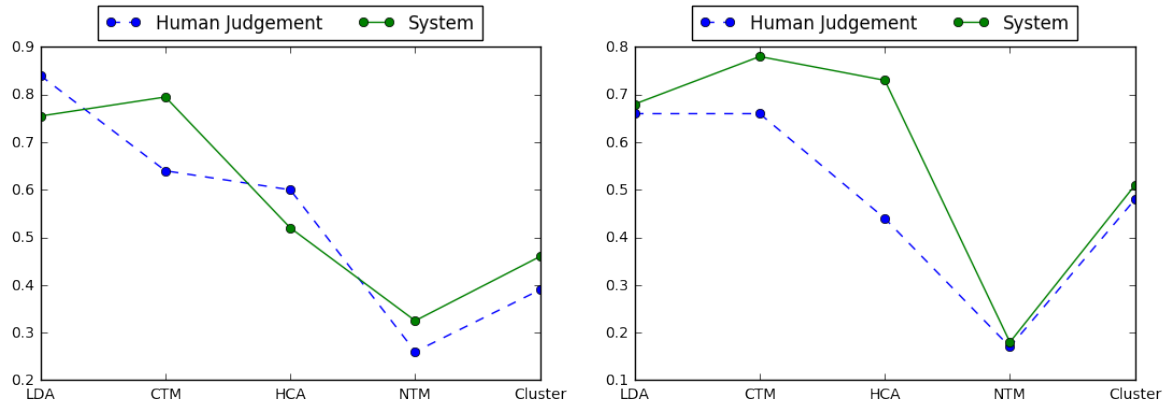
$$\sum_{0<i\leq m} \sum_{i+1\leq j\leq m} \log \frac{\#(w_i, w_j)}{\#(\cdot)}$$

where m denotes the number of topic words used, $\#(w_i, w_j)$ the number of documents where word w_i and w_j co-occur, and $\#(\cdot)$ is the total number of documents. We explore using two values of m here, 5 and 10.[6] These two values constitute the second and third

[4] http://www.lemurproject.org

[5] $N = 10$.

[6] That is, if $m = 5$, we compute pairwise probabilities using the top-5 topic words.



(a) APNEWS (b) BNC

Figure 4.3: Mean Model Precision Comparison

features of the SVR.

To train the SVR, we sample 1700 random documents and split them into 1600/100 documents for the training and test partitions respectively. The test documents are the same 100 documents that were previously used for intruder topics (Section 4.4.1). As the intruder topics are artificially generated, we can sample additional documents to create a larger training set for the SVR; the ability to generate arbitrary training data is a strength of our method.

We pool together all 5 topic models when training the SVR, therefore generating 8000 training and 500 development and testing instances for each dataset. For each document, the SVR is trained to rank the topics in terms of their likelihood of being an intruder topic.[7] The top-rank topic is selected as the system predicted intruder word, and model precision is computed as before (Section 4.4.1).[8]

4.5.2 System results

In Figure 4.3, we present the human vs. system mean model precision on the test partition for each of the topic models. We see that the the trend line for the system model precision very closely tracks that of human model precision. In general, the best systems — `lda` and `ctm` — and the worst systems — `ntm` and `cluster` — are predicted correctly. The correlation between the two is very high, at $r = 0.88$ and 0.87 for APNEWS and BNC,

[7] We use the default hyper-parameter values for the SVR ($C = 0.01$) and hence do no need require a development set for tuning.

[8] Note that system model precision is a binary value as there is only 1 system — as opposed to multiple annotators — for selecting an intruder word.

respectively. This suggests that the automated method is a reliable means of evaluating document-level topic model quality.

4.6 Discussion

To better understand the differences between human and system-predicted intruder topics, we present a number of documents and their associated topics in Table 4.6, focusing specifically on: (a) intruder topics that humans struggle to identify but our automatic method reliably detects; and (b) conversely, intruder topics which humans readily identify but our method struggles to detect.

Looking at the topics across the two types of errors, we notice that there are often multiple "bad" topics for these documents: occasionally the annotators are able to single out the worst topic while the system fails (1st and 2nd document), but sometimes the opposite happens (3rd and 4th document). In the first case, the top-ranking topic (*church, gay, ...*) from the topic model is associated with the document because of the service, but actually capturing a very different aspect of religion to what is discussed in the document, which leads our method astray. A similar effect is seen with the second document. In the case of the third and fourth documents, there is actually content further down in the document which is relevant to the topics the human annotators select, but it is not apparent in the document snippet presented to the annotators. That is, the effect is caused by resource limitations for the annotation task, that our automated method does not suffer from.

When we aggregate the top-level model precision values for a topic model, these differences are averaged out (hence the strong correlation in Section 4.5.2), but these qualitative analyses reveal that there are still slight disparities between human annotators and the automated method in intruder topic selection.

4.7 Summary

In this chapter, we demonstrated empirically that there can be large discrepancies between topic coherence and document–topic associations. By way of designing an artificial topic model, we showed that a topic model can simultaneously produce topics that are coherent but be largely undescriptive of the document collection. Finally, we propose a method to automatically predict document-level topic quality and found encouraging correlation with manual evaluation, suggesting that it can be used as an alternative approach for extrinsic topic model evaluation.

Error Type: High human MP Low system MP	**Document**	more than 2,000 attendees are expected to attend public funeral services for former nevada gov. kenny guinn . a catholic mass on tuesday morning will be followed by a memorial reception at palace station . the two-term governor who served from 1999 to 2007 died thursday after falling from the roof of his las vegas home while making repairs . he was 73 . guinn 's former chief of staff pete ernaut says attendance to the services will be limited only by the size of the venues . services start at 10 a.m. at st. joseph , husband of mary roman catholic church ...
	Topics	0: church gay marriage religious catholic same-sex couples pastor members bishop 1: casino las vegas nevada gambling casinos ford vehicles cars car 2: died family funeral honor memorial father death wife cemetery son X: students college student campus education tuition universities colleges high degree
	Document	the milwaukee art museum is exhibiting more than 70 works done by 19th century portrait painter thomas sully . it 's the first retrospective of the artist in 30 years and the first to present the artist 's portraits and subject pictures . sully was known for employing drama and theatricality to his works . in some of his full-length portraits , he composed his figures as if they were onstage . some of his subjects even seem to be trying to directly engage the viewer . milwaukee art museum director daniel keegan says the exhibit provides a new look ...
	Topics	0: china art chinese arts artist painting artists cuba world beijing 1: show music film movie won festival tickets game band play 2: online information internet book video media facebook phone computer technology X: kelley family letter leave absence left united jay weeks director
Error Type: Low human MP High system MP	**Document**	(ap) ? the west virginia lottery is celebrating its 28th birthday by doing what it does best : awarding large sums of money . the lottery will mark the milestone on thursday by giving away prizes of $ 1 million , $ 100,000 and $ 10,000 . the three finalists were selected out of thousands of entries from the lottery 's monopoly millionaire instant game . the finalists are josh schoolcraft of given , douglas schafer of wheeling and todd kingrey of charleston . all three are due at lottery headquarters in charleston to collect their winnings ...
	Topics	0: jackpot powerball mega lottery lotto jackpots prizes ticket megaplier tickets 1: mingo earl wheeling virginia ap charleston wvu huntington coalfields rockefeller 2: museum artifacts exhibit paintings artwork historical curator sculpture exhibition exhibits X: abercrombie ridley solace daley enclosures hobbyists hawaiian seventeen secondhand probate
	Document	a 75-year-old driver has died after a collision near o'neill in northern nebraska . the holt county sheriff 's office says the accident occurred wednesday afternoon , less than a mile east of o'neill . the office says thomas schneider halted at a stop sign and then turned east onto nebraska highway 108 . but he apparently turned too wide and went into the oncoming lane . his vehicle struck a westbound vehicle driven by 52-year-old gerald kemp , of niobrara . schneider was pronounced at the scene . the sheriff 's office says kemp suffered no visible injuries ...
	Topics	0: officers shot car shooting officer sheriff woman died killed hospital 1: service weather area storm miles airport snow river bridge emergency 2: prison prosecutors charges guilty trial judge case charged murder pleaded X: toll road rocky carpenter hogan indiana long harvey private director

Table 4.6: Document and topic examples for two types of errors. "MP" denotes model precision, "X" the intruder topic, and the indices the ranking of the topics. Pink (yellow) highlighted topics are those incorrectly selected by the system (humans) as intruder topics.

Chapter 5

Automatic Labelling of Topics with Neural Embeddings

5.1 Introduction

In Section 2.3 and Section 4 we addressed the problem of evaluation on topic model quality. The next segment of the problem statement is is how to present the output of a topic model to the human user. When the output is presented to a human user, a fundamental concern is the best way of presenting the rich information generated by the topic model, in particular, the topics themselves, which provide the primary insights into the document collection. The de-facto topic representation has been a simple term list, in the form of the top-10 terms in a given topic, ranked in descending order of $\Pr(w_j|t_i)$. The cognitive overhead in interpreting the topic presented as a list of terms can be high, and has led to interest in the task of generating labels for topics, e.g. in the form of textual descriptions as we discussed in Section 2.4 (Mei et al., 2007; Lau et al., 2011; Kou et al., 2015), visual representations of the topic words (Smith et al., to appear), or images (Aletras and Stevenson, 2013a). In the former case, for example, rather than the top-10 terms of ⟨*school, student, university, college, teacher, class, education, learn, high, program* ⟩, a possible textual label could be simply EDUCATION. Recent work has shown that, in the context of a timed information retrieval (IR) task, automatically-generated textual labels are easier for humans to interpret than the top-10 terms, and lead to equivalent-quality relevance judgements (Aletras et al., 2014). Despite this, the accuracy of state-of-the-art topic generation methods is far from perfect, providing the motivation for this work.

In this chapter, we propose an approach to topic labelling based on word and document embeddings, which both automatically generates label candidates given a topic input, and ranks the candidates in either an unsupervised or supervised manner, to produce the final topic label.

5.2 Methodology

In Section 2.4, we mentioned Lau et al. (2011)'s method of automatic labelling of topics. Building on it our method is made up of two steps: (1) topic label generation based on English Wikipedia; and (2) topic label ranking, based on a supervised learn-to-rank model. We detail each of these steps below.

5.2.1 Candidate Generation

To match topics to Wikipedia articles,[1] Lau et al. (2011) used an IR approach, by querying English Wikipedia with the top-N topic terms. However, in order to do this, they required external resources (two search APIs, one of which is no longer publicly available), limiting the general-purpose utility of the method. We propose an alternative approach: precomputing distributed representations of the topic terms and article titles using `word2vec` and `doc2vec`.

In Section 2.2, we discussed about neural embedding models like `word2vec` (Mikolov et al., 2013b) and `doc2vec` (Le and Mikolov, 2014). Here, we will use `word2vec` as a means of generating topic term and label representations. As with `word2vec`, we will use `doc2vec` as an alternative means of generating topic term and label representations. To this end, we train a `doc2vec` model on the English Wikipedia articles, and represent the embedding of a Wikipedia title by its document embedding. As `doc2vec` runs `word2vec` internally, word embeddings are also learnt during the training. Given the top-N topic terms, the topic embedding is represented by these terms' word embeddings. Based on the findings of Lau and Baldwin (2016) that the simpler `dbow` has less parameters, trains faster, and performs better than `dmpv` in several extrinsic tasks, we experiment only with `dbow`.[2] In terms of hyper-parameter settings, we follow the recommendations of Lau and Baldwin (2016).[3]

In addition to `doc2vec`, we also experiment with `word2vec` to generate embeddings for Wikipedia titles. By treating titles as a single token (e.g. concatenating *financial crisis* into *financial_crisis*) and greedily tokenising the text of all of the Wikipedia articles, we can then generate word embeddings for the titles. For `word2vec`, we use the `skip-gram` implementation exclusively.[4]

[1] As of 2016 there are over 5 million documents in English Wikipedia.

[2] We use Gensim's implementation of both `doc2vec` and `word2vec` for all experiments: `https://radimrehurek.com/gensim/`.

[3] `doc2vec` hyper-parameters: sub-sampling threshold $= 10^{-5}$, vector size $= 300$, window size $= 15$, negative sample size $= 5$, and training epochs $= 20$.

[4] `word2vec` hyper-parameters: sub-sampling threshold $= 10^{-5}$, vector size $= 300$, window size $= 5$, negative sample size $= 5$, and training epochs $= 100$.

For both doc2vec and word2vec, we first pre-process English Wikipedia,[5] using Wiki Extractor to clean and extract Wikipedia articles from the original dump.[6] We then tokenise words with the Stanford CoreNLP Parser (Klein and Manning, 2003), and lowercase all words. We additionally filter out articles where the article body is made up of less than 40 words, and also disambiguation pages. We also remove titles whose length is longer than 4 words, as they are often too specific or inappropriate as topic labels (e.g. *List of Presidents of the United States*). For word2vec, we remove any parenthesised sub-component of an article title — e.g. in the case of *Democratic Party (United States)*, we remove *(United States)* to generate *Democratic Party* — as we would not expect to find verbatim usages of the full title. This has the potential side-effect of mapping multiple articles onto a single ambiguous title, resulting in multiple representations for *Democratic Party*. While acknowledging that there are instances where the more specific title may be appropriate as a label, the generalised version is always going to be a hypernym of the original, and thus appropriate as a label candidate.

Given a topic, we measure the relevance of each title embedding (generated by either doc2vec or word2vec) based on the pairwise cosine similarity with each of the word embeddings for the top-10 topic terms, and aggregate by taking the arithmetic mean. Formally, the doc2vec relevance (rel_{d2v}) and word2vec relevance (rel_{w2v}) of a title a and a topic T is given as follows:

$$rel_{d2v}(a, T) = \frac{1}{|T|} \sum_{v \in T} \cos\left(E^{d}_{d2v}(a), E^{w}_{d2v}(v)\right) \tag{5.1}$$

$$rel_{w2v}(a, T) = \frac{1}{|T|} \sum_{v \in T} \cos\left(E^{w}_{w2v}(a), E^{w}_{w2v}(v)\right) \tag{5.2}$$

where $E^{d}_{d2v}(x)$ is the document embedding of title x generated by doc2vec; $E^{w}_{d2v}(y)$ is the word embedding of word y generated by doc2vec; $E^{w}_{w2v}(z)$ is the word embedding of word z generated by word2vec; $v \in T$ is a topic term; $|T|$ is the number of topic terms (10 in our experiments); and $\cos(\vec{x}, \vec{y})$ is the cosine similarity function.

The idea behind using both doc2vec and word2vec to generate title embeddings is that we observe that the two models favour different types of labels: doc2vec tends to favour fine-grained concepts, while word2vec favours more generic or abstract labels. As an illustration of this, in Table 5.1 we present one of the actual topics used later in our evaluation, and the top-5 article titles based on doc2vec and word2vec. This dichotomy is rooted in the differences in the modelling of context in the two models. In doc2vec, the

[5] The English Wikipedia dump used in all experiments is dated 2015-12-01.

[6] https://github.com/attardi/wikiextractor/

Top-10 Topic Terms	word2vec Labels	doc2vec Labels
	software	microsoft visual studio
blogs, vmware, server, virtual, oracle,	desktop	desktop virtualization
update, virtualization, application,	operating system	microsoft exchange server
infrastructure, management	virtualization	cloud computing
	middleware	windows server 2008

Table 5.1: The top-5 labels generated using doc2vec and word2vec title embeddings for the topic provided

title embedding is determined by the words that belong to the title, each of which is in turn determined by its context of use; it thus directly captures the compositional semantics of the title. With our word2vec method, on the other hand, the title embedding is determined directly by the neighbouring words of the title token in text, oblivious to the composition of words in the title.

To combine the strengths of doc2vec and word2vec, for each topic we generate a combined candidate ranking by summing the relevance scores using top-100 candidates from doc2vec and word2vec:[7]

$$rel_{d2v+w2v}(a, T) = rel_{d2v}(a, T) + rel_{w2v}(a, T) \tag{5.3}$$

5.2.2 Candidate Ranking

The next step after candidate generation is to re-rank them based on a supervised learn-to-rank model, in an attempt to improve the quality of the top-ranking candidates.

The first feature used in the supervised reranker is *LetterTrigram*, and based on the finding of Kou et al. (2015) that letter trigram vectors are an effective means of ranking topic labels. Our implementation of their method is based on measuring the overlap of letter trigrams between a given topic label and the topic words. For each topic, we first convert each topic label and topic words into multinomial distributions over letter trigrams, based on simple maximum likelihood estimation.[8] We then rank the labels based on their cosine similarity with the topic words. The rank value constitutes the first feature of the supervised learn-to-rank model. Additionally, this ranking by letter trigram method also

[7]From preliminary experiments we found that summing only the top-100 candidates from doc2vec and word2vec is better than summing all candidates. As we remove the parenthesised sub-component of an article title for word2vec (*Democratic Party (United States)* $\longrightarrow$ *Democratic Party*) , we observe that these titles tend to be very general and can ocassionally produce very high cosine similarity and skew the combined score for a number of similar labels (e.g. causing *Democratic Party* from a host of countries (*Democratic Party (United States)*, *Democratic Party (Australia)*, etc) to appear in the top ranking).

[8]For topic words, the letter trigrams are generated by parsing each of the topic words as separate strings rather than one concatenated string.

forms our unsupervised baseline, as we found that it to have the best unsupervised ranking performance of all our features, consistent with the findings of Kou et al. (2015).[9]

The second feature is *PageRank* (Page et al., 1998), in an attempt to prefer labels which represent more "core" concepts in Wikipedia. *PageRank* uses directed links to estimate the significance of a document, based on the probability of a random web surfer visiting a web page by either following hyperlinks or randomly transporting to a new page. We construct a directed graph from Wikipedia based on hyperlinks within the article text, and from this, compute a *PageRank* value for each Wikipedia article (and hence, title).[10]

Our last two features are lexical features proposed by Lau et al. (2011): (1) *NumWords*, which is simply the number of words in the candidate label (e.g. *operating system* has 2 words); and (2) *TopicOverlap*, which is the lexical overlap between the candidate label and the top-10 topic terms (e.g. *desktop virtualization* has a *TopicOverlap* score of 1 in our example from Table 5.1).

Given these features and a gold standard order of candidates (detailed in Section 5.3.1), we train a support vector regression model (SVR: Joachims (2006)) over these four features.

5.3 Datasets

For direct comparison with Lau et al. (2011), we use the same set of topics they used in their experiments. These were generated from 4 different domains: BLOGS, BOOKS, NEWS and PUBMED. In general, BLOGS, BOOKS and NEWS cover wide-ranging topics from product reviews to religion to finance and entertainment, whereas PubMed is medical-domain specific.

BLOGS is made up of 120k blog articles from the Spinn3r blog dataset; BOOKS is made up of 1k English language books from the Internet Archive American Libraries collection; NEWS is made up of 29k New York Times articles from English Gigaword; and PUBMED is made up of 77k PubMed biomedical abstracts. Lau et al. (2011) ran LDA on these documents and generated 100 topics for each domain. They filtered incoherent topics using an automated approach (Newman et al., 2010c), resulting in 45, 38, 60, 85 topics for BLOGS, BOOKS, NEWS and PUBMED, respectively.

[9]Note that we do not make use of the noun chunk-based label generation methodology of Kou et al. (2015), in line with the findings of Lau et al. (2011) that Wikipedia titles give rise to better label candidates than n-grams extracted from the topic-modelled documents.

[10]We use the following implementation for *PageRank*: https://www.nayuki.io/page/computing-wikipedias-internal-pageranks/

5.3.1 Gold Standard Judgements

To evaluate our method and train the supervised model, gold-standard ratings of the candidates are required. To this end, we used CrowdFlower to collect human judgements.[11] We follow the approach of Lau et al. (2011), presenting 10 pairings of topic and candidate label, and asking human judges to rate the label on an ordinal scale of 0–3 where 0 indicates a completely inappropriate label, and 3 indicates a very good label for the given topic.

To control for annotation quality, we make use of the original annotations released by Lau et al. (2011). We select labels with a mean rating ≥ 2.5 (good labels) and ≤ 0.5 (bad labels) to serve as controls in our tasks. We include an additional topic–label control pair in addition to the 10 topic–label pairs in a HIT. Control pairs are selected randomly without replacement, and randomly injected into the HIT. To pass quality control, a worker is required to rate bad labels ≤ 1.0 and good labels ≥ 2.0. A worker is filtered out if his/her overall pass rate over all control pairs is < 0.75.

Each candidate label was rated by 10 annotators. Post-filtered, we have an average of 6.4 annotations for each candidate label.[12] To aggregate the ratings for a candidate label, we compute its mean rating, and rank the candidate labels based on the mean ratings to produce the gold standard ranking for each topic.

We collect judgements for the top-19 candidates from the unsupervised ranking.[13] For candidate ranking (Section 5.2.2), we are therefore re-ranking the top-19 candidates.

5.4 Experiments

In this section we present the results of our topic labelling experiments, and compare our method with that of Lau et al. (2011). Henceforth we refer to our method as "NETL" (neural embedding topic labelling), and Lau et al. (2011) as "LGNB".

Following LGNB, we use **top-1 average rating** and **normalized discounted cumulative gain (nDCG)** (Järvelin and Kekäläinen, 2002; Croft et al., 2009) as our evaluation metrics. Top-1 average computes the mean rating of the top-ranked labels, and provides an evaluation of the absolute utility of the preferred labels. nDCG, on the other hand, measures the relative quality of the ranking, calibrated relative to the ratings of the gold-standard ranking. Similarly to LGNB, we compute nDCG for the top-1 (nDCG-1), top-3 (nDCG-3), and top-5 (nDCG-5) ranked labels.

[11] https://www.crowdflower.com/

[12] Post-filering, some candidates ended up with less than 3 annotations; these candidates were posted for another annotation round to gather more annotations.

[13] Ideally we would have liked to have collected judgements for as many candidates as possible, but due to budget constraints we were only able to have the top-19 annotated.

Test Domain	Training	Top-1 Avg.		nDCG-1		nDCG-3		nDCG-5	
		LGNB	NETL	LGNB	NETL	LGNB	NETL	LGNB	NETL
BLOGS	Baseline	1.84	**1.91**	0.75	**0.77**	0.77	**0.82**	0.79	**0.83**
	In-Domain	1.98	**2.00**	0.81	0.81	0.82	**0.85**	0.83	0.84
	Cross-domain: BOOKS	1.88	**1.91**	0.77	0.78	0.81	**0.83**	0.83	0.83
	Cross-domain: NEWS	**1.97**	1.92	**0.80**	0.78	0.83	0.84	0.83	0.84
	Cross-domain: PUBMED	**1.95**	1.90	**0.80**	0.77	0.82	0.83	0.83	0.83
	Cross-domain: All 3	—	1.92	—	0.78	—	0.84	—	0.84
	Upper Bound	2.45	**2.48**	1.00	1.00	1.00	1.00	1.00	1.00
BOOKS	Baseline	1.75	**1.97**	0.77	0.78	0.77	**0.82**	0.79	**0.83**
	In-Domain	1.91	**1.99**	**0.84**	0.82	0.81	0.82	0.83	0.84
	Cross-domain: BLOGS	1.82	**2.02**	0.79	**0.83**	0.81	0.82	0.82	**0.84**
	Cross-domain: NEWS	1.82	**1.99**	0.79	**0.81**	0.81	0.82	0.83	0.84
	Cross-domain: PUBMED	1.87	**1.97**	0.81	0.80	0.82	0.82	0.83	0.84
	Cross-domain: All 3	—	2.03	—	0.83	—	0.83	—	0.84
	Upper Bound	2.29	**2.49**	1.00	1.00	1.00	1.00	1.00	1.00
NEWS	Baseline	1.96	**2.04**	0.80	**0.82**	0.79	**0.84**	0.78	**0.85**
	In-Domain	2.02	2.02	**0.82**	0.80	0.82	**0.84**	0.84	0.85
	Cross-domain: BLOGS	2.03	2.03	**0.83**	0.81	0.82	**0.84**	0.84	0.85
	Cross-domain: BOOKS	**2.01**	1.98	**0.82**	0.79	0.82	0.83	0.83	0.84
	Cross-domain: PUBMED	2.01	2.00	**0.82**	0.79	0.82	0.83	0.83	0.84
	Cross-domain: All 3	—	1.99	—	0.79	—	0.84	—	0.84
	Upper Bound	2.45	**2.56**	1.00	1.00	1.00	1.00	1.00	1.00
PUBMED	Baseline	1.73	**1.94**	0.75	**0.79**	0.77	**0.80**	0.79	**0.82**
	In-Domain	1.79	**1.99**	0.77	**0.81**	0.82	0.81	**0.84**	0.82
	Cross-domain: BLOGS	1.80	**1.98**	0.78	**0.80**	0.82	0.81	**0.84**	0.82
	Cross-domain: BOOKS	1.77	**1.98**	0.77	**0.80**	0.82	0.81	0.83	0.82
	Cross-domain: NEWS	1.79	**1.98**	0.77	**0.80**	0.82	0.81	**0.84**	0.82
	Cross-domain: All 3	—	2.01	—	0.81	—	0.81	—	0.82
	Upper Bound	2.31	**2.51**	1.00	1.00	1.00	1.00	1.00	1.00

Table 5.2: Results across the four domains. Boldface indicates the better system between NETL and LGNB (with an absolute difference > 0.01).

5.4.1 Results

Following LGNB, we present results for: (a) the unsupervised ranker (based on letter trigrams); (b) the supervised re-ranker in-domain, based on 10-fold cross validation, averaged over 10 runs with different partitionings; (c) the supervised re-ranker cross-domain; and (d) the upper bound, based on a perfect ranking of the candidates. For cross-domain learning, we train our model using one domain and test it on a different domain, or alternatively combine data from three domains and test on the remaining fourth domain, e.g. training on BOOKS +NEWS +PUBMED and testing on BLOGS. Cross-domain results give us a more accurate picture of the performance of our methodology in real-world applications (where it would be unrealistic to expect that there would be manual annotations of label candidates for that domain). We primarily use the in-domain results to gauge the relative quality of the cross-domain results.

We present the results in Table 5.2, displaying the performance of NETL and LGNB

Domain	Topic Terms	Label Candidate
BLOGS	vmware server virtual oracle update virtualisation application infrastructure management microsoft	virtualisation
BOOKS	church archway building window gothic nave side value tower	church architecture
NEWS	investigation fbi official department federal agent investigator charge attorney evidence	criminal investigation
PUBMED	rate population prevalence study incidence datum increase mortality age death	mortality rate

Table 5.3: A sample of topics and their topic labels generated by NETL.

side by side for ease of comparison.[14] For each domain, the unsupervised baseline of NETL is based on the overlap of letter trigrams of the generated candidates with topic words (see Section 5.2.2). The unsupervised baseline of LGNB ranks the labels using a lexical association measure (Pearson's χ^2).

Looking at the performance of our method, we can see that the supervised system improves over the unsupervised baseline across all domains with the exception of a small drop observed in NEWS. Surprisingly, there is relatively little difference between the in-domain and cross-domain results for our method (but greater disparity for LGNB, especially over BOOKS; for NEWS, our cross-domain models actually outperform the in-domain model). The most consistent cross-domain results are generated when we combine all 3 domains, an unsurprising result given that it has access to the most training data, but encouraging in terms of having a single model which performs consistently across a range of domains.

We next compare NETL to LGNB, first focusing on the top-1 average rating metric. The most striking difference is the large improvement over PUBMED. LGNB attributed the poor performance over PUBMED to it being more domain-specific (and a poorer fit to Wikipedia) than the other domains, and suggested the need of biomedical experts for annotation. Our experiments found, however, that the performance of PUBMED is comparable to other domains. Additionally, the improvement in BOOKS is also quite substantial. Overall, NETL is more consistent across different domains and outperforms LGNB over 2 domains (BOOKS and PUBMED), and the difference between NETL and LGNB is small for NEWS and BLOGS. The other observation is the upper bound performance of NETL is uniformly better than that of LGNB, implying we are also generating better label candidates (we revisit this in detail in Section 5.4.2.1).

Moving to nDCG, the performance difference for nDCG-3/5 is largely indistinguishable for the two systems. LGNB, however, outperforms NETL in NEWS for nDCG-1 whereas

[14] LGNB results are taken directly from the original paper.

NETL does better inPUBMED for nDCG-1 .

To give a sense of the sort of labels generated by NETL, we present a few topics and their top-ranked labels in Table 5.3.

5.4.2 Breaking Down NETL vs. LGNB

The results for NETL and LGNB in Table 5.2 conflate the candidate label selection and ranking steps, making it hard to get a sense of the relative impact of the different design choices implicit in the two sub-tasks. To provide a better comparison between the two methodologies, we present experiments evaluating the candidate generation and ranking method of the two systems separately.

5.4.2.1 Candidate Generation

In Table 5.2 we saw that NETL has a higher upper bound than that of LGNB, suggesting that the generated candidate labels were on average better. This is despite the average number of topic label candidates actually being higher for LGNB (25 vs. 19). Here, we present a more rigorous evaluation of the candidate generation method of both systems.

For each topic, we determine the mean, maximum and minimum label ratings for a given topic, and plot them in boxplots in Figure 5.1, Figure 5.2, Figure 5.3 aggregated per domain. The mean rating boxplot shows the average quality of candidates, while the maximum (minimum) rating boxplot reveals the average best (worst) quality of candidates that are generated by the two systems.

Looking at the boxplots, we see very clearly that NETL generates on average higher-quality candidates. Across all domains for mean, maximum and minimum ratings, the difference is substantial.

To provide a quantitative evaluation, we conduct one-sided paired t-tests to test the difference for all pairs in the boxplots. Except for the maximum ratings on BLOGS, all tests are significant ($p < 0.05$). These results demonstrate that NETL generates better candidates than LGNB (in all of the best-case, average-case and worst-case scenarios).

5.4.2.2 Candidate Ranking

Next, we directly compare the ranking method of NETL and LGNB. Using candidates generated by NETL, we re-rank the candidates using the ranking method of each of LGNB and NETL, and compare the results.

Both LGNB and NETL train an SVR re-ranker, using a partially-overlapping set of features. For LGNB, the ranking methodology uses 7 lexical association measures (PMI,

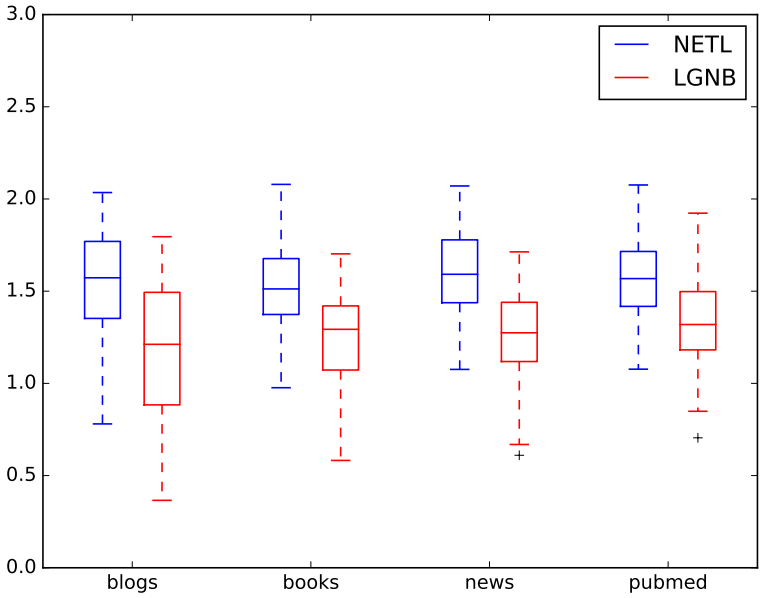


Figure 5.1: Mean Ratings of candidates generated by NETL and LGNB

Student's t-test, Dice's coefficient, Pearson's χ^2 test, log likelihood ratio, conditional and reverse conditional probability), 2 lexical features (the same 2 features that NETL uses: *NumWords* and *TopicOverlap*), and a search engine score based on Zettair. NETL, on the other hand, uses only 4 features: *LetterTrigram*, *PageRank*, *TopicOverlap* and *NumWords*.

For LGNB, we exclude the Zettair search engine score feature (as it was found to be an unimportant feature), and generate the lexical association features by parsing English Wikipedia. We train 2 SVR models using LGNB and NETL features. Results are presented in Table 5.4.

Using the same candidates, we see that NETL's features produce better rankings, outperforming LGNB's features across all domains. This shows that not only does NETL generate better candidates, but also ranks them better than LGNB.

5.5 Discussion

To better understand the contribution of each feature in NETL, we perform feature ablation tests (Table 5.5). An interesting observation is that different features appear to have different impact depending on the domain. Looking at BLOGS, we find that there is a considerable drop in top-1 average rating when we remove the *PageRank* feature. Similarly, ablating *LetterTrigram* appears to have a significant impact on PUBMED as well as some influence

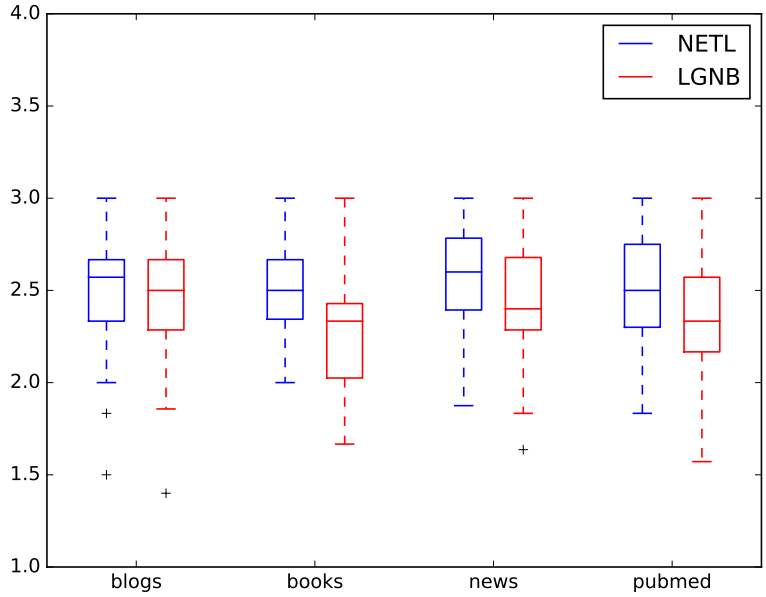


Figure 5.2: Maximum Ratings of candidates generated by NETL and LGNB

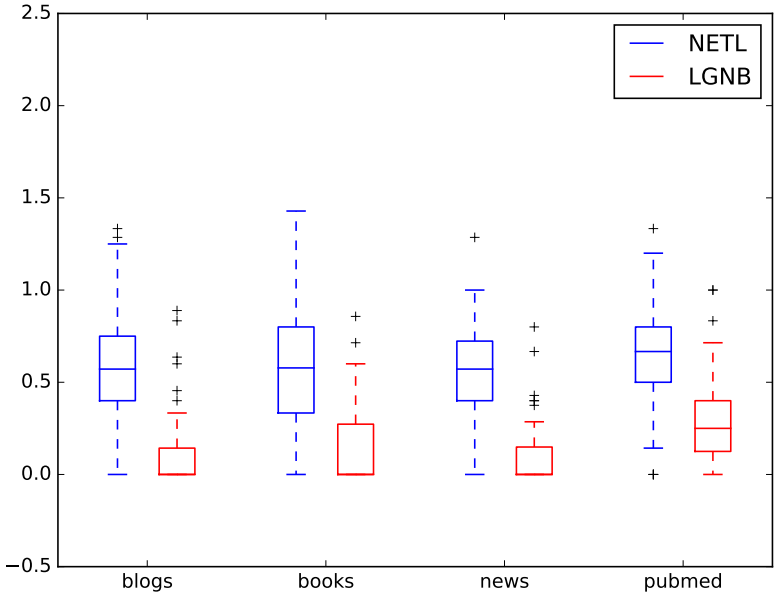


Figure 5.3: Minimum Ratings of candidates generated by NETL and LGNB

Test Domain	Features	Top-1 Avg.	nDCG-1	nDCG-3	nDCG-5
BLOGS	LGNB	1.92	0.79	0.81	0.82
	NETL	**2.00**	**0.81**	**0.85**	**0.84**
BOOKS	LGNB	1.86	0.77	0.79	0.80
	NETL	**1.99**	**0.82**	**0.82**	**0.84**
NEWS	LGNB	1.87	0.75	0.79	0.81
	NETL	**2.02**	**0.80**	**0.84**	**0.85**
PUBMED	LGNB	1.89	0.77	0.79	0.81
	NETL	**1.99**	**0.81**	**0.81**	0.82

Table 5.4: Comparison of ranking performance with NETL features and LGNB features. Boldface indicates the better system between NETL and LGNB (with an absolute difference > 0.01).

Test Domain	BLOGS	BOOKS	NEWS	PUBMED
All Features	2.00	1.99	2.02	1.99
$-$***LetterTrigram***	1.99 ($-.01$)	1.96 ($-.03$)	2.02 ($\pm.00$)	1.93 ($-.06$)
$-$***PageRank***	1.93 ($-.07$)	1.980 ($-.01$)	2.00 ($-.02$)	2.00 ($+.01$)
$-$***TopicOverlap***	2.00 ($\pm.00$)	2.03 ($+.04$)	2.04 ($+.02$)	1.95 ($-.04$)
$-$***NumWords***	1.97 ($-.03$)	2.02 ($+.03$)	2.02 ($\pm.00$)	2.00 ($+.01$)

Table 5.5: Feature ablation results based on in-domain top-1 average ratings.

on BOOKS. As far as NEWS is concerned, we observe feature ablation does not play a big role. These observations indicate there is some degree of complementarity between these features, and that combining them produces robust and consistent performance across different domains.

Additionally, we explored using different numbers of topic terms when computing topic and title relevance for candidate ranking (we tested using top-5/10/15/20 topic terms). In general, we find that performance drops with the increase in topic terms. We also experiment with weighting each topic term with its word probability. We observed an improvement, although the difference is so marginal that we omit the results from the paper. Lastly, we tried computing relevance by first computing the centroid of topic terms before computing the cosine similarity with a candidate title. Again, we found little gain with this approach.

One feature type that we expect would have high utility is graph connectivity over the graphical structure of the Wikipedia categories or similar, along the lines of Hulpus et al. (2013). We leave this to future work. Methods based on keyphrase extraction such as Zhao et al. (2011) are also potentially worth exploring, although it remains to be seen whether notions such as "interestingness" benefit topic label selection.

5.6 Summary

In this chapter, we propose a neural embedding approach to automatically label topics using Wikipedia titles. Our methodology combines document and word embeddings to select the most relevant labels for topics. Compare to a state-of-the-art competitor system, our model is simpler, more efficient, and achieves better results across a range of domains.

Chapter 6

Conclusion and Future Work

In this chapter, we summarize the findings of each chapter and propose future work.

In chapter 2, we gave an insight on topic models. We traced back the first topic model and then gave a detailed account on some of the current topic models. Next we talked about topic model evaluation both at intrinsic level with metrics like perplexity and extrinsic level with metrics like topic coherence and topic intrusion. Lastly, we gave an account on presentation of topics. We reviewed the literature that discussed different methodologies to find a succinct textual phrase or label for representation of topics.

In chapter 3, we re-introduced the neural topic model and implemented it to reproduce the results of the original paper. We detailed the neural topic model's architecture and other preprocessing details for different datasets. Next, we applied it to different datasets and saw that despite producing coherent topics it was not able to capture the broader concepts of document collection. Additionally, we figured out that there were discrepancies between topic coherence and document–topic associations.

Chapter 4 discussed about the evaluation of topic models both at the topic level and document level. We first evaluated various topic models at topic level using topic coherence. Next, we designed an artificial topic model to show that that a topic model can simultaneously produce topics that are coherent but be largely undescriptive of the document collection. We introduced the topic intrusion task to evaluate document-level topic quality of a topic model and collected human annotations for it. Additionally, we proposed an automatic evaluation to predict document-level topic quality. Finally, we argued that a good topic model should perform well on both the criteria of evaluation.

In chapter 5, we proposed the task of automatic labelling of topics. The task was to get a succinct phrase or a label that summarizes the topic represented as list of words. We divided are task into two parts: candidate generation and candidate ranking. For the first part of the task we used Wikipedia articles with neural embeddings like `word2vec` and `doc2vec`. For the second part of the task we constructed features like letter trigram, pagerank and

lexical features to train a SVR to rank the generated labels. Compared to a state-of-the-art topic labelling system, our methodology was simpler, more efficient, and found better topic labels.

6.1 Future Work

Although the thesis gave us new insights into topic models but it still leaves a few questions unanswered or with a scope of improvement.

In chapter 4, we discussed about automatic evaluation to predict document-level topic quality. The evaluation showed that it correlated with the manual annotations. But at the document level the qualitative analysis revealed that there were still disparities between human annotators and the automated method in intruder topic selection. To this end, we could collect further annotations to able to get more insight into reasoning of these disparities. Additionally, an improvement in supervised learning approach is also desired. One way suggested would be to get better features to model the the relations of document and topics in a better way for SVR. Alternatively, we could altogether try a different route by using a neural network classifier for automatic evaluation as it sidesteps the need for intensive feature engineering

In chapter 5, our method defined a new benchmark for labelling of topics. But it is still some distance from the upperbound scores of human annotations. This shows that there is still scope of improvement in the methodology especially in supervised learning and feature engineering to mimic the human annotations.

Bibliography

Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. 2014. Representing topics labels for exploring digital libraries. In *Proceedings of Digital Libraries 2014*. London, UK.

Nikolaos Aletras and Mark Stevenson. 2013a. Representing topics using images. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*. Atlanta, USA, pages 158–167.

Nikos Aletras and Mark Stevenson. 2013b. Evaluating topic coherence using distributional semantics. In *Proceedings of the Tenth International Workshop on Computational Semantics (IWCS-10)*. Potsdam, Germany, pages 13–22.

Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. 2008. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM-08)*. Washington, DC, USA, pages 3–12.

David Blei and John Lafferty. 2006. Correlated topic models. *Advances in Neural Information Processing Systems* 18.

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

Wray L Buntine and Swapnil Mishra. 2014. Experiments with non-parametric topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 881–890.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A novel neural topic model and its supervised extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, USA, pages 2210–2216.

Allison June-Barlow Chaney and David M. Blei. 2012. Visualizing topic models. In *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM 2012)*. Dublin, Ireland.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 21 (NIPS-09)*. Vancouver, Canada, pages 288–296.

Changyou Chen, Lan Du, and Wray Buntine. 2011. Sampling table configurations for the hierarchical poisson-dirichlet process. *Machine Learning and Knowledge Discovery in Databases* pages 296–311.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice*. Addison Wesley.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391.

Gabriel Doyle and Charles Elkan. 2009. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pages 281–288.

Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. 2016. Using word embedding to evaluate the coherence of topics from Twitter data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. pages 1057–1060.

Karl Grieser, Timothy Baldwin, Fabian Bohnert, and Liz Sonenberg. 2011. Using ontological and document similarity to estimate museum exhibit relatedness. *ACM Journal of Computing and Cultural Heritage* 3:10:1–10:20.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101(suppl 1):5228–5235.

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

Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. 2013. Unsupervised graph-based topic labelling using DBpedia. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. Rome, Italy, pages 465–474.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20(4).

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, USA, pages 217–226.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. Sapporo, Japan, pages 423–430.

Wanqiu Kou, Li Fang, and Timothy Baldwin. 2015. Automatic labelling of topic models using word vectors and letter trigram vectors. In *Proceedings of the 11th Asian Information Retrieval Societies Conference (AIRS 2015)*. Brisbane, Australia, pages 229–240.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany, pages 78–86.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. 2011. Automatic labelling of topic models. Portland, USA, pages 1536–1545.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the EACL (EACL 2014)*. Gothenburg, Sweden, pages 530–539.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*. Beijing, China, volume 14, pages 1188–1196.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. Baltimore, USA, pages 55–60.

Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. 2007. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*. San Jose, USA, pages 490–499.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations, 2013*. Scottsdale, USA.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013c. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. pages 3111–3119.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*. Edinburgh, UK, pages 262–272. http://www.aclweb.org/anthology/D11-1024.

David Newman, Timothy Baldwin, Lawrence Cavedon, Sarvnaz Karimi, David Martinez, and Justin Zobel. 2010a. Visualizing document collections and search results using topic mapping. *Journal of Web Semantics* 8(2–3):169–175.

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

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010c. Automatic evaluation of topic coherence. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*. Los Angeles, USA, pages 100–108.

Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank citation ranking: Bringing order to the web. Stanford Digital Libraries SIDL-WP-1999-0120.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Kevin Seppi, Niklas Elmqvist, and Leah Findlater. 2017. Evaluating visual representations for topic understanding and their effects on manually generated labels. *Transactions of the Association for Computational Linguistics* 5:1–15.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Leah Findlater, Jordan Boyd-Graber, and Niklas Elmqvist. to appear. Evaluating visual representations for topic understanding and their effects on manually generated labels. *Transactions of the Association for Computational Linguistics* .

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2004. Sharing clusters among related groups: Hierarchical dirichlet processes. In *NIPS*. pages 1385–1392.

Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*. Montreal, Canada, pages 1105–1112.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, USA, pages 424–433.

Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*. Portland, USA, pages 379–388.