# Representing Automatically Generated Topics


by

**Areej N. Alokaili**

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Computer Science

February 2021

*This thesis is dedicated to my beloved mom and dad.*

# DECLARATIONS

I, Areej N. Alokaili, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously been presented for an award at this, or any other, university. Some parts of this thesis have been previously published by the author in the following:

- IWCS 2019 (Chapters 3 and 4)
- SIGIR 2020 (Chapters 5 and 6)

Areej N. Alokaili

February 2021

# ACKNOWLEDGMENTS

First, I would like to thank Allah, without whom nothing is possible.

My sincere gratitude goes to my supervisor, Dr Mark Stevenson, for his support and guidance throughout this journey. I have been fortunate to have a kind supervisor who cared about my study and unconditionally shared his invaluable knowledge. Dr Stevenson, your constructive yet positive comments were a blessing and the primary reason I maintained emotional balance and finished this research.

I also want to thank my co-supervisor, Dr Nikolaos Aletras, for his support throughout this process, including his contributions to many discussions that helped shape my research.

Special thanks to my PhD panel committee members, Professor Rob Gaizauskas and Dr Steve Maddock, for their feedback and constructive comments. I want to express my gratitude to my examiners, Professor Aline Villavicencio and Professor Yulan He, who greatly helped improve this thesis.

I am incredibly grateful to my beloved husband Musaed, whose support and sacrifice have been invaluable. To my lovely daughter Hussah, you made me push through and work hard.

My appreciation also goes to my mom and dad; I cherish your never-ending love, support, prayers, encouragement and, most importantly, belief in me.

Many thanks to my sister Amal. I appreciate you standing by my side, our daily talks and keeping me company despite the distance.

Thank you to all my friends in Sheffield: Rehab, Amal, Fatima, Dalia, Tarfah, Rabab and Hessa. Because of you, my journey was pleasant, and I hope that our

friendship lasts a lifetime.

Finally, I am extremely grateful for the scholarship provided to me by the Saudi government represented by King Saud University, which helped me pursue this dream.

# ABSTRACT

Topic models are widely used in natural language processing (NLP). Ensuring that their output is interpretable is an essential area of research with a wide range of applications in several areas, such as the enhancement of exploratory search interfaces. Conventionally, topics are represented by their most probable words. However, these representations are often difficult for humans to interpret. Evaluating representations also presents further challenges. Ideally, humans can gauge the quality of the topics, but it is not always feasible in practical terms. This thesis addresses the limitations related to the output of the topic model in three ways.

First, it proposes and explores a range of alternative representations of topics by re-ranking topic words. Re-ranking adjusts the weights of the words and aims to identify informative words in the topics. This approach is a straightforward remedy, as topics tend to contain "noisy" words. Additionally, two approaches to evaluating the topics are proposed: (1) an automatic approach based on a document retrieval task; and (2) a crowdsourcing task. Both approaches demonstrate that re-ranking words improves topic interpretability. In addition, two alternative visual forms of the topic are explored, and a simple list of words representation shows to be more useful than a word cloud.

Second, the thesis introduces a new approach to assigning topics with short descriptive labels. Labelling topics is an important task that aims to improve access to large document collections. Previous work on the automatic assignment of labels to topics has relied on a two-stage approach: (1) retrieve candidate labels from a large pool; and then (2) re-rank candidate labels. However, these approaches can only assign candidate labels from a restricted set that may not include any suitable ones. The new approach uses a sequence-to-sequence neural-based approach to generate labels that do not have this limitation. In addition, two new synthetic datasets of pairs of topics and labels are created to train the models.

Third, this thesis conducts an empirical study on the proposed labelling approaches and performs quantitative and qualitative analyses of the generated labels. The labels are evaluated with gold labels that were rated by humans, and the labels are also evaluated with the topics themselves. The proposed approaches generate appropriate labels that are coherent and relevant to the topics.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ACRONYMS

| | |
|---|---|
| ANNs | Artificial Neural Networks. |
| CNNs | Convolutional Neural Networks. |
| EM | Expectation-Maximization. |
| FCs | Fully Connected Neural Networks. |
| GRU | Gates Recurrent Unit. |
| IDF | Inverse Document Frequency. |
| IR | information retrieval. |
| LDA | Latent Dirichlet Allocation. |
| LSA | Latent Semantic Analysis. |
| LSTM | Long Short-Term Memory. |
| MeSH | Medical Subject Headings. |
| MTurk | Amazon Mechanical Turk. |
| NLP | Natural Language Processing. |
| NYT | New York Times. |
| PLSA | Probabilistic Latent Semantic Analysis. |
| RNNs | Recurrent Neural Networks. |
| Seq2Seq | Sequence-to-Sequence. |
| SVD | Singular Value Decomposition. |
| TF-IDF | Term Frequency–Inverse Document Frequency. |
| VB | Online Variational Bayes. |

# 1

# INTRODUCTION

The amount of textual information has increased rapidly because of the continuous advancements made in computer science. Most digital information is created in an unstructured form, such as in web pages, emails, and social media posts. Therefore, ways are needed to index and organise the information into an accessible form (Boyd-Graber et al., 2017). In making online data more accessible, manual annotation mainly involves reading and understanding documents to produce a description of each, which is a comprehensive process that requires many resources. Therefore, automatic methods, such as topic models, are necessary to help annotate large amounts of data. Annotated data have various uses, including building an exploratory interface that allows the user to directly interact with the document collection (Chaney and Blei, 2012). The output of topic models has also been used widely in various applications of Natural Language Processing (NLP), such as part-of-speech (POS) (Toutanova and Johnson, 2008), sentiment analysis (or opinion mining) (Ren et al., 2016), and summarisation (Haghighi and Vanderwende, 2009). Therefore, processing and improving the output of a topic model is vital in enhancing exploratory tools and in cases where annotated data produced by the topic model are used in other

tasks. This thesis focuses on the interpretability of the results produced by the topic model.

The rest of this chapter is organised as follows. A brief background of the problem of large data collection analysis and the motivation for needing automatic approaches to the task is presented in section 1.1. In section 1.2, the scope and aim of the thesis are stated, and the main contributions are defined in section 1.3. Section 1.4 presents the structure of the thesis, and the author's previously published materials are listed in section 1.5.

## 1.1 Motivation

Several unsupervised methods are used to analyse unstructured data, such as those that discover clusters and reduce dimensionality (Murphy, 2012). This thesis concerns the management of large text data collections, where it is often useful to reduce the dimensions and produce a lower representation of the data. Topic modelling is the main statistical method used to analyse and summarise data collections (Blei et al., 2003, Hofmann, 1999).

Topic modelling is based on a statistical algorithm that aims to discover patterns in words by examining a set of documents and discovering key classes called "topics" based on the statistics of each word. The algorithm also groups and arranges the documents into the discovered topics. A document usually belongs to multiple topics in different proportions. For example, Figure 1.1 shows latent topics discovered in the sample documents by topic models. The figure also shows the occupations of topics in the document; for example, the words shown in the document belong to topics on *accidents*, *transport*, *health*, and *police*. These topic names (i.e., labels) are manually created to facilitate references to the topics.

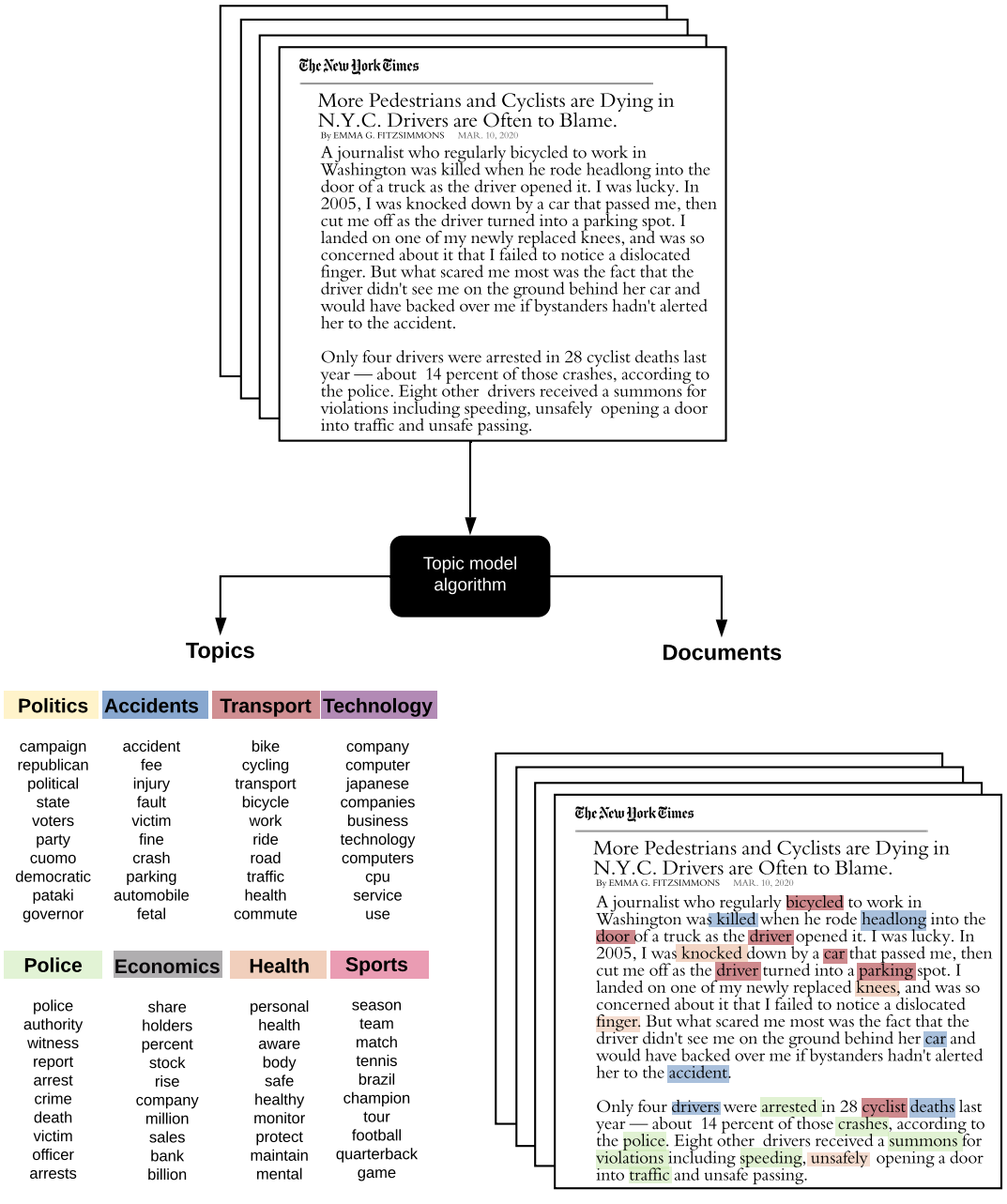



Figure 1.1: Topic model examines a collection of documents (e.g., news articles) and produces two outputs, topics and document classification, in the set of latent topics.

In research on machine learning, topic modelling was developed to automatically code the content of a collection of texts into a set of substantively meaningful topics. During this process, human involvement is not required, so prior human annotation, labelling, or hand-coding is not necessary to infer a model (Mohr and Bogdanov, 2013).

Classifying documents in this way is useful in many areas. For instance, the common method used to search in general is via keywords or phrases. This approach is useful when users know what they want to find. However, in cases where a large collection of documents is handed to a user who has no prior knowledge of their contents, it is useful to have a system (i.e., topic modelling) for the higher classification of these documents into topics. Topic models have been widely used in NLP, and they have been utilised in a range of tasks, including query expansion (Yi and Allan, 2009), sentiment analysis (He et al., 2011), document summarisation (Nagwani, 2015), and information retrieval (Wei and Croft, 2006). Furthermore, topic models have been used in various fields beyond text collection organisation. Two applications of topic models are described below:

- **Medical data application** Healthcare data can accumulate substantially, and they can include diagnoses, prescribed medications, and patient demographics. Topic models are used to discover the latent health status groups among patients and their corresponding characteristics. The discovered latent groups are used to develop a better clinical decision support system that provides the following capabilities (Lu et al., 2016):
  - Evaluates the likelihood of data in records, which allows the identification of outliers and therefore improves data quality.
  - Predicts diagnoses and medications based on a partial medical record.
  - Identifies pairings between diagnoses and medications.

  Automatic methods of data analysis provide opportunities for researchers and practitioners to create convenient and promising analytical tools that improve

the quality, safety, and efficiency of healthcare services (Lu et al., 2016).

- **Scientific data application** Expert-finding tasks, such as the peer review process, can benefit from topic modelling. For example, conference chairs use topic modelling to match expert reviewers to submitted papers. In such tasks, several constraints are considered in making the final recommendation, such as the reviewer's expertise, conflicts of interest, the number of reviewers per paper, and the number of papers per reviewer. Because of these aspects, the process can be intensive and time-consuming, so finding a means of accomplishing it automatically would relieve the burden on the conference chair. Topic models are used to discover the latent expertise of a person. Moreover, learned latent topics are used to match reviewers to suitable research papers (Mimno and Mccallum, 2007).

As shown in Figure 1.1, topics are usually represented using a set of words that are not ideal, as they tend to rely on the user's interpretation and knowledge. Alternative representations have been proposed, such as images or assigning topics with textual labels, as shown in Figure 1.1. In addition, it is important to design a useful representation of the entire document collection and its patterns. To represent topics well, it is useful to know how the mind processes information, learns, generalises, thinks, reasons, and makes decisions. Tufte and Schmieg (1985) stated that graphical representations of data could be more precise and reveal more than quantitative numbers. Moreover, representations should focus the user's attention on the quantitative data and not the design. To gain the maximum usage of visual representations, the principles of graphical excellence should be followed. To consider a representation to be graphically excellent, it should present interesting data clearly, precisely, and efficiently to the user. In other words, as described by (Tufte and Schmieg, 1985):

> *Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.*

Evaluating the effectiveness of a topic representation is considered a challenge. One way to address this challenge is through the application of qualitative methods, in which evaluating the representation involves humans by crowdsourcing the representations and gathering ratings. Another way is to estimate the quality of the representation according to its effectiveness in aiding the user in performing a task.

## 1.2 Thesis Aim and Scope

The previous section briefly introduced topic models and discussed the wide range of their applications. In the context of using topic models to summarise document collections and present the output to users, it is essential to address the final representation of the topic model's output. Therefore, the main aim of this thesis is to *improve the interpretability of automatically generated topics.*

Topic interpretability is an essential area of research because of its application in enhancing exploratory search interfaces (Aletras et al., 2014, Chaney and Blei, 2012, Smith et al., 2017) and developing interpretable machine-learning models (Paul, 2016). Topic interpretability is especially important when users interact directly with the topics, such as in exploratory search interfaces, which differs from the case where topics are integrated in another task (e.g., query expansion and word sense disambiguation). To achieve the aim of this thesis, several challenges are addressed:

- Topics are probability distribution over the entire vocabulary of the document collection; and therefore, only the top n words in the probability distribution are selected to represent the topic. Identifying the terms that are the most useful to appear within the top n is essential to produce interpretable topics for the user. To demonstrate:

  {*even, plan, health, coverage, group, insurance, house, bill, government, reform*}

  In this topic, commonly used words and less informative words (e.g., *even* and

*group*) appear before words such as *health* and *insurance*, which distracts the user from important information and reduces interpretability of the topic.

- Another challenge is deciding the ideal cardinal number. Setting the value of $n$ to a small number risks missing useful words that are below this cut-off. However, setting $n$ as a large number affects the topics' interpretability, which requires longer reading time, in addition to taking up more space in the visualisation tools. The same topic presented earlier is shown below after informative words are brought forward. It is represented by 5, 10, and 15 words.

  top 5: {*health, care, coverage, clinton, proposal*}

  top 10: {*health, care, coverage, clinton, proposal, insurance, medicare, bill, government, reform*}

  top 15: {*health, care, coverage, clinton, proposal, insurance, medicare, bill, government, reform, congress, house, president, plan, bills*}

  In the first option of five words, the topic is compact; however, compared with the representation with 10 words, critical words, such as *medicare* and *reform*, are omitted. Therefore, the topic subject is not reflected properly. Ten words per topic seems to achieve a good balance between keeping the topic compact and delivering the required information.

- Even after improving the topics by showing informative words first, it is not guaranteed that these words will be known by all users. Some words may require background knowledge, which reduces the interpretability of the topics. It has been shown that associating topics with labels reduces the cognitive load required to interpret them (Aletras et al., 2017). Labels can be used not only to facilitate the perception of topics but also to replace or represent topic terms. The topic shown above can be assigned or replaced by the label {*clinton health care plan*} or the label {*health care plan*}.

- Evaluating the improved topic representations poses a challenge, as there is no

standard approach to follow. This thesis adopts the evaluation of each proposed representation to accommodate the task and aim. In the first representation, which is improved topics with informative words shown first, two evaluation approaches are proposed: (1) automatic evaluation through an information retrieval (IR) task; and (2) a human study evaluation approach through crowdsourcing. In the second topic representation (i.e, textual labels), the labels were evaluated through the combination of human rating and automatic quality estimation.

This thesis is concerned with the outputs of topic models, and it explores ways to improve them. In particular, this thesis aims to improve identified topics using topic models as individual representations. Improvement in the overall representation of the topic model is not included in the scope of this thesis, such as improving an exploratory tool that incorporates many topics and represents documents organisation within the topics.

The experiments conducted in this thesis perform topic modelling using Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and are applied to English corpora. However, the proposed representation and metrics could be extended to other topic model implementations as well as to languages other than English.

## 1.3 Thesis Contributions

This section lists the contributions of this thesis by chapters:

- Chapter 3

  This chapter explores various ranking metrics for the topic's terms and introduces a novel ranking metric that is competitive with the state of the art. The chapter also introduces a novel approach to the task of topic evaluation. This evaluation approach evaluates the re-ranked topics automatically through an IR-based task.

- Chapter 4

  This chapter introduces an alternative evaluation approach to topic representation through a human study formulated as a crowdsourcing task. The study also explores the effects of different visual representations of topics on human performance.

- Chapter 5

  This chapter introduces a novel approach to automatically generating labels for topics. The approach follows a Sequence-to-Sequence (Seq2Seq) neural model architecture, and various configurations for the models are explored.

- Chapter 6

  This chapter includes an empirical study of various models in the task of topic labelling, which were introduced in the previous chapter. It also includes detailed descriptions of two synthetic datasets created to train the labelling models. The chapter also introduces the novel usage of contextual neural embeddings to evaluate the results of the proposed models.

## 1.4 Thesis Overview

Each chapter is summarised as follows:

- **Chapter 2: Background**

  This chapter provides background information about topic models in general and the methods used to evaluate them. The chapter pays particular attention to the visualisation of topic model outputs and includes approaches to improve their representations. It also describes the techniques used to estimate the interpretability of the representations. The chapter also provides background information about neural networks.

- **Chapter 3: Re-Ranking Automatically Generated Topics**

  This chapter introduces ways to create alternative topic representations. The proposed approach aims to improve the interpretability of a topic by re-ranking its terms. The chapter also introduces an automatic approach to measuring the quality of alternative topic representations.

- **Chapter 4: Human Evaluation of Topic Interpretability**

  This chapter introduces an alternative approach to further evaluating the topic representations presented in the previous chapter from a human perspective. The approach is formulated as a crowdsourcing task, in which the effects of interpretable topic representations on human performance are measured. The study also examines the effects of presenting the topic to the users in different visual forms.

- **Chapter 5: A Neural Approach to Automatically Labelling Topics**

  This chapter presents another topic representation in which topics are assigned labels. The labels are created through a novel approach using neural networks. The chapter defines the labelling task and various neural labelling models.

- **Chapter 6: Evaluation of Neural Approaches**

  Chapter 5 presents the implementation and experimental details of the label generation models introduced in the previous chapter. This chapter describes in detail the creation of two datasets that are used to train the models. The testing datasets are also described in detail. The chapter discusses automatic evaluation approaches to estimating the quality of generated labels. In addition, a qualitative analysis is conducted to determine the coherence and relevance of the generated labels.

- **Chapter 7: Conclusion and Future Work**

  The last chapter summarises this thesis and provides recommendations and directions for future work.

## 1.5 Previously Published Materials

Some parts of this thesis were published in peer-reviewed conference publications:

- Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. Re-ranking words to improve interpretability of automatically generated topics. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers (IWCS '19)*, pages 43–54, Gothenburg, Sweden, 2019.

- Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. Automatic generation of topic labels. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, pages 1965–1968, Virtual Event, China, 2020.

  In addition, most of the data and codes used in this thesis were released publicly and can be found at https://github.com/areejokaili/.

# 2

# BACKGROUND

Chapter 1 has briefly introduced topic models and has shown their ability to organise and summarise large collections of unstructured data. It has also established some of the challenges that accompany the application of topic models. Mainly, with focus on the topic models' interpretability and beneficial usage. Therefore, this chapter consists of formal and detailed descriptions of the topic modelling task and details of various approaches to perform topic modelling. It also includes the various and diverse forms of evaluating and assessing the quality of the topic model and its outputs. The chapter also covers the alternative representations developed for the topics and for the model as a whole.

This chapter begins by presenting an overview of topic models in section 2.1. The section also describes the mathematical formulation of various approaches to topic models and present different variations that stemmed from them. Next, several works on estimating and improving the topic model's quality and its outputs are presented in section 2.2. Section 2.3 shows various alternative representations for the topics as a stand-alone element and as a part of an overall system. Section 2.4, describes recent research aimed at improving the output of topic models after

training. Section 2.5 includes a brief description of neural networks since they are used in later chapters. Finally, a summary of the chapter appears in section 2.6.

## 2.1 Data Exploration

Learning the semantics (meanings) of large collections of documents is a sophisticated problem that has been widely investigated (Yang, 2012). Many data collections are highly structured, but others are unstructured (or semi-structured), such as data consisting of text. For example, a tremendous amount of data are created and circulated on microblog sites like Twitter. Extracting information from microblogs has become useful for discovering public opinion on different issues such as analysing messages (i.e., tweets) over a timespan to give insight into what happened during that time since people tend to tweet about matters in their life. Twitter has many tweets and an enormous amount of tweets posted every second, making manual inspection of such data an impossible task (Steinskog et al., 2017).

Topic models are a group of data mining techniques that explores a large collection of data and can identify groups of co-occurring words that summarises the data collection automatically. This section presents topic model techniques using the following notation and terminology (Blei et al., 2003):

- $w$ as a *word*, which is represented as a distinct token since topic models use bag-of-words notation.
- $\mathbf{w}$ as a *document*, a sequence of N words $\mathbf{w} = w_1, w_2, ..., w_N$.
- $\mathcal{D}$ as a *corpus*, a collection of documents $\mathcal{D} = \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M$.

In addition to the above terminologies and notations, additional terminologies are defined that are commonly used in topic models and therefore will be used in this thesis. It is important to explicitly define them as they can have other meanings in different domains.

- **Topic** In a topic model, a topic refers specifically to a subject derived from a

document collection. It usually consists of a group of words that are related semantically and that represent a common subject.

- **Terms** or **words** These are often used interchangeably in topic models.
- **Topic representation** The approach used to represent the topic to the user in a human-readable form. The topics as a raw output of the topic model are a low dimensional representation and therefore not easily understood by users.

### 2.1.1 Classic Topic Models

#### Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) was proposed to overcome the problems (or shortcomings) of document retrieval using queries that perform exact matches of words. LSA is a semantic approach that maps documents from high dimensionality count vectors to a lower dimensional space called latent semantic space using Singular Value Decomposition (SVD) which reflects the major associative patterns in the data. LSA output provides information beyond the lexical level as it provides semantic relation between documents vectors and words vectors. LSA takes a large matrix of document-word associations where rows correspond to documents and columns correspond to words, then LSA produces "semantic" space that places terms and documents close to each other if they are related (i.e., if they belong to the same concept). These new positions in the semantic space provide a new index for the information retrieval process. For example, query terms are located in this space and their concepts are identified, then documents that are nearby in the semantic space (i.e., share the same concepts with the terms) are returned as a result of the query.

The data collection $\mathcal{D}$ is represented as a sparse documents/words co-occurrence matrix $A$. $A$ of size $M \times V$. The SVD theorem states that matrix $A$ can be

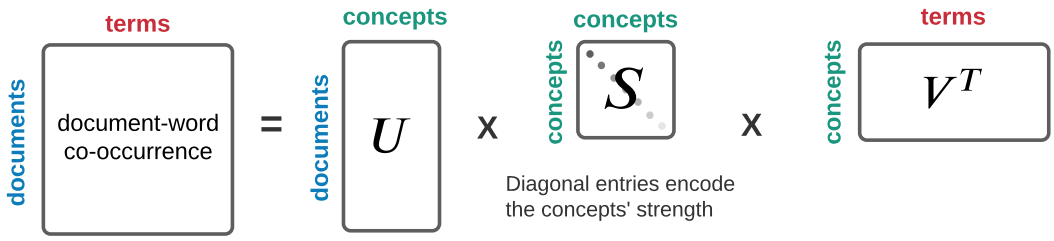



Figure 2.1: In LSA, the document-word co-occurrence matrix is decomposed into three matrices using SVD. Figure adapted from (Steyvers and Griffiths, 2007).

decomposed into three matrices as follows:

$$A = U \cdot S \cdot V^T, \tag{2.1}$$

where $U$ columns are the eigenvectors of $A\dot{A}^T$, and $S$ is a matrix with non-zero entries only on the diagonal and these values are the square root of the eigenvalues of $A\dot{A}^T$ called singular values. Singular values in $S$ encode the strength of the concepts and are sorted in a descending order which is useful for dimensionality reduction. The original matrix $A$ can be formed again by multiplying the three matrices $U$, $S$, and $V^T$. Figure 2.1 shows the decomposition of matrix $A$ into three matrices $U$, $S$, and $V^T$ using SVD.

Dimensionality reduction can be performed to represent the document collection with fewer number of concepts by keeping the $K$ singular values with the largest scores. In another words, LSA approximates $A$ by truncating K entries from the decomposition matrix $S$ giving $A'$. The approximation aims to minimize $||A - A'||_F$, where $||...||$ is the **Frobenius-norm** defined as $||D|| := sqrt(\sum_{ij} D_{ij}^2)$. Therefore, $A'$ with the $K$ largest singular values is given by:

$$A' = U_K \cdot S_K \cdot V_K^T, \tag{2.2}$$

where $U_k$ is a matrix of size $M \times K$ with rows as document vectors, $S_K$ is a diagonal matrix of the eigenvalues of $A$, and $V_K^T$ is a matrix of size $K \times V$ where

each columns represent a word vector. LSA produces this new representation for documents where each document in $U_K$ is represented as a linear combination of the concepts (i.e., topics). Concepts vectors $C_K$ are the product of the word vectors $V_K^T$ and the diagonal matrix $S_K$ as follows:

$$C_K = V_K^T \cdot S_K \tag{2.3}$$

The new low-dimension vector representation of concepts can be used to identify similar documents, such that documents with similar concepts tend to be close in the latent semantic space. Also, similar documents are identified even when they do not share terms with each other as long they share terms with another documents.

Although LSA has been shown to identify word synonyms, it has been less successful in identifying polysemy and yields concepts that are either incoherent or hard to interpret (Landauer and Dumais, 1997).

### 2.1.2 Probabilistic Topic Modelling

Probabilistic topic models discover the hidden semantic structure of a document collection based on a hierarchical Bayesian analysis of the original unstructured texts. They uncover patterns between words and documents to identify the latent structure of the text collection (Blei and Lafferty, 2009a).

Before describing the process of analysing the document collection using a probabilistic topic model, there are additional notations that are used in this section. Table 2.1 shows the notations and their description.

For example, from Figure 2.2 consider the two documents ($\mathbf{w}_1$ and $\mathbf{w}_2$) from a collection of documents $\mathcal{D}$ with $V$ possible vocabulary words. Passing the documents to a topic model will result in the generation of two distributions: document-topic ($\theta$) and topic-word ($\phi$) distributions. From the first distribution ($\theta$), we can identify that the first document ($\mathbf{w}_1$) has around 18% probability under topic 4 and from

Table 2.1: Topic model notations used in section 2.1.2.

| Notation | Description |
|---|---|
| $\theta$ | the multinomial distribution of documents over topics, therefore $\theta^m$ is the distribution of topics for the m-th document. |
| $\phi$ | the multinomial distribution of topics over words, therefore $\phi_t$ is the distribution over the words in the vocabulary $V$ for topic $t$. |
| $\alpha$ | the Dirichlet prior for topics concentration in documents. |
| $\beta$ | the Dirichlet prior for words concentration in topics. |
| $z$ | the topic assignments for words in a document $\mathbf{w}$, therefore $z_{\mathbf{w},n}$ is the topic assignment for the $n$-th word in document $\mathbf{w}$. |

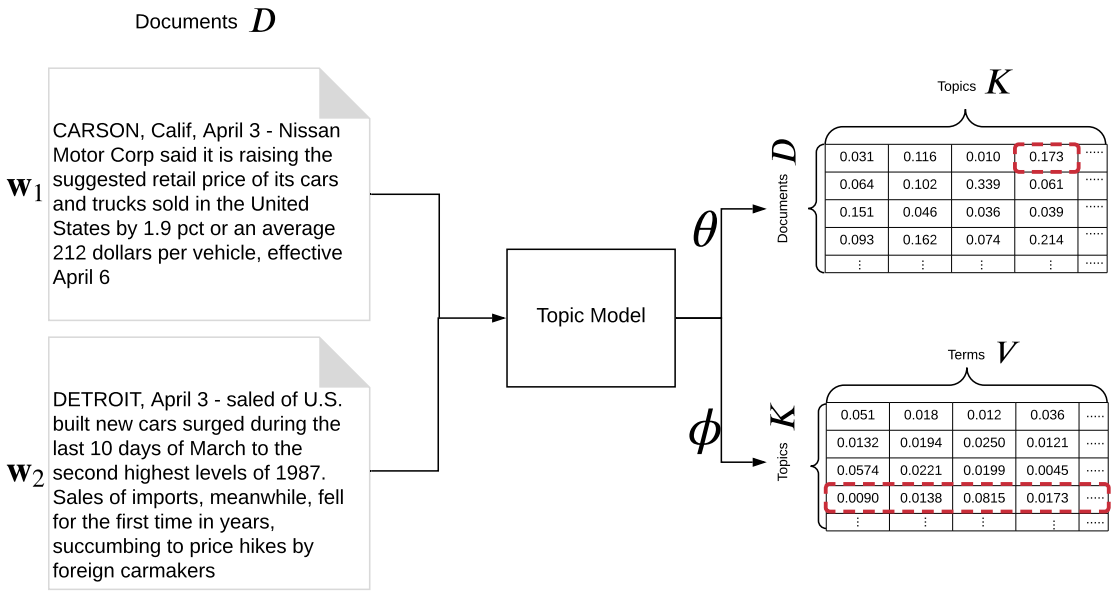



Figure 2.2: Example inputs to a topic model and the generated outputs.

the second distribution ($\phi$) we can identify the highest probability terms under topic 4. So, given the two documents without any additional annotation we can get an automatic summary of the document collection including the topics discussed in the document collection (left output in Figure 1.1 from Chapter 1) and topic proportion in each document (right output in Figure 1.1 from Chapter 1). Figure 2.3 shows the matrix decomposition of the probabilistic topic model.

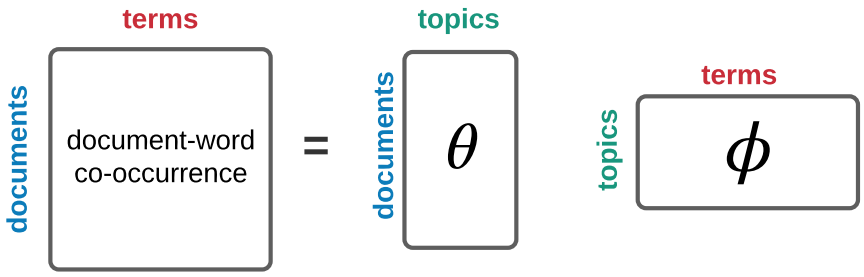



Figure 2.3: Topic model transforms the document-word co-occurrence matrix into two matrices $\theta$ and $\phi$. Figure adapted from (Steyvers and Griffiths, 2007).

The next two sections describe the common techniques for probabilistic topic models, namely Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA).

### Probabilistic Latent Semantic Analysis (PLSA)

PLSA (Hofmann, 1999) is a probabilistic approach for analysing a document collection that was derived from LSA. LSA performs linear transformation by aligning the documents with the co-occurrence table. Then, SVD is performed to transform documents from high-dimensional count vectors to a low dimensionality vectors. LSA has been used in different problems and shown to be a useful data analysis tool (Landauer, 2002). On the other hand, LSA is not a generative approach and the alignment has to be recomputed every time additional documents are added to the collection. It also optimises ad hoc variable and has difficulty in identifying polysemous words (Hofmann, 2001).

In contrast, PLSA is a statistical technique based on a mixture decomposition derived from the aspect model (Hofmann, 1999). It transforms documents to a lower vector representations and assumes that each document is a mixture of multiple topics. The main idea is to find a probabilistic model with the hidden topics that can generate the original observed document collection. Each document is created by first picking a topic from the set of topics then generating document words based on the topic's multinomial probability distribution. Therefore, the result from PLSA is

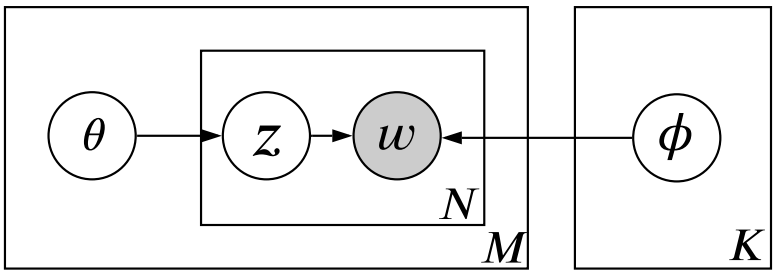



Figure 2.4: PLSA graphical model

that every document is represented as mixture of topics. The plate notation[1] shown in Figure 2.4, provides a quick overview of the variable association in PLSA, in which the topic latent variable $z$ is associated with the observed variable $w$. The generative process of PLSA can be defined in the following way:

1. Select a document $\mathbf{w}$ with probability $p(\mathbf{w})$:
   (a) For each word $w_n, n \in N$ in the document $\mathbf{w}$:
   Select a topic $z_n$ with probability $p(z_n|\mathbf{w})$,
   Generate a word with a probability $p(w_n|z_n)$.

Similarly, the generation process in a joint probability model of a document $\mathbf{w}$ and a word $w$ is as follow:

$$\begin{aligned} P(\mathbf{w}, w) &= P(\mathbf{w})P(w \mid \mathbf{w}) \\ &= P(\mathbf{w}) \sum_z P(w \mid z)P(z \mid \mathbf{w}) \\ &= \sum_z P(\mathbf{w})P(w \mid z)P(z \mid \mathbf{w}) \\ &= \sum_z P(\mathbf{w}, z)P(w \mid z) \\ &= \sum_z P(\mathbf{w} \mid z)P(z)P(w \mid z). \end{aligned} \tag{2.4}$$

The unknown parameters to be estimated are the document distribution over topic $P(z \mid \mathbf{w})$ and the probability distribution of topics over words $P(w \mid z)$. Parameters

[1] In a plate notation, nodes represent random variables, shaded nodes represent observed variables, directed lines show possible probabilistic dependence, rectangles show repetition, and the letter in the bottom right of the rectangle shows the repetition frequency.

estimation are usually performed using Expectation-Maximization (EM)(Dempster et al., 1977). EM performs estimation of parameters in two steps: the expectation step (i.e., E-step) and maximization step (i.e., M-stop). At first, EM initializes all unknown parameters randomly, then it performs the E-step, which computes the expected likelihood of the complete document collection given the current parameters and the observed documents. Then, the M-step is performed where the model re-estimates all the parameters by maximizing the likelihood of the collection. These steps continue to alternate until the optimization function (i.e., likelihood) converges.

Two constraints are maintained through the estimation of the latent variables. First, the sum of all words in one topic distribution should be 1, such that

$$\forall i \in [1, K], \sum_{j=1}^{V} P(w_j|z_i) = 1. \tag{2.5}$$

Second, all topics mixtures for one document $\mathbf{w}$ should sum to 1:

$$\forall d \in [1, M], \sum_{i=1}^{K} P(z_i|\mathbf{w}_d) = 1. \tag{2.6}$$

There are a few shortcomings of PLSA. PLSA models the document as a mixture of the topics, however, the model only learns the topic mixtures for the documents in the training set. Therefore, PLSA has the limitation of not being a generative model for new documents because it learns the possible topic proportions from the data it was trained on (Hofmann, 1999). Also, the number of parameters for the model grows linearly with the number of training documents. Linear overgrowing suggests the possibility of overfitting. Although, overfitting has been avoided by the generalisation of a maximum likelihood model fitting by Tempered Expectation Maximization (TEM)(Hofmann, 2001).

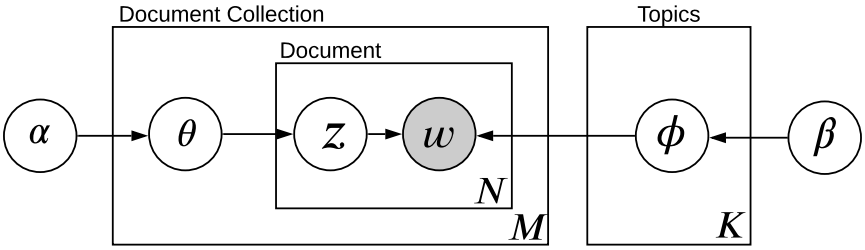



Figure 2.5: LDA graphical model

### Latent Dirichlet Allocation (LDA)

LDA (Blei et al., 2003) is a generative extension of PLSA that has been developed to address both of its shortcoming, namely its none generative capabilities and overfitting.

The graphical model representation for LDA in a plate representation is shown in Figure 2.5. The representations shows that LDA has three levels. First, the latent prior $\alpha$ and $\beta$ are corpus-based parameters and are sampled once in the process of generating a corpus. The variable $\phi$ is a topic-level variable and it is sampled $K$ times, while the variable $\theta$ is a document-level variable and it is sampled once per document. The variables $z$ and $w$ are word-level variables that are sampled once per word per document.

Given the number of topics, the distribution over words $\phi$, and distribution over topics $\theta$[2], the LDA's generative process is as follows:

1. For each topic $t_k, k \in K$:
   (a) Choose a distribution over words $\phi \sim \text{Dir}(\beta)$,
2. For each document $\mathbf{w}_m$, $m \in M$:
   (a) Choose $\theta \sim \text{Dir}(\alpha)$,
   (b) For each word $w_n$, $n \in N$:
      a. Choose a topic $z_n \sim \text{Multinomial}(\theta)$,
      b. Choose a word $w_n$ from $p(w_n|z_n, \beta)$.

[2]for now, lets assume that $\phi$ and $\theta$ are available, more on inferring these distributions shown shortly.

Similar to the above generative process, the model's joint probability of the whole corpus $\mathcal{D}$ is as follows:

$$P(\mathbf{w}, z, \theta, \phi \mid \alpha, \beta) = \prod_{d=1}^{M} P(\theta_d \mid \alpha) \prod_{k=1}^{K} P(\phi_k \mid \beta) \prod_{n=1}^{N} P(z_{dn} \mid \theta_d) P(w_{dn} \mid \phi z_{dn}) \tag{2.7}$$

The distributions $\phi$, $\theta$, and $z$ in the above equation are the variables that need to be estimated. Inferring the exact distributions is computationally challenging. There are various algorithms to approximate the inference of these distributions: expectation-maximisation (EM) (Hofmann, 1999); message passing (Zeng et al., 2013); variational inference (Blei et al., 2003); Online Variational Bayes (VB) (Hoffman et al., 2010); and Gibbs sampling (Griffiths and Steyvers, 2004). VB is the algorithm used to generate the topics in Chapter 3. VB is based on an online stochastic optimization approach that does not require a full pass over the entire corpus, and therefore can be applied to a large dataset. It can also accommodate cases where new data streams arrive constantly. VB has been demonstrated to find good topics comparable to those found by variational inference in less time (Hoffman et al., 2010).

There are a number of implementations of topic models such as *Gensim*[3] which uses online VB, *Stanford topic modelling toolbox (TMT)* (Ramage et al., 2009) which uses Gibbs sampling and *MALLET*[4] that also uses Gibbs sampling in addition to some open source implementations for topic modelling by David Blei's lab[5].

### 2.1.3 Variations of Topic Models

Going forward, this thesis will focus on LDA as the topic model of choice. LDA is the most popular topic modelling framework and has been used in diverse domains beyond text, such as music analysis (Hu and Saul, 2009), health record analysis (Lu

[3]http://radimrehurek.com/gensim/
[4]http://mallet.cs.umass.edu/
[5]https://github.com/Blei-Lab

et al., 2016), and image annotation (Liénou et al., 2010). Different variations to LDA have been proposed to overcome some of its drawbacks, to address specific tasks or to use labelled data.

**Address a Drawback** One of the limitations of LDA is that it lacks representation for correlation between documents. For example, a document that belongs to a "sports" topic is more likely to be similar to documents in "health" compared to documents in "business" topic. Correlated Topic Models (CTM) (Blei and Lafferty, 2006) form a topic relation graph via a covariance matrix of logistic normal distribution rather than Dirichlet to model the topic's correlations, in another word to identify correlation between documents that are related to similar topics. Similarly, the Pachinko Allocation Model (PAM) (Li and Mccallum, 2006) captures the correlation between topics and it models multiple topics correlations unlike CTM which only models pairwise correlations in the topics.

Topic models methods depend upon the presumption that meanings are relational (Saussure, 1959) which means topics are a constellation of words that tends to appear together whenever the content is about those topics. Each document is treated as a bag of words, and topic models capture co-occurrences regardless of these words' syntax, narrative or location. These assumptions make sense from the point of view of computational efficiency, but word order is essential in regards to meaning. For example, the word "*bank*" can refer to a financial institution or a blood repository in hospitals. Each of these meanings of the bank is called its **word sense**. Certainly, word location aids in the process of inferring to which topic or topics a word refers (Jurafsky and Martin, 2014, Wallach, 2006). Bigram Topic Model (BTM) (Wallach, 2006) is an extension to LDA with N-gram language representation instead of bag of words. Consequently, BTM only generates bi-gram words.

Griffiths et al. (2007) proposed an LDA Colocation Model (LDACOL), which extends the bi-gram model and can generate both uni-gram and bi-gram words. However, only the first term in a bi-gram word has a topic assignment but not the

second term. This limitation of LDACOL was resolved in the Topical N-gram Model (TNG) (Wang et al., 2007). Jameel and Lam (2013) proposed an Unsupervised Topic Segmentation (NTSeg) model that preserves the words' order and document structure. Hence, it generates n-gram words and multi-level topics, specifically, document segment topics and word topics. We have presented a number of studies where the word's location is integrated into topic models and has been shown to be useful. However, it often produces a complicated model that needs intensive, time-consuming computations (Yandex and Loukachevitch, 2015). Several attempts have been proposed to speed up the computations (Porteous et al., 2008, Zhu et al., 2013a,b).

Finally, the quality of the model depends on the hyper-parameter settings, namely vocabulary size ($V$), the number of topics ($K$), and Dirichlet parameters ($\beta$ and $\alpha$). $\beta$ and $\alpha$ control the sparsity of the topic distributions. $\beta$ represents topic-word proportion, and therefore high $\beta$ value results in topics that consist of most of the words in the vocabulary and the other way around with smaller $\beta$ value. The same goes for $\alpha$ a higher value generates documents that are composed of more topics than those generated with lower $\alpha$ value. The number of topics can affect the topic's interpretability. The choice of inferring a small numbers of topics usually results in very generic and broad topics whereas a large number of topics will result in topics that are uninterpretable and in which the models combine uncommon words (Sbalchiero and Eder, 2020, Stevens et al., 2012, Steyvers and Griffiths, 2007). Choosing the right values for topic model parameters influences the performance of LDA models. In some cases, it is difficult to know the data structure (e.g., when processing image data). Therefore, non-parametric Bayesian statistics have been used to define the models and automatically find the appropriate number of topics (Griffiths et al., 2005, Teh et al., 2006, Zhang et al., 2013).

**Address a Specific Task** A number of topic models have been developed to address various tasks other than data exploration, including social network

analysis (SNA) (Cha and Cho, 2012) where follow relationships between users are used to identify groups based in the user interests. Author-Recipient-Topic (ART) (Mccallum et al., 2005) a model that discovers the discussed topics in email messages conditioned on the sender-recipient relationships. People's roles are also identified by measuring the similarity between their distributions. Classification tasks have also been attempted using topic models. For example, a classification model that distinguished between phishing and legitimate emails has been trained on semantic features learned using topic models (Bergholz et al., 2008).

**Use Labelled Data** All the presented topic models so far are unsupervised probabilistic topic models that do not require any manually annotated datasets. They come with some drawbacks in certain applications such as not considering the class label of the document during inference in text classification. Also, a lot of datasets come paired with response variables such as text documents paired with a category, user reviews paired with rating number and web pages paired with the number of diggs[6]. Supervised topic models are topic models of documents and responses, conditioned to find topics predictive of the response. Supervision of topic models can be done in two approaches, namely, downstream supervised topic model (DSTM) and upstream supervised topic model (USTM) (Zhang et al., 2013) .

Supervised Latent Dirichlet Allocation (sLDA) was the first attempt to supervise topic models (Mcauliffe and Blei, 2008). The motivation behind sLDA is to solve prediction problems such as predicting a movie rating from reviews or predicting an essay's grade. A downstream supervised topic model is trained by maximising the joint likelihood of the content data and the responses (Zhu et al., 2012). The model is built upon a corpus of documents paired with responses from which the latent topics predictive of the responses are inferred. When a new unlabelled document is given, the response would be predicted based on the document's latent topic. SLDA predictive ability outperformed regression and unsupervised LDA. Another supervised

[6]diggs indicates the popularity of a web page and consequently the web page gets featured in the front page of the news aggregation platform `Digg.com`

topic model for classification is a discriminative variation on LDA called DiscLDA. DiscLDA is trained by maximising the *conditional* likelihood of the responses given the contents (Lacoste-Julien et al., 2008).

Apart from employing two-stage heuristics such as DSTM and USTM, Zhu et al. (2012) proposed *maximum entropy discrimination latent Dirichlet allocation* (MedLDA) which integrates the approach behind maximum margin prediction model (e.g., SVMs) with LDA. MedLDA adopts the discriminative max-margin principle into the framework of supervised topic models to enhance the performance of classification and to make more efficient use of the side information (i.e., ratings, labels associated with documents, or images). MedLDA reported achieving state of the art performance in latent topic discovery and prediction.

Jameel et al. (2015) presented a supervised topic model that was built upon their previous unsupervised model (Jameel and Lam, 2013). The model preserves word order and includes useful side-information such as class labels for text documents.

Another variation is for data that contains connected observations. For example, a follow-graph connecting social networks accounts and hyperlinked networks of web pages. Research has focused more on finding patterns and communities in this kind of data. Relational topic models (Chang and Blei, 2009) are supervised topic models that can build a model of hidden content and structure in standard and network datasets. Supervised topic models have been successfully applied to different domains such as image classification (Wang et al., 2009, Zhang et al., 2013). Based on sLDA, multi-class supervised latent Dirichlet allocation (mcLDA) (Wang et al., 2009) was proposed as a multiclass extension to sLDA that discovers the patterns in images and predicts their class and annotations.

## 2.2 Evaluation of Topic Models

In supervised tasks such as classification and regression, the predicted labels are compared to expected labels to evaluate the quality of the classifier. Whereas in unsupervised tasks such as topic modelling, evaluation can be challenging since we do not have in advance the anticipated topics and there are multiple possible candidate topics. In topic models there is a trade off between a model that represents that data and models that generate interpretable topics. Therefore, evaluation should be based on the intended usage of the model. For example, developing a model that extracts topical features from the data will have a different objective to a model that summarises a data collection in order to present the results to an end user. Therefore, there are no explicit *quantitative* methods that fit all intended applications of topic models. The evaluation directions are categorised into (1) measuring model fit, (2) measuring topics quality.

### 2.2.1 Measure Model Fit

**Held-out Likelihood (Intrinsic Metric)** Topic models are unsupervised methods that process a large volume of unstructured data (i.e., data without prior annotations) and generate a lower dimension representation of observed data. Therefore, one can measure how well the learned model represents that data through measuring the predictive likelihood of held-out documents (i.e., perplexity). Perplexity is usually used in language modelling, and it estimates the probability of words in held-out documents that are not used in training (Wallach et al., 2009). Perplexity is computed as the exponent of likelihood of unseen data ($\mathcal{D}_{test} = \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M$) as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{D}_{test}) &= \sum_{d=1}^{M} \log p\left(\mathbf{w}_d \mid \phi, \alpha\right), \\ \text{Perplexity}(\mathcal{D}_{test}) &= \exp\left(-\frac{\mathcal{L}(\mathcal{D}_{test})}{\sum_{d=1}^{M} N_d}\right), \end{aligned} \tag{2.8}$$

where $N_d$ is the number of words in document $\mathbf{w}_d$. Perplexity is a decreasing function of $\mathcal{L}$ and the lower the perplexity the better. A number of previous work have used perplexity as the evaluation metric of choice, including (Blei and Lafferty, 2006, Dieng et al., 2020).

**Secondary Task (Extrinsic Metric)** Model fit was also measured by using the topic model's outputs within secondary tasks, such as document classification (Lu et al., 2011, Xie and Xing, 2013), sentiment summarisation (Titov and Mcdonald, 2008), and information retrieval (Wei and Croft, 2006). However, these evaluation tasks formulate the metric based on predicting the documents' ground truth (e.g., classes in document classification) which is subjective. Therefore, the next section presents evaluation methods for assessing the quality of the topics and their interpretability.

### 2.2.2 Measure Topic Quality

Previous evaluation methods are useful for evaluating the predictive model but they do not capture the interpretability of the topics for the user. It has been shown that topic models with low perplexity (e.g., CTM) may infer less meaningful topics for the users than other models with high perplexity (Chang et al., 2009). Therefore, other approaches were introduced to capture the model's topics quality.

#### Topic Coherence

A topic is considered coherent when its terms have high semantic similarity and together indicate a sole subject. Consider the topic {*student, college, summer, program, degree, training, medical, career, students, dean*}. All terms are from the same subject and one can easily infer that they are related to graduate education. On the contrary, the topic {*theatre, dates, user, show, profit, firm, digital, friend, choice, ticket*} consists of terms from different domains and it is hard to interpret, therefore less useful to the user. An additional selection of coherent and incoherent

topics is shown in Figure 2.6, where the terms in the coherent topics together refer to a subject. Where the topics labelled as incoherent, there is less association between the terms and therefore would be considered less useful in indicating the topic's subject.

| Coherent | | Incoherent | |
|---|---|---|---|
| space | health | dog | king |
| earth | disease | moment | bond |
| moon | aids | hand | berry |
| science | virus | face | bill |
| scientist | vaccine | love | ray |
| light | infection | self | rate |
| nasa | hiv | eye | james |
| mission | cases | turn | treas |
| planet | infected | young | byrd |
| mars | asthma | character | key |

Figure 2.6: A selection of coherent and incoherent topics (Newman et al., 2010).

**Automatic Coherence Metrics** The first attempt to automate the evaluation of topic coherence was to identify topics as insignificant if their probability distribution is distributed equally over all words (AlSumait et al., 2009). Newman et al. (2010) presented another method that captures the model's coherence by looking at the top 10 words in the topic and measure their relatedness using pointwise mutual information (PMI). Mimno et al. (2011) replaced PMI with log conditional probability to compute topic coherence. Concepts from hierarchical ontologies, such as WordNet, were used to capture the conceptual relevance of a topic (Musat et al., 2011). However, WordNet has been shown not to be useful in evaluating the topic's coherence (Newman et al., 2010). Aletras and Stevenson (2013b) proposed computing topic coherence by constructing a distributional semantic space which locates semantically similar words near each other. Wikipedia articles were used as reference data to extract features and compute word frequencies. Coherence was computed as the pair-wise cosine similarity between the topic's terms vectors.

Usually, the topic model is trained with optimisation on perplexity, and the calculation of topic coherence is performed after training the topic model. However, optimising the training of the topic model on perplexity has been shown to produce sub-optimal topics (Chang et al., 2009). Therefore, several recent works have proposed incorporating coherence in the topic model training and optimising coherence rather than perplexity (Ding et al., 2018, Gui et al., 2019). Ding et al. (2018) proposed word embedding based topic coherence (WETC), this coherence metric leverages pre-trained word embeddings to compute coherence since word embedding carry semantic information about words that is similar to PMI. WETC approach is highly efficient since it does not involve extracting words' co-occurrence frequency from a large corpus and therefore applicable to be incorporated in training the topic model.

**Human-in-the-loop** Involving users in formal settings for evaluation is another way to measure the coherence of the topic from the point view of the user. Chang et al. (2009) proposed two tasks: (1) *word intrusion* and (2) *topic intrusion*. In a word intrusion task, users are shown topics as a list of words where an *intruder* word is added to the topic and the topic is considered coherent if users are able to find the intruder word successfully. On the other hand, topic intrusion follows the same idea of planting an intruder element to a set of elements but in this case it is a topic rather than a word. For each document, a number of related topics are shown to the user with an intruder topic and users are asked to identify the out of context topic. Figure 2.7 shows the word intrusion task (left) and topic intrusion task (right). Word intrusion measures the coherence of the topic's words while topic intrusion measures how well the documents are decomposed as a mixture of topics.

Even though involving humans in the judgement process is effective, it can be time consuming and expensive. Therefore, an attempt to automate word intrusion tasks was proposed by Lau et al. (2014). Their automated approach evaluated interpretability at a near human level of accuracy. Later, Lau and Baldwin (2016) suggested that topic coherence should be computed on several cardinality (e.g., using

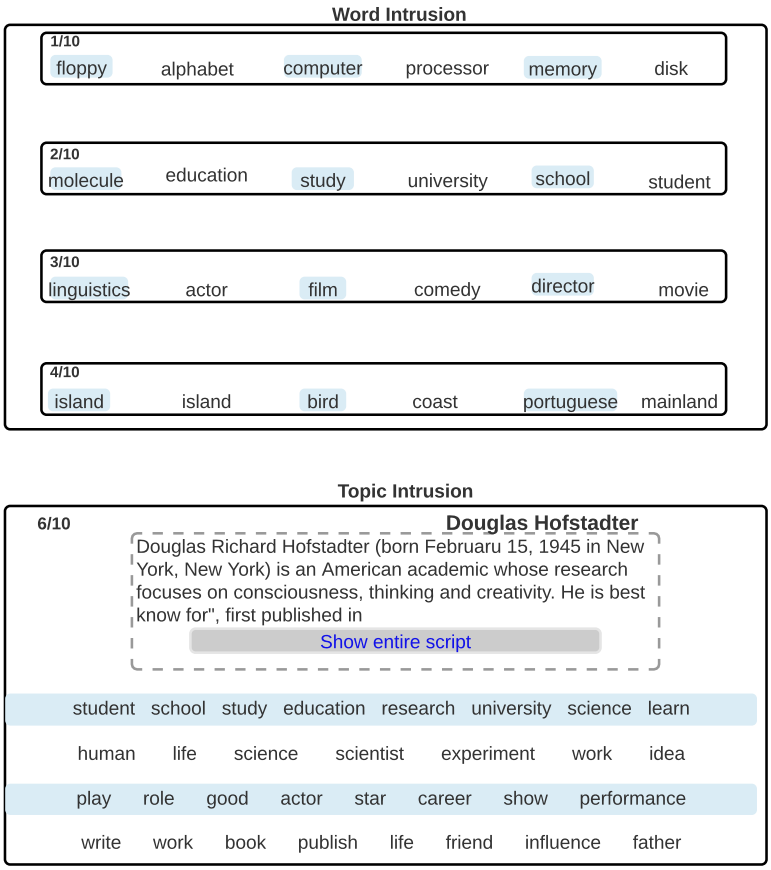



Figure 2.7: *Word intrusion* and *task intrusion* tasks to evaluate topic model quality by adding an intrusion word or topic and then asking the user to find the word/topic that does not belong with the others (Chang et al., 2009)

topics with top 10, top 20, ..., top $n$ words) settings and then aggregated over them.

Most of the work to evaluate topic models tends to focus on topic coherence and does not assess the topic's effectiveness in describing documents (i.e., document-level evaluation). Bhatia et al. (2017) disagree with accepting that topic-level evaluation as a reliable metric for the model's quality and proposed an analysis of the allocation of topics to documents to predict the quality of the model. They have shown that there is a discrepancy between topic-level (i.e., topic coherence) and document-level evaluations. Also, they presented the first automated approach to *topic intrusion* task, which is more suited for large-scale model evaluation than manual evaluation. They rank the topics for a document based on their likelihood of being an intruder topic using support vector regression (SVR)(Joachims, 2006) and the top topic is the

system's predicted intruder. This approach has shown a high correlation with manual evaluation. Later on, they also proposed an improved approach to the automatic intruder topic identification through ranking the topics for a given document using neural networks (Bhatia et al., 2018).

#### Topic Stability

Another study looked into measuring the quality of topics based on the volatility of change in the topic distribution throughout the training. Variability of the topic's word distribution during training can indicate the topic consistency and therefore quality. Xing and Paul (2018) proposed *topic stability*, a metric that measures the degree to which a topic's parameters change during training, and it showed higher correlation with human judgements than coherence metrics.

## 2.3 Visualisation of Topic Models

Probabilistic topic models provide ways to index, summarise and analyse large unlabelled documents by their hidden themes (topics). However, presenting the user with the raw output of the topic model does not promote the user's understanding of the document collection. Visual representations of topics are a common way to show the results of the topic model, which aids in the understanding of the data. The visualisation includes representing the individual topics themselves (i.e., the top $n$ terms) and representing the whole model (i.e., the topics and documents).

### 2.3.1 Visualisation of a Topic

The most common topic visual representation is a simple list of the top $n$ words of the topic, ranked based on their probability (Figure 2.8). Different variations of the list have been proposed: they can be represented vertically (Chaney and Blei, 2012, Eisenstein et al., 2012) or horizontally (Gardner et al., 2010, Smith et al., 2015) or

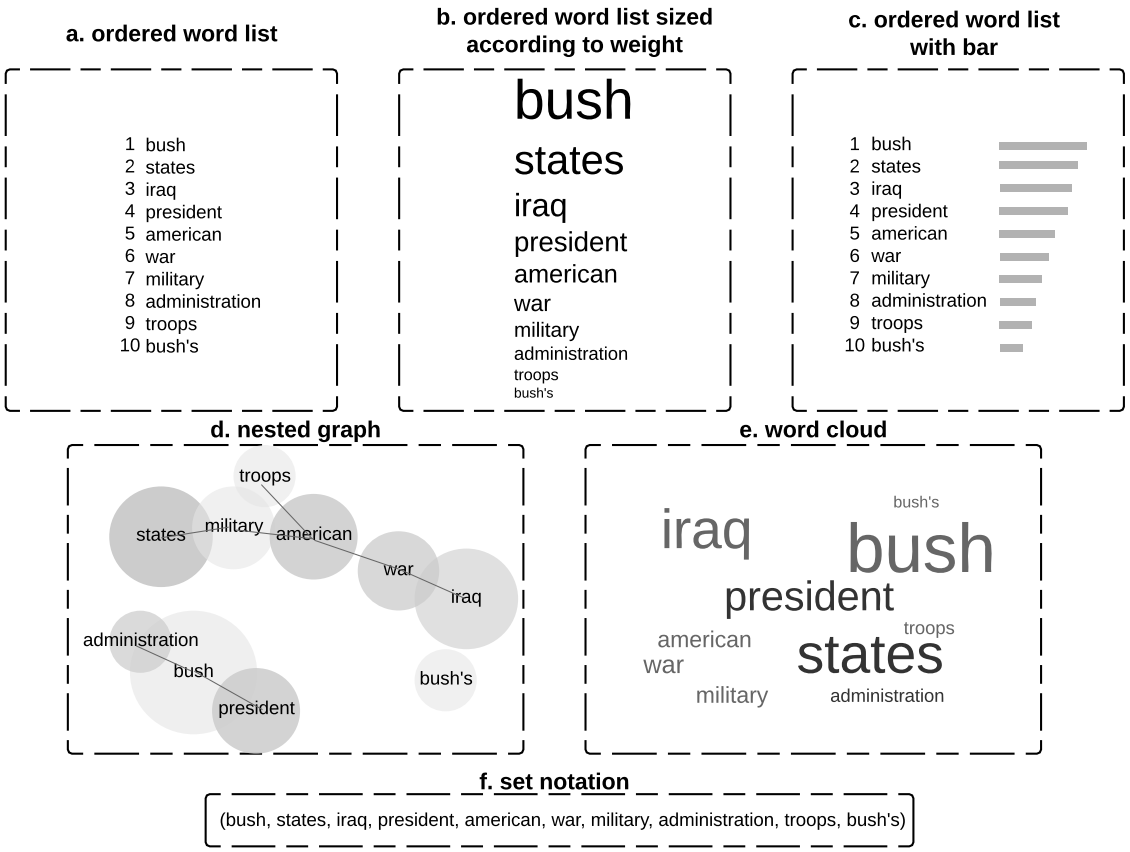



Figure 2.8: Common visual representations of topics.

using a set notation (Chaney and Blei, 2012) (Figure 2.8.f). The weighting assigned to each term can be included in the representation through sizing the term according to its probability for the topic (Nguyen et al., 2013)(Figure 2.8.b). Moreover, weight has also been represented by a bar graph next to a word instead of resizing the word (Figure 2.8.c). Using a list with the bar delivers two key pieces of information, representing the words' ordering according to probability and the weight associated with the words for a specific topic (Smith et al., 2015, 2017).

Information-dense representations have also been used to represent the topics, such as *word cloud* (Figure 2.8.e) and *network graph* (Figure 2.8.d). Word cloud (or *tag cloud*) is a graphical representation of text that is usually used to summarise keywords in text. Keywords are usually uni-grams and the importance or probability of each keyword is shown in the form of colour or size. There are many variations in

the layout of word clouds [7]. Words can be placed randomly but sized based on their probability (Rampage et al., 2010), the rank order of the word (Barth et al., 2014), or based on a combination of both (Smith et al., 2017).

A network graph represents each topic as a collection of circular nodes where each node represent a word. Links between words indicate high relevance based on document-level co-occurrences (Smith et al., 2014b). Another variation to this network graph is the one presented by Smith et al. (2017) where words are linked and located close to each other, if they appear frequently together in the document collection. The probability of the word is represented by the circle size around the word. The network graph is created using a force-directed graph layout algorithm that locates words closer to each other based on their co-occurrence (Fruchterman and Reingold, 1991).

Word cloud has visual appeal, however it is difficult to make comparisons between word clouds and they can be overwhelming and confusing. Word storms (Castella and Sutton, 2014) were proposed to address these drawbacks. Word storms is an approach to coordinating word clouds such that the same word appears in approximately the same position and colour across multiple word clouds, which should facilitate comparisons and hence increase the interpretability of topics in the document collection. Figure 2.9 shows an example of independent word clouds (top two) and coordinated word clouds (word storms)(bottom two).

So far, this section has surveyed various visual representations that have been shown to be useful in aiding the interpretation of topics by users. However, choosing a suitable representation (or set of representations) depends strongly on the application and its potential users.

[7] https://github.com/amueller/word_cloud

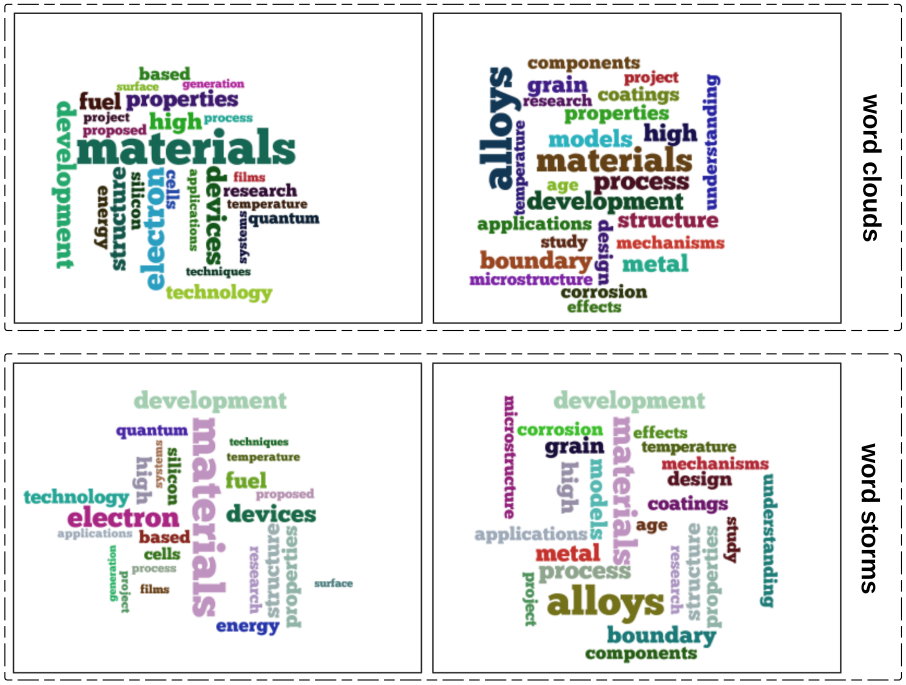



Figure 2.9: The top two word clouds are created independently from each other, while the bottom two are word storms that are coordinated and show the same terms in similar locations and colours (Castella and Sutton, 2014).

### 2.3.2 Visualisation of Documents Collection

All the representations shown so far in section 2.3.1 are concerned with finding effective and informative representations for the topic without taking into consideration the representation of the whole document collection. This section presents systems and tools to model, summarise, and visualise the entire corpus in a user-friendly interface. Some tools take it further and allow users to interact with the modelled data.

A number of exploratory systems for topic models have been developed where users can browse through documents, topics and terms (Chaney and Blei, 2012, Gardner et al., 2010, Snyder et al., 2013). These systems usually use simple visualisation to represent topics such as listing their top $n$ words. Figure 2.10 shows a topic model visualisation engine of Wikipedia articles where the topics are shown in a set notation of the top 3 terms. Clicking on a topic shows its top 10 terms, documents, and related topics.

Another visualising system compactly represents topic models to allow users to

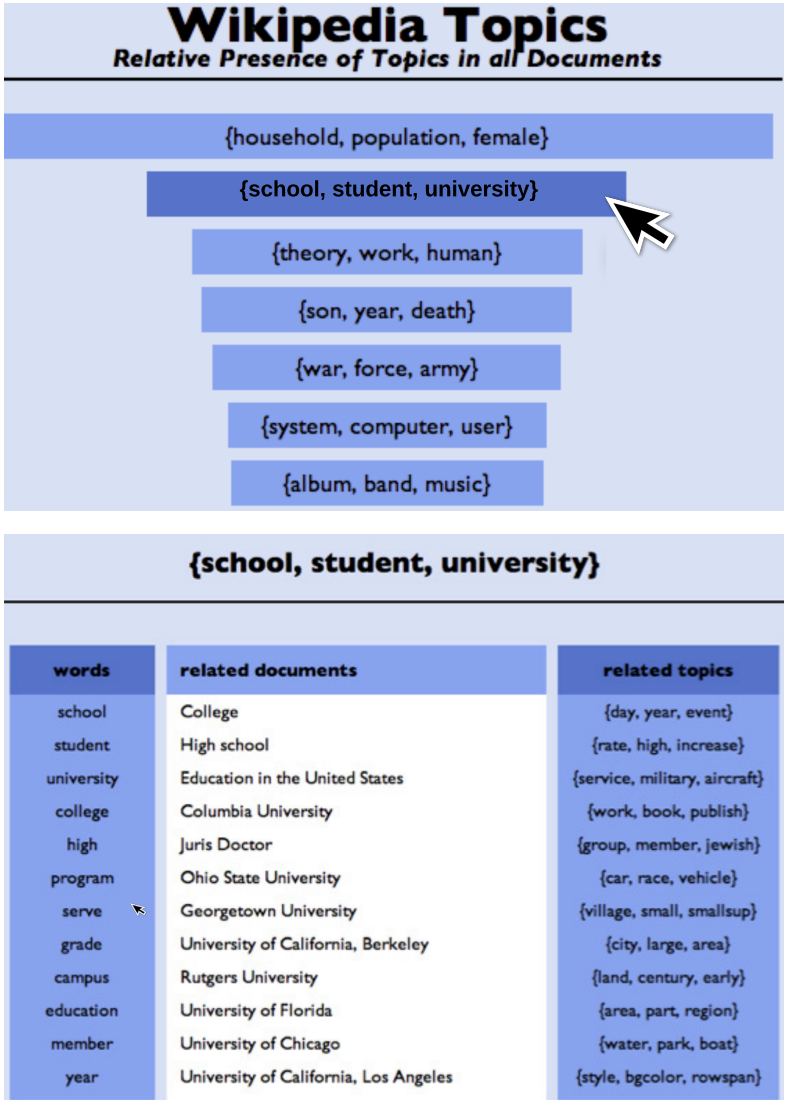



Figure 2.10: Chaney layout (Chaney and Blei, 2012).

easily understand the topics without the need to visualise documents too. Termite (Chuang et al., 2012) is a compact tool that visualises the topics and terms in a matrix layout, as shown in Figure 2.11. The tool uses a *saliency* metric to quantify how much information each term conveys about a topic. Saliency is computed as the product of the term's probability under the topic and the term's distinctiveness. Terms that occur under all topics are less informative, thus more highly discriminative terms are preferred. the same authors have also proposed *seriation*, an algorithm that identifies clustering patterns amongst terms and therefore, allows for comparison between topics. Terms are ordered either *alphabetically*, by *frequency*, or using *seriation*. Even

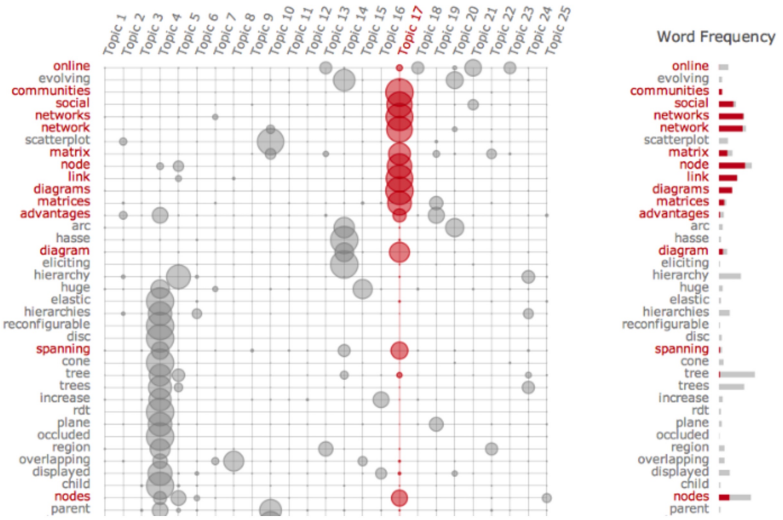



Figure 2.11: Termite layout (Chuang et al., 2012).

though Termite provides a compact representation and helps the user glance into the data, it only shows the top-ranked terms and does not show all the terms, making it useful for a global view of the whole topic model and not for a deep exploration.

Similarly, Sievert and Shirley (2014) proposed a web-based visualisation tool called LDAvis that provides a broad overview of the topics in the data. In addition to that, the tool also provides a way to show how the topics differ from each other while allowing the user to view the terms of which each topic is composed. They have also proposed a novel method for measuring the relevance of the term to a topic and it can be interactively adjusted which helps in inspecting the model. LDAvis is shown in Figure 2.12.

Smith et al. (2014b) have also proposed an interactive visualisation, Hiérarchie. The visualisation consists of a hierarchical representation of the data collection in the form of rings in a sunburst chart.

Engaging the user with the visualisation system makes it more useful as each user can customise and seek different information as needed. TopicViz (Eisenstein et al., 2012) is an interactive exploratory system that organises and summarises a large collection of research papers. The system supports traditional keyword search where the user query a word and related documents are returned (the left panel of

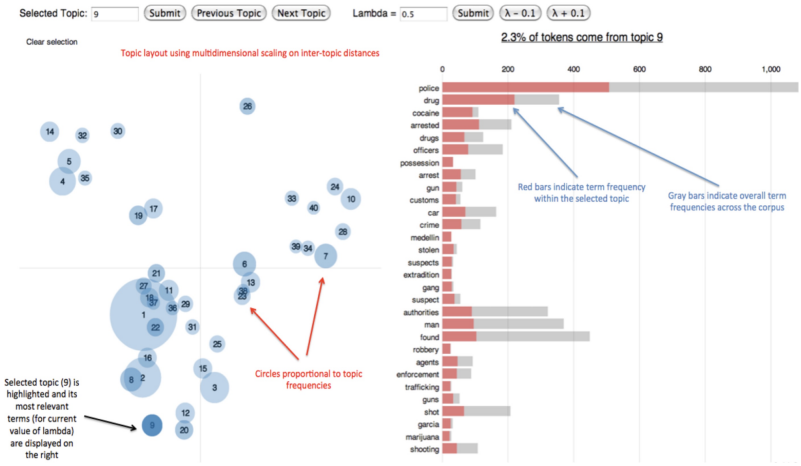



Figure 2.12: LDAvis tool layout Sievert and Shirley (2014)

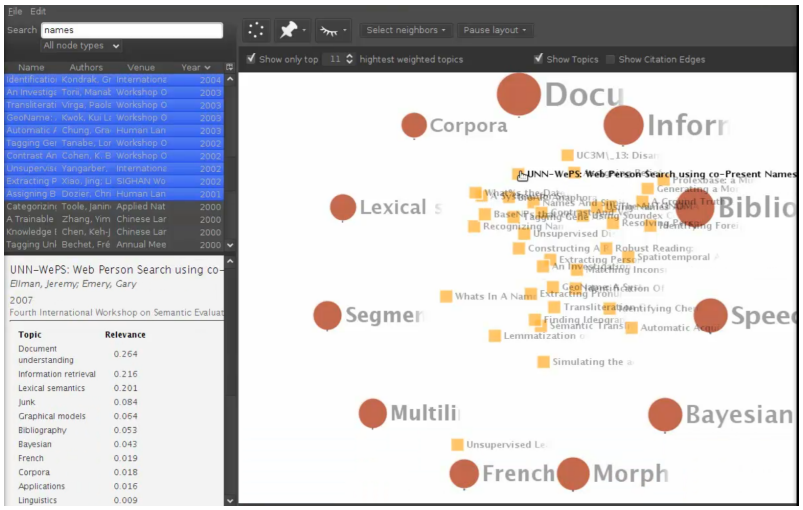



Figure 2.13: TopicViz exploratory tool interface (Eisenstein et al., 2012).

the system interface in Figure 2.13). The system also provides another visualisation for these returned documents where they are represented with their related topics in a directed graph. In the graph, each document is represented as a square; topics as circles and citations as edges (edges not shown in the figure).

Other studies aim to address the limitation of representing the relationship between topics. Blei and Lafferty (2007) proposed CTM (presented in section 2.1.3) an exploratory tool for a large collection of documents. CTM discovers the correlation between topics in the corpus by using a more flexible topic distribution, and it visualises the whole set of topics in the data in a graph form. In addition, the popularity of the topic is represented by the font size of its terms. Topics are

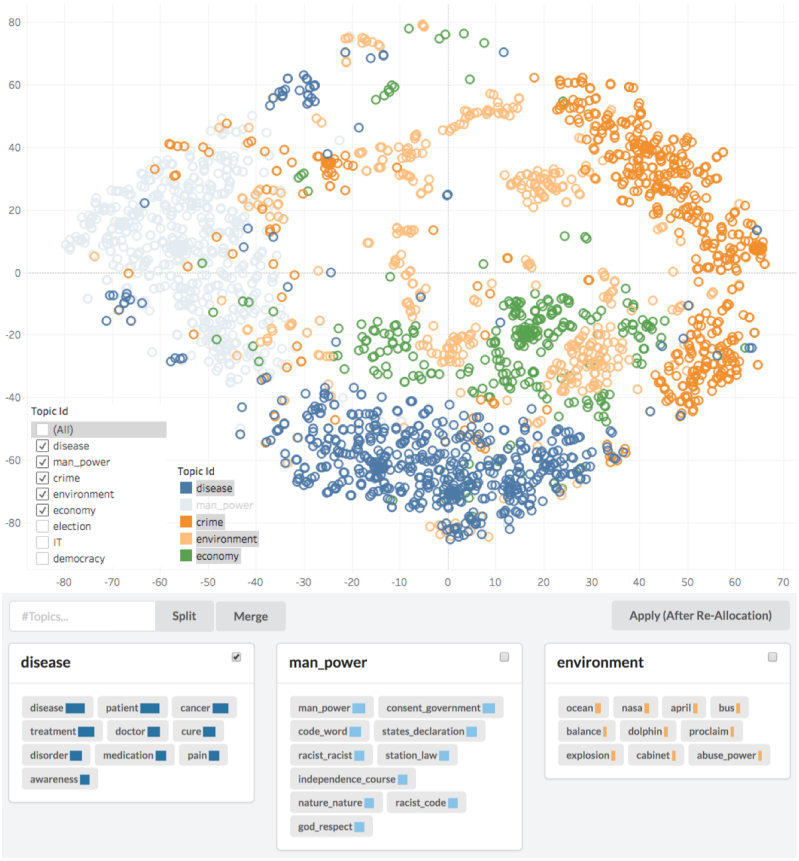



Figure 2.14: Topic model visualisation tool with interactive capabilities (Cai et al., 2018)

shown as nodes with the top 5 most probable terms and the correlation between topics is shown by the links between the topics nodes. Smith et al. (2014a) proposed a way to lower the cost of incorporating relationship information into the model, and proposed visualising relationships between topics and terms in an efficient manner. They choose a network graph representation for the individual topic (Figure 2.8.d). Topics that are more connected to each other are located at the center of the visualisation, whereas the least connected topics are located at the borders. The tool also supports interaction with the user where they can add/remove topic words, link words within a topic, and separate words between different topics. Another interactive tool that represents topic in the semantic space and which facilitates users' understanding of the relationships between topics is suggested by Cai et al. (2018). From Figure 2.14, the tool provides two functions: the upper panel shows how well

the topics are clustered according to the topic model space and the lower panel shows a visualisation of the terms within a topic that allows for topic manipulation including the addition/removing of terms and the merging/splitting of topics.

### 2.3.3 Evaluation of Topic Visualisation

Evaluating the effectiveness of a visualisation tool is a challenging task. Usually, the visualisation's usefulness is measured by conducting a user study, where the users' success in performing a task indicates that visualisation's benefit. Castella and Sutton (2014) evaluated the usefulness of word storms (shown in Figure 2.9) by asking the users to perform a simple task. Users were presented with six word clouds, either generated independently or in coordinated storms, and were asked to check the presence of a specific word or to identify the most similar/different word clouds. The user study affirmed the usefulness of word storms over independent word clouds.

A more sophisticated task such as information retrieval has been used to evaluate visualisations. Aletras et al. (2017) asked users to find documents presenting topics in one of the following representations: top $n$ words, textual phrases and images. The effectiveness of topic representations was measured based on the number of retrieved documents and the relevance of the retrieved documents. They found that the top $n$ words deliver more accurate information about a topic, enabling the retrieval of more relevant documents. Smith et al. (2017) asked the users to give labels to topics using four different topic visual representations. Then, to evaluate the representations, another set of users were asked to rate those manually generated labels and how well they described the documents.

## 2.4 Post-Processing of Topics

Improving the interpretability of topic models is an important area of research. A range of approaches have been developed including computing topic coherence (Ale-

tras and Stevenson, 2013b, Lau et al., 2014, Mimno et al., 2011, Newman et al., 2010), determining optimal topic cardinality (Lau and Baldwin, 2016), and corpus pre-processing (Schofield et al., 2017).

Moreover, post-processing the topics after generation is a simple approach to improve their interpretability. Post-processing has been incorporated in interactive tools where the user can remove/add terms in a topic (Nguyen et al., 2013, Smith et al., 2014b), split/merge topics (Cai et al., 2018), define topic labels (Nguyen et al., 2013, Smith et al., 2014a), change word order (Cai et al., 2018), and adjust terms' probabilities (Cai et al., 2018). Such tools are shown in more details in section 2.3.2.

However, improving the topics before presenting them to the user is essential and does not rely on the users' expected alterations. Automatic improvements to the topics include: removing stopwords (Schofield et al., 2017), assigning labels to topics that summarise the topic's subject, re-ranking topic terms to surface informative and distinctive terms as part of the topic's top $n$ terms.

The following sections present work conducted to improve topics post inference through re-ranking (section 2.4.1) and labelling (section 2.4.2).

### 2.4.1 Topic Terms Re-Ranking

Topics are multinomial distributions over a predefined vocabulary of words. The standard approach to representing topics has been to show the top $n$ words with the highest probability, e.g., (Blei and Lafferty, 2009a,b). However, these words may not be the most informative words facilitating the topic's subject. It is anticipated that some words relevant to a particular topic have not been assigned with a high probability due to data sparseness or low frequency in the training corpus (Chang et al., 2009, Lau et al., 2010). A range of approaches to re-ranking the topic's words have been proposed. For example, let us consider the following topic with its default top 10 terms, ⟨*data, visual, information, visualisations, technique, user, based, visualisation, large, paper*⟩. After re-ranking, terms such as *system* and *analysis*

could be ranked above less informative ones such as *based* and *paper* resulting in the following topic, ⟨*data, visual, information, visualisations, technique, user, visualisations, large, system, analysis*⟩.

Blei and Lafferty (2009a) proposed a re-ranking method inspired by the Term Frequency–Inverse Document Frequency (TF-IDF) word weighting which includes two types of information: firstly, the probability of a word given a topic of interest and, secondly, the same probability normalised by the average probability across all topics. The intuition behind this approach is that good words for representing a topic will be those which have both high probability for a given topic and low probability across all topics. Blei and Lafferty (2009a) did not describe any empirical evaluation of the effectiveness of their approach.

Other word re-ranking methods have also combined information about the overall probability of a word and its relative probability in one topic compared to others. Chuang et al. (2012) describe a word re-ranking method applied within a topic model visualisation system. Their approach combines information about the word's overall probability within the corpus and its distinctiveness for a particular topic which is computed as the Kullback Leibler (KL) divergence between the distribution of topics given the word and the distribution of topics. Bischof and Airoldi (2012) developed an approach for hierarchical topic models which balances information about the word frequency in a topic and the exclusivity of that word to that topic relative to a set of similar topics within the hierarchy. Similarly, Sievert and Shirley (2014) proposed a relevance metric to rank the terms that combines information about the term's overall probability in the corpus and the term's probability in the topic as a logarithmic weighted average (similar to the information used by (Chuang et al., 2012)).

Others have proposed approaches that only take into account the relative probability of each word in a topic compared to the others. Song et al. (2009) introduced a word ranking method based on normalising the probability of a word in a topic

with the sum of the probabilities of that word across all topics. They evaluated their method against two other methods, the topic model's default ranking and the approach proposed by Blei and Lafferty (2009a), and found that it performed better than either. A similar method was proposed by Taddy (2012) who used the ratio of the probability of a word given a topic and the word's probability across the entire document collection.

Xing and Paul (2018) proposed using information gathered while fitting the topic model. They made use of topic parameters from posterior samples generated during Gibbs sampling and re-weighted words based on their variability. Words with high uncertainty (i.e., their probabilities fluctuate relatively highly) are less likely to be representative of the topic than those with more stable probability estimates.

Topic re-ranking has also been explored within the context of measuring topic quality (Gollapalli and Li, 2018). A main claim of this work is that word importance should not only depend on its probability within a topic but also on its association with relevant neighbouring words in the corpus. This information is incorporated by constructing topic-specific graphs capturing neighbouring words in a corpus. The PageRank (Brin and Page, 1998) algorithm is used to assign word importance scores based on centrality and then re-rank words based on their importance. The top $n$ words with the highest PageRank values are used to compute the topic's quality.

A common characteristic of previous work on topic word re-ranking is that it has been carried out within the context of specific applications (e.g., topic visualisation) and approaches have been evaluated in terms of these applications, if at all. The fact that word re-ranking methods have been considered in previous studies demonstrates their importance. The lack of direct and systematic evaluation is addressed in this thesis. Chapter 3 compares several word ranking methods and evaluates them using an automatic information retrieval evaluation task, in addition to introducing a human study for evaluating the topics in chapter 4.

### 2.4.2 Topic Labelling

As previous studies have shown, topic terms can be noisy and difficult to interpret, therefore assigning textual labels to topics could help understanding the topic's subject easier. For example, a topic with the following words ⟨*school, student, university, college, teacher, class, education, learn, high, program*⟩ could be labelled with *Education*.

Early approaches to labelling relied on manual assignment of appropriate labels to topics (Mei and Zhai, 2005, 2006, Mei et al., 2006, Wang and Mccallum, 2006). Manual labelling is usually *subjective* and probably affected by the user's personal opinions in addition to the time and cost incurred during the gathering of the labels (Mei et al., 2007). Mei et al. (2007) proposed the first probabilistic automatic method to generate labels for topics. They formulated the labelling task as an optimisation problem which involves minimising KL divergence between word distribution in a topic and between word distribution in candidate labels. Then the candidate labels are ranked based on their relevance and the top $n$ are chosen to represent the topic. Following that, Magatti et al. (2009) introduced an algorithm to label topics based on the topics' similarity to categories from a taxonomy or a topic hierarchy tree.

Furthermore, Lau et al. (2010) proposed a novel method to select the best words to label topics instead of representing topics with all their top $n$ words. They used a number of features to help identify the best word for each topic including: conditional probabilities, PMI, WordNet hypernym relations, the word rank given by the topic model, and a distributional similarity score.

A number of studies have defined the labelling task as a search and rank approach where the topic terms are used to query a reference dataset (e.g., Wikipedia titles) and the returned candidate labels are ranked to find the most appropriate label. Lau et al. (2011) presented a method to create labels by querying the top $n$ words from Wikipedia automatically then ranking the candidate labels by lexical and association

features using a supervised method, SVR. While Hulpus et al. (2013) opted to use the structured data in DBpedia[8] to label topics. Their method starts by retrieving DBpedia concepts based on the topic words and then matching topics with the most relevant concepts using graph centrality.

Aletras and Stevenson (2014) proposed a similar approach to those of Lau et al. (2011). Instead of using a supervised model (i.e., SVR) to identify the most appropriate label among the candidate labels, they created a graph from the words returned from the query then used PageRank (Page et al., 1999) to identify candidate labels by weighting the words in the graph. Their unsupervised graph-based approach generated better labels than those generated by the approach of Lau et al. (2011).

Bhatia et al. (2016) also proposed a simpler and more efficient approach for automatic generation of topic labels compared to the state-of-the-art approach of Lau et al. (2011). Their method consists of two steps: first, they generate topic labels using English Wikipedia then they rank the labels based on a combined word and document embeddings created using word2vec (Mikolov et al., 2013b) and doc2vec (Le and Mikolov, 2014), respectively.

Beyond labelling topics with words or phrases, summarisation techniques have also been used to create labels for topics. Cano Basave et al. (2014) proposed the first such approach to label topics created from Twitter, whereas Wan and Wang (2016) extracted several summary sentences from documents related to topics.

Images have also been used as labels to topics instead of text phrases as they can be understood quickly and are independent of the user's language (Aletras and Stevenson, 2013a). Candidate images are retrieved by querying the topic's top $n$ words in a search engine. The most representative image is found by ranking the topic's candidate images based on their associated meta data and using a graph-based algorithm (PageRank (Page et al., 1999)). Furthermore, Aletras et al. (2017) proposed an improved approach to matching topics with representative images and

[8]http://dbpedia.org

created a more generic approach that estimates the degree of association between any topic-image pairing using a deep neural network. This approach has reported better performance compared to Aletras and Stevenson (2013a) in regards to speed and accuracy.

Most existing topic labelling approaches are performed in two steps: (1) Retrieval where candidate labels are identified (e.g., by querying the topic's top $n$ words), and (2) Ranking to identify the most related label among others in the candidate pool. Such extractive approaches can be limited by the coverage of the pool that is used to retrieve the candidates (e.g., Wikipedia article titles or topic terms) and the effectiveness of the relevance ranking method. Therefore, in Chapter 5 an alternative neural-based approach that does not suffer from this limitation is proposed. Labels that were generated from the neural-based models were evaluated against the topics themselves to assess their relatedness, in addition to comparing them to gold labels that were rated for appropriateness by humans.

## 2.5 Artificial Neural Networks

Artificial Neural Networks (ANNs) are networks with multiple inputs, hidden layers and outputs. The simplest form of ANN is a perceptron with one neuron (Mitchell, 1997). A neuron receives a set of real-valued inputs $x = (x_1, x_2, \ldots, x_n)$ and produces an output $y$ by performing a linear combination operation on the inputs:

$$z = wx + b \tag{2.9}$$

$$y = a = g(z) \tag{2.10}$$

where $w$ is the weight which the input contributes to the output. For example, $w_i$ is the weight contribution of $x_i$ to the output $y$. $b$ is a scalar referred to as the bias, and $g(.)$ is an activation function. Figure 2.15 also shows the neuron with the inputs, operations, and the outputs.

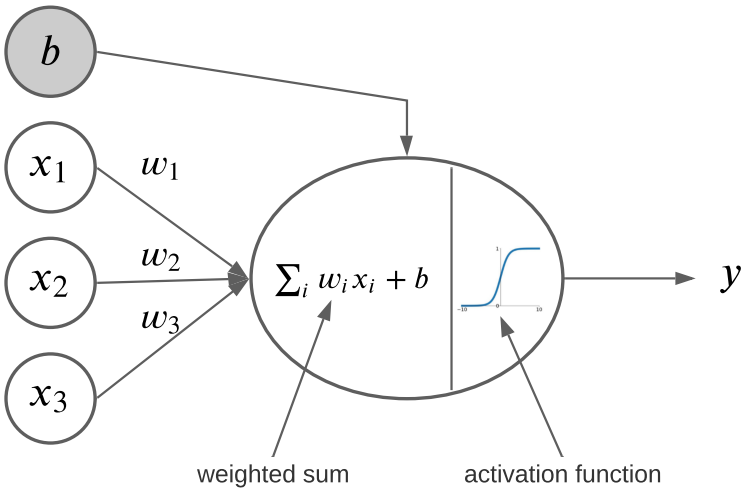



Figure 2.15: A visual representation of a single neuron with multiple inputs, and the associated weights. The output $y$ is computed as the sum of a combination of the inputs after passing through an activation function to normalise the result. In this figure the activation function used is sigmoid (figure adapted from (Sze et al., 2017)).

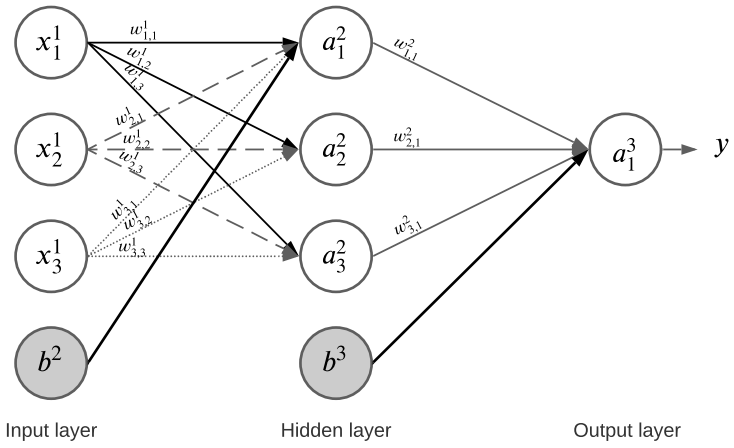



Figure 2.16: A simple ANN using three neurons with three inputs.

Usually an ANN consists of multiple neurons connected together (also called Fully Connected Neural Networks (FCs)). Figure 2.16 shows an ANN with three inputs (also called features) and three neurons, $a_1^2, a_2^2, a_3^2$ and their outputs are also the input to the last neuron at the output layer which produces the final output $y$. The input to unit $i = 1$ layer $l = 2$ is computed as:

$$a_1^2 = g(w_{1,1}^1 x_1^1 + w_{2,1}^1 x_2^1 + w_{3,1}^1 x_3^1 + b^2) \tag{2.11}$$

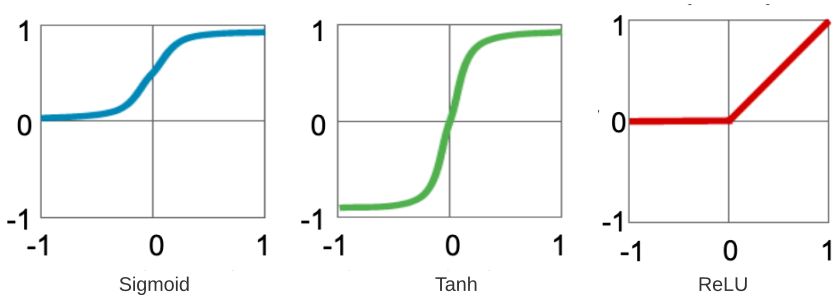



Figure 2.17: Line-plot for the commonly used non-linear activation functions (Sze et al., 2017).

Activation functions are non-linear operations to control the range of the values in the result by adjusting the values to the required range. For example the case of binary classification in which classes either belong to 0 or 1, the score returned by the function $g$ is 1 if it exceeds a certain threshold or with 0 otherwise. This activation function is called the Step function. There are other activation functions commonly used in ANNs such as: sigmoid, hyperbolic tangents (tanh) and rectifier linear units (ReLU)(Nair and Hinton, 2010). The mathematical formulations for these functions are shown below (Sze et al., 2017):

$$\text{Sigmoid } (z) = \frac{1}{1 + e^{-z}} \tag{2.12}$$

$$\text{Tanh}(z) = \frac{(e^{z} - e^{-z})}{(e^{z} + e^{-z})} \tag{2.13}$$

$$\text{ReLU}(z) = \begin{cases} 0, & z \leq 0 \\ z, & z > 0 \end{cases} \tag{2.14}$$

Figure 2.17 shows the activation function's range of results. The sigmoid function compresses the values to a range between [0, 1] where large positive values are assigned to 1, and large negative values are assigned to zero, which results in saturated values at 0 and 1. Tanh was proposed to address the saturation problem by extending the range to [-1, 1], however, the saturation problem still exists at 1 and -1. ReLU results in values with range $[0, -\infty]$ which solves the saturation problem for positive values.

Another activation function is the softmax function, which is used to normalise a vector of values into probabilities such that the probabilities range between [0, 1] and their sum equals 1. Softmax is usually used as the last activation function in a network for multi-class classification. The probabilities returned by the function represent the probability for each class, where the one with the highest probability is the predicted class. Given a set of real values $z = (z_1, z_2 \ldots, z_K)$, the softmax function is computed as follows:

$$\text{Softmax}(z) = \frac{e^{z_i}}{\sum_{j=0}^{K} e^{z_j}} \quad \text{for } i = 0, 1, \ldots, K \tag{2.15}$$

The ANN shown in Figure 2.16 represents a very simple FC model that is not used in this thesis. There are many variations to this network that usually include a large number of inputs, layers, neurons, and outputs, which is referred to as FC in this work. A number of ANN variations are presented in the following sections: A convolutional neural networks (CNNs) in section 2.5.1; recurrent neural networks (RNNs) in section 2.5.2; and an attention-based networks (Transformer) in section 2.5.3.

### 2.5.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) (LeCun et al., 1989) are a type of neural networks for processing data that are usually in a grid-like shape, such as image data represented in a 2D matrix of image elements (i.e., pixels) where each element stores a value for its intensity, ranging from 0 to 255. CNNs have been used in a variety of applications and across domains. CNNs have been used in commercial applications, such as when CNNs were used to read handwriting which helped read handwritten checks (LeCun et al., 1989). CNNs successfully detect objects and regions in images with many advanced applications, such as in self-driving cars (Hadsell et al., 2009).

The mathematical operation performed in CNNs is a linear operation called *convolution* instead of the general matrix multiplication performed in FCs. The

convolution operation takes two functions and produces another function based on a combination of the two functions. Suppose $f(i)$ and $w(i)$ are two functions, the convolution operation[9] is as follows (Goodfellow et al., 2016):

$$g(i) = (f * w)(i) = \int_0^i f(i-a)w(a)da \tag{2.16}$$

In CNN, the first argument (function $f$) in the convolution operation is referred to as the **input** and the second argument (function $w$) as the **kernel** (also called **filter**). The output of the operation $g$ is referred to as the **feature map**. The way a convolution operation is performed in a CNN is by sliding the filter matrix over the input matrix spatially and performing dot products at each spatial location. The filter slides with a specific stride, which is the number of unit steps taken over the input matrix, and each filter will produce its own feature map. Figure 2.18 shows the convolution steps to create the feature map given an input matrix (e.g., an image represented as a 2D matrix) and a kernel matrix.

[9] $*$ denotes the convolution operation.

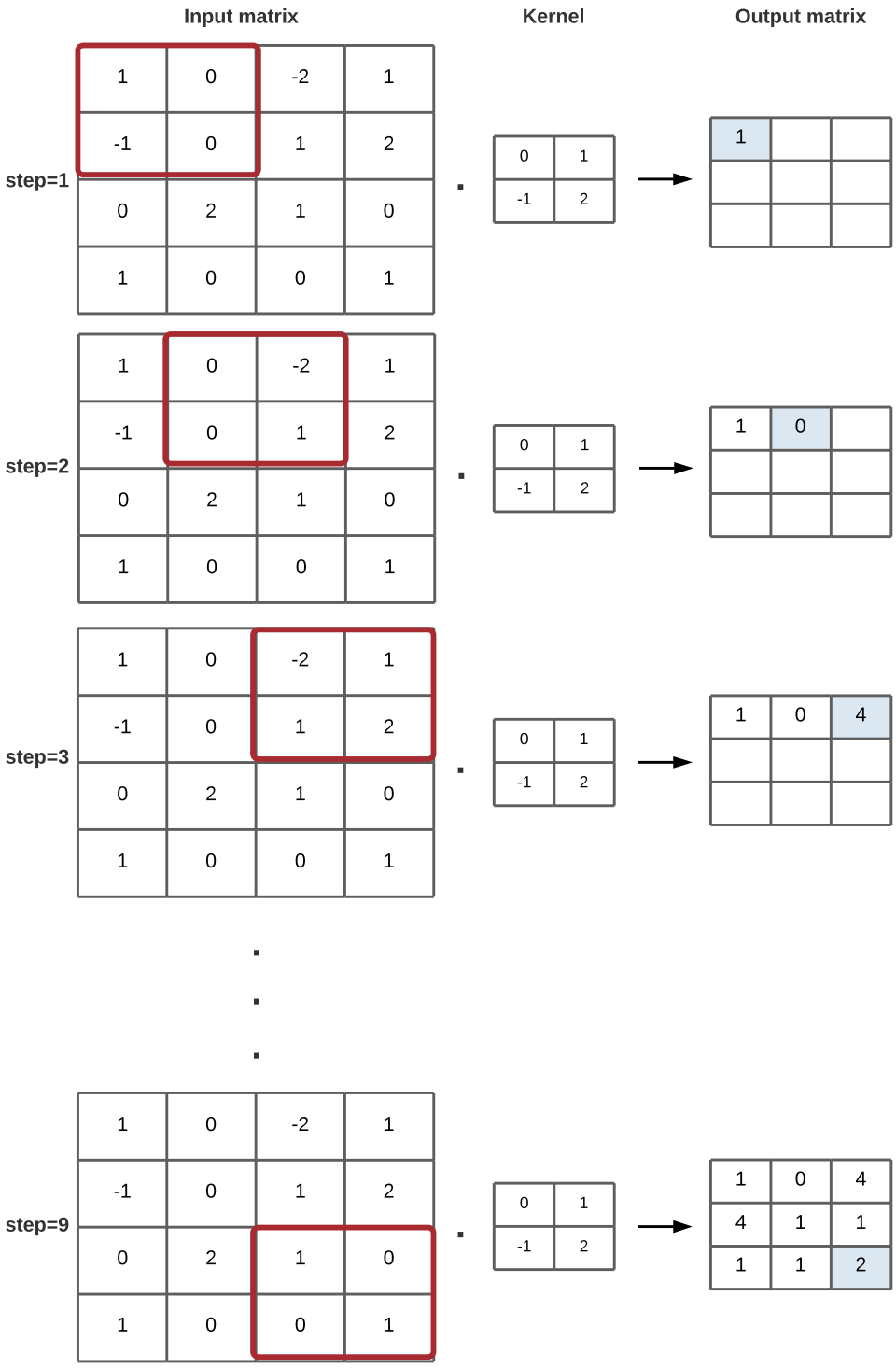



Figure 2.18: Steps in convolution operation between a 2D matrix and a 2D kernel matrix with a stride of 1.

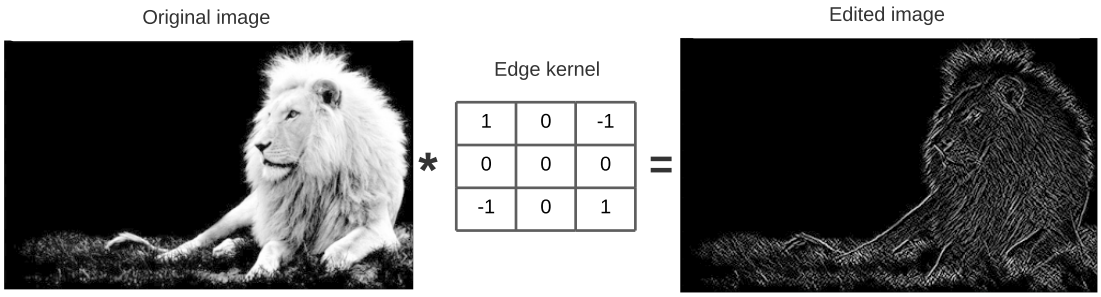



Figure 2.19: An example of convolution operation performed between an image and an edge kernel (adapted from https://beckernick.github.io/_posts/2016-09-17-convolutions).

The resulting matrix (i.e., image) can be an improved image with highlighted or blurred edges. For example, the image in Figure 2.19 when combined with an edge detecting kernel is transformed into another image with the edges emphasised. These extracted properties of the image can be used as features for a secondary task such as image classification.

Generating the feature map in a CNN starts with a kernel $W$ of size $f \times f$ sliding over the input matrix $X$. In each step, the convolution operation is performed between $W$ and a part of $X$ called the *receptive field* (i.e., the patch of the matrix shown in a red square in Figure 2.18). The resulting value represents a feature in the feature map (i.e., one step from the Figure 2.18). The value of the tuple $(i, j)$ in the feature map for layer $l$ , denoted as $y^l_{ij}$, is computed as follows:

$$y^l_{ij} = b^l + \sum_{a=1}^{f} \sum_{b=1}^{f} w^l_{ab} x^{l-1}_{(i+a)(j+b)} \tag{2.17}$$

$$a^l_{ij} = g(y^l_{ij}) \tag{2.18}$$

where $b^l$ is a bias vector and $g(.)$ is a nonlinear activation function (ReLU). ReLU normalises the feature map by keeping features with positive values and turning all other features to zero (Eq. 2.14).

The convolution step is usually followed by a pooling operation. The pooling

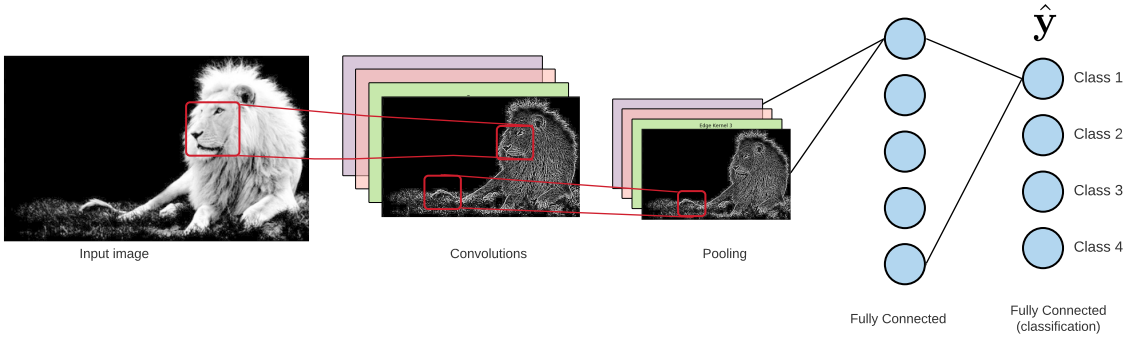



Figure 2.20: CNN model architecture for image classification (LeCun et al., 1989).

operation down-samples the resulting feature map and therefore decreases its size. The most used pooling techniques are max-pooling and average-pooling. Max-pooling picks the maximum value from the feature map, where average-pooling, as the name indicates, averages the values. For example, a feature map of size $4 \times 4$ is reduced to a size $2 \times 2$ using a $2 \times 2$ pool size (i.e., half the original feature map size).

In image classification tasks, the reduced feature map of the input matrix (e.g., an image in this case) is then passed through a FC to connect together all the features and produce a classification for the image (unlike in the convolution step where each patch of the image (receptive field) contributes a feature in the feature map). The following equation represents the FC at the end of the classification model where the output of the previous convolution layer $a_i^l$ is flattened to a vector $x_i$:

$$o_i = g(w_i x_i + b_i) \tag{2.19}$$

where $w_i$ is the corresponding trainable weight matrix; $b_i$ denotes a bias vector and $g(.)$ is the activation function.

Finally, a softmax function is used on the FC output to transform it to a probability distribution:

$$\hat{y}_i = \text{softmax}\,(o_i) \tag{2.20}$$

Figure 2.20, shows the full process from extracting the image features to predicting its class. Before moving on to another type of ANNs, a list of the common advantages

Table 2.2: Advantages and Disadvantages of CNNs (Gehring et al., 2017).

| Advantage | Disadvantage |
|---|---|
| (1) Calculations can be performed in parallel. | (1) Suffer from the vanishing and exploding of gradient. |
| (2) Capture spatial information. | (2) Require a lot of data. |
| (3) Extract relevant features automatically from the data. | (3) Take a long time to train. |
| (4) Share parameters across different parts of the input. | (4) Need deep stack of convolutional blocks to capture long dependency. |

and disadvantages of CNNs is shown in Table 2.2.

### 2.5.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986) are variant of neural networks that process sequential values $(x_1, \ldots, x_n)$. Common applications of RNNs include machine translation (Cho et al., 2014), speech analysis (Santos et al., 2016), and image captioning (Vinyals et al., 2015).

The network's capability to process a sequence of values is achieved through sharing parameters across different parts of the model (Goodfellow et al., 2016). We can think of an RNN as a recursive FC network but with a single model sharing parameters across all the time steps. Figure 2.21 shows the unfolded network in an RNN that illustrates the forward flow to compute outputs and loss. First, during the forward pass at time $i$, the input $x_i$ is mapped into the hidden representation $h_i$ then to an output $o_i$. The softmax function is used to normalise the probability of $o_i$ and the predicted output $\hat{y}_i$ is the one with the highest probability. The following

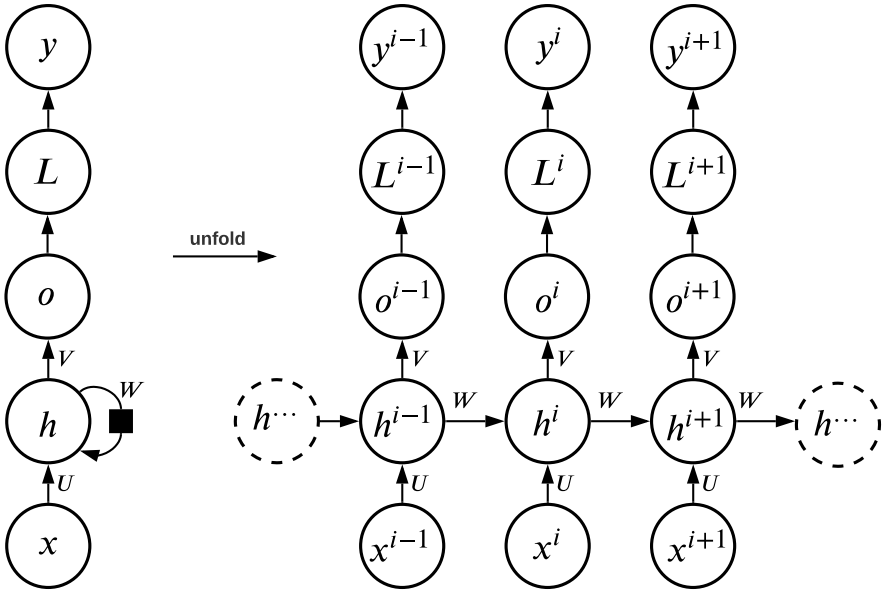



Figure 2.21: RNNs computational graph (Goodfellow et al., 2016).

equations represent the process described here:

$$a_i = b + Wh_{i-1} + Ux_i \tag{2.21}$$

$$h_{(i)} = \tanh(a_i) \tag{2.22}$$

$$o_i = \boldsymbol{c} + Vh_i \tag{2.23}$$

$$\hat{y}_i = \text{softmax}(o_i) \tag{2.24}$$

where $U$, $W$, $V$ are the weight matrices used to parametrise the connection between the input to hidden, hidden to hidden and hidden to output, respectively. $b$ and $c$ are bias vectors.

The backward flow computes the gradients, based on the loss, to update the weights. The backward operations start with computing the loss $\mathcal{L}_i$ between the predicted output $\hat{y}_i$ and the target output $y_i$ that measures how far off is the predicted output. Therefore, the final model loss is defined based on the loss of each time step. Consequently, the weights in the model are updated by back propagation through time.

A common issue with RNNs is the *vanishing of gradients*, which is caused by

multiplying the hidden states many times with the weight matrix in the forward steps and again during the back propagation steps. This leads to values escalating or vanishing which makes it impossible to train the model. The issue has been mitigated in newer variants of the RNN using gate units that control the flow of information, namely Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gates Recurrent Unit (GRU) (Cho et al., 2014), which are the networks used in the proposed labelling models in this chapter.

### LSTM

The LSTM has three gates that control the flow of information: the input gate $g$, forget gate $f$ and output gate $o$. The LSTM produces two outputs: a memory cell $c$ and a hidden state $h$. At time step $i$, the gates and outputs are defined as follows:

$$
\begin{aligned}
g_i &= \text{sigmoid}(b_g + U_{xg}x_i + W_{hg}h_{i-1}) \\
f_i &= \text{sigmoid}(b_f + U_{xf}x_i + W_{hf}h_{i-1}) \qquad (2.25) \\
o_i &= \text{sigmoid}(b_o + U_{xo}x_i + W_{ho}h_{i-1}) \\
\tilde{c}_i &= \tanh(b_c + U_{xc}x_i + W_{hc}h_{i-1}) \qquad (2.26) \\
c_i &= f_i \odot c_{i-1} + g_i \odot \tilde{c}_i \qquad (2.27) \\
h_i &= o_i \odot \tanh(c_i) \qquad (2.28)
\end{aligned}
$$

where $W_*$, $U_*$ denotes the weight matrices, $b_*$ refers to the bias. Each gate produces a value between 0 and 1 to control how much of the information is to be passed or removed. In particular, the input gate controls the information added to the cell state $\tilde{c}_i$, while the forget gate controls how much to remove or forget from the previous state $c_{i-1}$. Finally, the memory state $c_i$ can also be shut off by the output gate. The output of the model at time step $i$ is computed using the same method as in Eq. 2.20.

Table 2.3: Advantages and Disadvantages of RNNs (Goodfellow et al., 2016).

| Advantage | Disadvantage |
|---|---|
| (1) Capture sequential information in data. | (1) Suffer from the vanishing and exploding of gradient specially with a large number of time steps. |
| (2) Share parameters across time steps. | (2) Computation is sequential and therefore cannot perform calculations in parallel. |
| | (3) Less efficient in handling long sequences. |

**GRU**

Unlike the LSTM, the GRU has two gates, namely, reset $r$ and update $u$. The reset controls how much of the previous hidden state contributes to the current hidden state, while the update gate controls how much of the previous state and current hidden state is used. Therefore, their equations are updated as follows:

$$r_i = \text{sigmoid}\left(b_r + W_{xr}x_i + W_{hr}h_{i-1}\right) \tag{2.29}$$

$$u_i = \text{sigmoid}\left(b_u + W_{xu}x_i + W_{hu}h_{i-1}\right) \tag{2.30}$$

$$\tilde{h}_i = \tanh\left(b_h + W_{xh}x_i + W_{hh}\left(r_i \odot h_{i-1}\right)\right) \tag{2.31}$$

$$h_i = u_i \odot h_{i-1} + (1 - u_i) \odot \tilde{h}_i \tag{2.32}$$

The output at the $ith$ time step can be computed by applying a softmax function to $h_i$ as shown in Eq. 2.20.

Finally, before moving on to present and describe another family of ANNs, a summary of the common advantages and disadvantages of RNNs is shown in table 2.3.

### 2.5.3 Attention-based Neural Networks

Attention-based neural networks (Transformer (Vaswani et al., 2017)) are composed of linear layers, attention mechanisms and normalisation functions. Like the convolutional networks and unlike recurrent networks, Transformers do not employ any

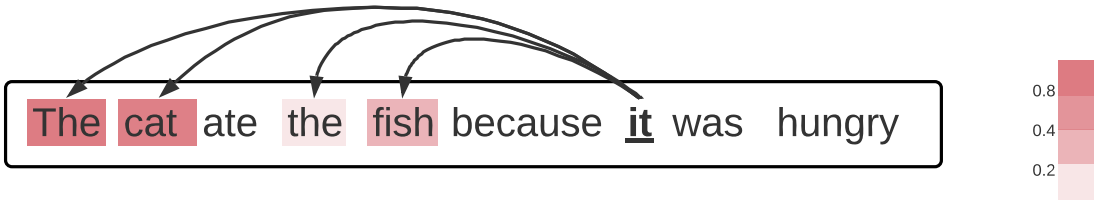



Figure 2.22: An example of the weights given to the words in a sentence in relation to the word "it" using self-attention.

recurrent flow of information. This allows the network to train in parallel and reduces the overhead of computations. Transformers have been successfully applied to wide range of tasks such as machine translation (Akoury et al., 2019), text representation (Devlin et al., 2019), and object detection (Carion et al., 2020).

The transformer is composed of $L$ transformer blocks, each comprising two components: *multi-head self-attention* and *feed forward* (FF). The first component in the transformer block is the multi-head self-attention layer, where it encodes the sequence and allows the model to attend to parts of the sentence. This is where the degree of association between the words is encoded in the Transformer. For example, the word "it" in the sentence in Figure 2.22 is referring to "cat" and "the" and therefore the self-attention for the word "it" gives more weight for association with "cat" and "the" than with the word "fish". Such relationships are encoded using the self-attention technique. Given a sequence $x = (x_1, x_2, \ldots, x_n)$, attention is computed as follows:

$$Q = W^q x, \quad K = W^k x, \quad V = W^v x \tag{2.33}$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \tag{2.34}$$

where $W^*$ are weight matrices and $d$ is the hidden dimension. The attention is computed $h$ times instead of once hence the name *multi-head self-attention*. The $h$ computations of attentions are performed in parallel and their resulted weight vectors are concatenated to form one attention vector that is ready to be passed to

Table 2.4: Advantages and Disadvantages of Transformers (Vaswani et al., 2017).

| Advantage | Disadvantage |
|---|---|
| (1) Calculations can be performed in parallel and thus shorter training time. | (1) Accept a fixed-length inputs. |
| (2) Capture longer dependencies further away in a sequence. | (2) Not all attention heads are useful. |
| (3) Share parameters across layers. | (3) Large amount of calculations needed for attention. |

the next component in the transformer block:

$$\text{MultiHead}(Q, K, V) = Z = \text{Concatenate}\left(\text{head}_1, \ldots, \text{head}_h\right) W^O \tag{2.35}$$

$$\text{head}_\text{i} = \text{Attention}\left(Q_i, K_i, V_i\right) \tag{2.36}$$

Another variation of self-attention is *masked self-attention*, which is mainly used at the decoder. This variation is similar to the attention described earlier except that masked self-attention is only allowed to attend to previous words and not future words.

The second component of the transformer block is a FF layer which is applied to each position in the sequence independently. It consists of two linear layers (i.e., FCs) as follows:

$$\text{FF}(Z) = \max\left(0, ZW_1 + b_1\right) W_2 + b_2 \tag{2.37}$$

where $W_*$ are weight matrices and $b_*$ are bias vectors.

Residual connections (He et al., 2016) and layer normalisation (Lei Ba et al., 2016) are applied after every component in the Transformer block. Layer normalisation stabilises the weights in neural networks by normalising the values across the hidden dimension. Residual connection and normalisation are performed by adding the input to the layer to its output then normalised across the features.

Table 2.4 lists the advantages and disadvantages of using a Transformer architecture.

## 2.6 Summary

This chapter has presented a survey of unsupervised methods for analysing and modelling a large collection of data using topic models. In section 2.1, a number of topic model variants were described. Further, this section also covered several extensions to the topic model to accommodate a specific task, or address a drawback. A variety of evaluation approaches for the output of the topic model have been presented in section 2.2. Communicating the output of the topic model to the user is critical, and various types of representation were presented in section 2.3, including individual topic representations and tools for representing a whole document collection. Section 2.4 presented the work done on improving topics through post-processing. Finally, section 2.5 includes a brief introduction of ANNs in addition to the details of various ANNs including CNNs, RNNs and Transformers.

# 3

# RE-RANKING AUTOMATICALLY GENERATED TOPICS

## 3.1 Introduction

As shown in the previous chapter (section 2.3.1), in the standard approach to representing topics, the top $n$ words are shown with the highest probability based on the topic (Blei and Lafferty, 2009a,b). However, these words may not be the ones that are the most informative about the topic. The hypothesis is that some words that are relevant to a particular topic have not been assigned a high probability because of data sparseness or low frequency in the training corpus (Chang et al., 2009, Lau et al., 2010). Therefore, the goal is to identify these words and re-rank the list that represents the topic to make it more comprehensible.

In section 2.4.1, the literature review presented a range of approaches to re-ranking the topic terms. In previous work on topic-word re-ranking, its effectiveness was

evaluated in the context of the application of topic models, such as topic visualisation. This chapter introduces a direct and systematic approach to evaluating re-ranking methods. It compares several word ranking methods and evaluates them based on an information retrieval (IR) task that does not rely on human judgement. The re-ranked words are used to form a query and retrieve a set of documents from a collection. The effectiveness of the word re-ranking is then evaluated in terms of how well it retrieved documents in the collection in relation to the topic.

The rest of this chapter is organised into seven sections. First, section 3.2 provides a formal definition of the word re-ranking task, followed by a definitions of multiple approaches to word re-ranking in sections 3.3 and 3.4. The evaluation approach employed for the re-ranked topics is described in section 3.5. Sections 3.6 and 3.7 includes the results and discussion. Finally, the conclusion of this chapter is provided in section 3.8.

## 3.2 Task Definition

Topics are represented with a subset of their total probability distribution. Therefore, this subset (i.e., the top 10 terms) is assumed to be representative. However, these words may not be the most informative about the topic. For example, Table 3.1 shows topics that are represented by the 30 most probable words. The words displayed in bold font are more general and less informative (e.g., word with high document frequency), while the remaining words are more likely to represent a coherent thematic subject. In topic two, relevant words (e.g., investment and fund) have been assigned with lower probability compared with less informative words (e.g., percent and million). As a result, because these words do not appear in the top 10 words, re-ranking is used to address this issue.

Table 3.1: Examples of topics represented by the 30 most probable words from NYT. Less informative words are shown in **bold**.

| | |
|---|---|
| **Topic 1** | space museum **years** history science earth mission **could** art shuttle universe flight **people theory world** radar crew **site** pincus plane **three** scientists **day century** pilot exhibit **back anniversary** landing **project** |
| **Topic 2** | **percent million** market company stock **billion sales** bank shares **price business** investors **money** share **companies rates** fund **interest rate quarter prices** investment funds financial amp analysts **growth industry york** banks |
| **Topic 3** | film **even** movie **world** stars **man much little story good way** star **best** show **see well seems american people love** hollywood director **big ever rating though great seem** production **makes** |
| **Topic 4** | **officials agency office report department** investigation government **former** federal **charges secret information card** cia law agents security **documents case** investigators **official** fraud intelligence illegal **commission service** police **cards** enforcement attorney |

Given a trained topic model over corpus $\mathcal{D}$, two probability distributions are generated:

1. Document-topic ($\theta$) as a matrix with dimensions $[M \times K]$, and
2. Topic-word ($\phi$) as a matrix with dimensions $[K \times V]$

where $K$ denotes the number of topics, $M$ denotes the number of documents in the collection, and $V$ denotes the vocabulary size.

To re-rank the topics' terms, the $\phi$ matrix is used to extract the topics. Each vector in $\phi$ is a topic with an index that is equal to the number of words in the vocabulary. Each index in the vector contains the probability of a word belonging to a topic. Therefore, a topic $t$ is defined as $\hat{\varphi}_t = \phi_{t,*}$ and the probability of a word $w$ given a topic $t$ produced by a topic model, which is denoted as $\hat{\varphi}_{t,w}$.

In this thesis, LDA is the topic model approach used, but re-ranking can be applied to any topic model that estimates probabilities of words associated with topics.

## 3.3 Re-Ranking Using Corpus Statistics

This section explores a range of methods for word re-ranking based on the main approaches that have been applied to the problem (see section 2.4.1). The following methods are used to re-rank topic words.

**Original LDA Ranking ($\mathbf{R_{Orig}}$)** The most obvious and commonly used method for ranking words associated with a topic is to use $\hat{\varphi}_{w,t}$ to score each word, i.e., $score_{w,t} = \hat{\varphi}_{w,t}$. The ranking generated by this scoring function is equivalent to choosing the $n$ most probable words for the topic and is referred to as $R_{Orig}$.

**Normalised LDA Ranking ($\mathbf{R_{Norm}}$)** The first re-ranking method is a simple extension of $R_{Orig}$, which represents approaches that normalise the probability of a word given a particular topic by the sum of probabilities of that word across all topics (Song et al., 2009, Taddy, 2012). This measure is computed as follows:

$$score_{w,t} = \frac{\hat{\varphi}_{w,t}}{\sum_{j=1}^{K} \hat{\varphi}_{w,j}} \tag{3.1}$$

where $K$ denotes the number of topics in the model. This approach scales the importance of words based on their overall occurrence in all topics in the model and down-weights those that occur frequently.

**TF-IDF Ranking ($\mathbf{R_{TFIDF}}$)** The second re-ranking method was proposed by Blei and Lafferty (2009a) and represents methods that combine information about the probability of a word in a single topic, including information about its probability across all topics (Bischof and Airoldi, 2012, Chuang et al., 2012, Sievert and Shirley, 2014). Blei and Lafferty re-ranked each word as follows:

$$score_{w,t} = \hat{\varphi}_{w,t}\ log\frac{\hat{\varphi}_{w,t}}{\left(\prod_{j=1}^{K}\hat{\varphi}_{w,j}\right)^{\frac{1}{K}}} \tag{3.2}$$

**Inverse Document Frequency (IDF) Ranking ($\mathbf{R_{IDF}}$)** The final word re-ranking method explored in this section is a variant on the previous method, which considers a word's distribution across documents rather than topics. This method has not been explored in previous research. In this approach, each word is weighted by the Inverse Document Frequency (IDF) score across the corpus used to train the topic model:

$$score_{w,t} = \hat{\varphi}_{w,t}\ log\frac{|\mathcal{D}|}{|\mathcal{D}_w|} \tag{3.3}$$

where $\mathcal{D}$ is the entire document collection, and $\mathcal{D}_w$ are the documents within $\mathcal{D}$ containing the word $w$.

## 3.4 Re-Ranking Using Distributional Semantics

This section introduces word re-ranking methods in addition to those presented earlier in section 3.3. Previously proposed ranking methods are based on statistical information extracted from the dataset used to create the topics. Alternative ranking methods based on distributional semantic models are proposed. A short overview of word embeddings, dense real-valued representations of words learned using distributional semantic models, are discussed next.

### 3.4.1 Embedding Space

Natural language features, such as words, letters, and digits, are discrete. Therefore, they are not represented as real-value vectors but as a one long *one-hot* vector of the entire vocabulary. For example, each word is represented by a vector of length equal to $V$ ($V$ is the total number of words in the corpus). Each index in the vector

belongs to a word from the vocabulary and only one index can be set to 1 while the rest of the elements in the vector are set to zero. The $v^{th}$ word in the vocabulary is represented by a vector where $w_v = 1$ and $w_i = 0$ for $i \neq v$. This sparse or high-dimensional word representation does not represent relationships between the words (Goldberg and Hirst, 2017). Instead, a dense nonlinear representation of the words is used, in which each word is represented using a vector of dimension $d$ that is usually smaller than the vocabulary size. For example, the word "*cat*" in a one-hot encoding representation over a vocabulary of 20,000 words will be represented as a vector of size $[1 \times 20,000]$. The same word can be represented as a vector of a size $[1 \times 128]$ that is embedded in a semantic space with 128 dimensions, which is also called the embedding space.

Embedding-based representation is useful because it detects semantic relationships between words and therefore gives them similar vectors (e.g., *queen* and *king*) (Goldberg and Hirst, 2017). Such capabilities are useful in many applications, such as predicting the word's nearest neighbours, which can be used to expand search queries for similar words. Word embeddings are popular representations that have been used in numerous NLP tasks, such as machine translation (Mikolov et al., 2013a), document classification (Sebastiani, 2002), and named entity recognition (Turian et al., 2010). Because of the success of word embeddings, this section explores the use of such word representations in re-ranking the topic's words.

Pre-trained word embeddings are used in the re-ranking metrics. Pre-trained word embeddings were created using large datasets that incorporated information from a secondary source, such as Wikipedia, beyond the dataset used to generate the topics. In this experiment, the topic's words are re-ranked by placing semantically similar words near each other. Three commonly used word embeddings are explored: (1) *Word2Vec* (Mikolov et al., 2013b); (2) *Glove* (Pennington et al., 2014); and (3) *FastText* (Bojanowski et al., 2016).

The Word2vec model learns the vector representation of words based on the

continuous bag-of-words (CBOW) and skip-gram (SKIP-G) models. These models efficiently learn quality vector representations from a large amount of unstructured text data using a simple neural network architecture (Mikolov et al., 2013b). A pre-trained Word2vec model was used. The model consists of three million word embeddings with 300 dimensions, which were trained on the Google News dataset of approximately 100 billion words.[1]

Glove word representations capture global corpus co-occurrence statistics using a global log-bilinear regression model (Pennington et al., 2014). A pre-trained trained model with 300 dimension vectors for 1.9 million words was trained on the Common Crawl dataset of around 42 billion words.[2]

FastText is a fast approach to learning word representations based on skip-gram models (Bojanowski et al., 2016). FastText can learn representations of out-of-vocabulary (OOV) words by splitting the word into character-level n-grams and calculating the final word embedding as the average of its n-grams. However, in this thesis, a static pre-trained version of FastText was used, which did not accommodate this OOV words function[3]. The pre-trained model consisted of 2.5 million 300-dimension vectors trained on 600 billion words.[4]

### 3.4.2 Embedding-based Ranking Methods

Re-ranking a topic's words starts by extracting the words' embedding vectors from one of the models described in section 3.4.1. Given two words, $w_i$ and $w_j$, each is represented by a vector ($\mathbf{w}_i$ and $\mathbf{w}_j$). The cosine similarity is used to calculate the similarity between the two words as follows:

$$\text{sim}(\mathbf{w}_i, \mathbf{w}_j) = \mathbf{w}_i^\top \mathbf{w}_j \tag{3.4}$$

[1] code.google.com/p/Word2vec/
[2] https://nlp.stanford.edu/projects/Glove/
[3] Later in this chapter, the number of OOV words for each dataset is reported.
[4] https://github.com/facebookresearch/FastText/

Two scores are computed for each word in the top $N$ topic words, the word similarity (*pairwise_sim*) and the centroid similarity (*centroid_sim*), which result in the ranking methods $R_{Pair}$ and $R_{Cent}$, respectively. The *pairwise_sim* is the pairwise cosine similarity between the word's vector and the vectors of the top $N$ words in the topic (Eq. 3.5). The *centroid_sim* is computed as the cosine similarity between the word vector and the centroid vector of the top $N$ words from the topic (Eq. 3.6).

$$\text{pairwise\_sim}_{w,t} = \frac{1}{N}\sum_{j=1}^{N}\text{sim}(\mathbf{w}, \mathbf{w}_j) \tag{3.5}$$

$$\text{centroid\_sim}_{w,t} = \text{sim}(\mathbf{w}, \mathbf{e}) \tag{3.6}$$

where $\mathbf{e}$ is the normalised centroid vector of $N$ top words in the topic $t$. Note that the top $N$ words are included in the re-ranking method, and only the top $n$ words are selected to represent the topic. For example, the ranking method considers the top 50 words ($N = 50$) in the topic. After re-ranking, the top 10 words ($n = 10$) are selected to represent the topic. Multiple combinations of the product of ($R_{Pair}$, $R_{Cent}$, $\hat{\varphi}$, and IDF) are explored as term-ranking methods.

The vocabulary size included in word embedding models is limited. It consists of the vectors for approximately 3 million, 400 thousand, and 2.5 million words in Word2vec, Glove and FastText, respectively. Therefore, terms that are not available in the embedding model (i.e., OOV) will remain at the default weight given by LDA.

To better understand the effects of re-ranking topic words, consider the topics with various representations (see Table 3.2) that were a result of re-ranking. The first row of each topic represents the baseline rank produced by the topic model ($R_{Orig}$), and the other rows show the topic after re-ranking using corpus-based metrics in Equations 3.1, 3.2 and 3.3, respectively, in addition to ranking using distributional semantics, as in Equations 3.5 and 3.6. The bold words included in the original ranking ($R_{Orig}$) were down-weighted and removed by at least two methods. Underlined words were weighted higher by a re-ranking method and included in the

Table 3.2: Examples of topic representations produced using various ranking approaches. Words in the $R_{Orig}$ representation that were removed by at least two methods are shown in **bold**. Words that were ranked higher by the other approaches and included in the topic representation are underlined.

| | Method | Topic 1 |
|---|---|---|
| Corpus statistics | $R_{Orig}$ | company **million** executive **year** business **number chief firm group private** |
| | $R_{Norm}$ | nardelli deductible semel fisch earmark backdating reit vornado weichert citibank |
| | $R_{TFIDF}$ | company million executive firm companies equity firms taxes financial broker |
| | $R_{IDF}$ | company million executive firm business listed number financial chief companies |
| Distributional semantics | $R_{Pair}$ | business company investment companies financial firms corporate pay firm funds |
| | $+\hat{\varphi}$ | company million business executive firm companies financial year private market |
| | $+\hat{\varphi} + IDF$ | company million executive business firm companies financial firms private equity |
| | $R_{Cent}$ | business street corporation company money office public broker chief private |
| | $+\hat{\varphi}$ | company business executive year chief private firm office companies million |
| | $+\hat{\varphi} + IDF$ | company business executive firm chief private broker financial companies office |
| | **Method** | **Topic 2** |
| Corpus statistics | $R_{Orig}$ | team **year first last** players **time** football **sports play back** |
| | $R_{Norm}$ | federer nascar touchdown earnhardt mickelson nadal selig belichick henin roddick |
| | $R_{TFIDF}$ | team players football giants sports cup golf bowl championship manning |
| | $R_{IDF}$ | team players football sports giants golf cup race year woods |
| Distributional semantics | $R_{Pair}$ | game season play playing team players teams championship player going |
| | $+\hat{\varphi}$ | team players year first last football play time game season |
| | $+\hat{\varphi} + IDF$ | team players football game play season win sports golf teams |
| | $R_{Cent}$ | teams going back club never football game way team get |
| | $+\hat{\varphi}$ | team year football players time back last going game first |
| | $+\hat{\varphi} + IDF$ | team football players teams year game race golf sports club |

topic representation.

## 3.5 Evaluation of Topic Interpretability via Document Retrieval

This section presents an automatic evaluation approach to different topic representations obtained by re-ranking the topic words. The automated evaluation is based on an IR task in which the re-ranked topic words are used to form a query and retrieve documents that are relevant to the topic. The motivation for this approach is that the most effective re-rankings are the ones that can retrieve documents related to the topic, while ineffective re-rankings are not able to distinguish them from other documents in the collection. This evaluation method does not rely on human

judgement, therefore, it is convenient for the repetitive evaluation of ranking metrics.

### 3.5.1 Evaluation Process

Figure 3.1 illustrates the evaluation process. First, the evaluation approach assumes a given document collection (step 1) in which each document is mapped to a label (or labels) indicating its topic. For example, in Figure 3.1 the NYT annotated dataset is used, which has manually assigned labels to its articles (step 2). These labels are referred to as *gold standard topics* to distinguish them from the automatically generated topics created by the topic model.

A set of automatically generated topics is created by running a topic model over a document collection, such as the NYT annotated dataset. For each gold standard topic, a set of all documents labelled with that topic is created (step 3). The document-topic distribution created by the topic model is then used to identify the most probable automatically generated topic within that set of documents (steps 4 and 5). This is achieved by summing the document-topic distributions and choosing the automatically generated topic that has the highest value (steps 6, 7, and 8). Even though the documents under the gold topic are independent of each other, identifying the dominant topic for these documents through the described approach can still be considered a helpful approach.

A query is then created by selecting the re-ranked top $n$ words, using one of the re-ranking metrics described in sections 3.3 and 3.4, from that automatically generated topic (steps 9 and 10) and using it to retrieve a set of documents from the collection (steps 11 and 12). The set of retrieved documents is then compared with the set of all documents labelled with the gold standard label using trec_eval[5] (steps 13 and 14).

[5] https://github.com/usnistgov/trec_eval

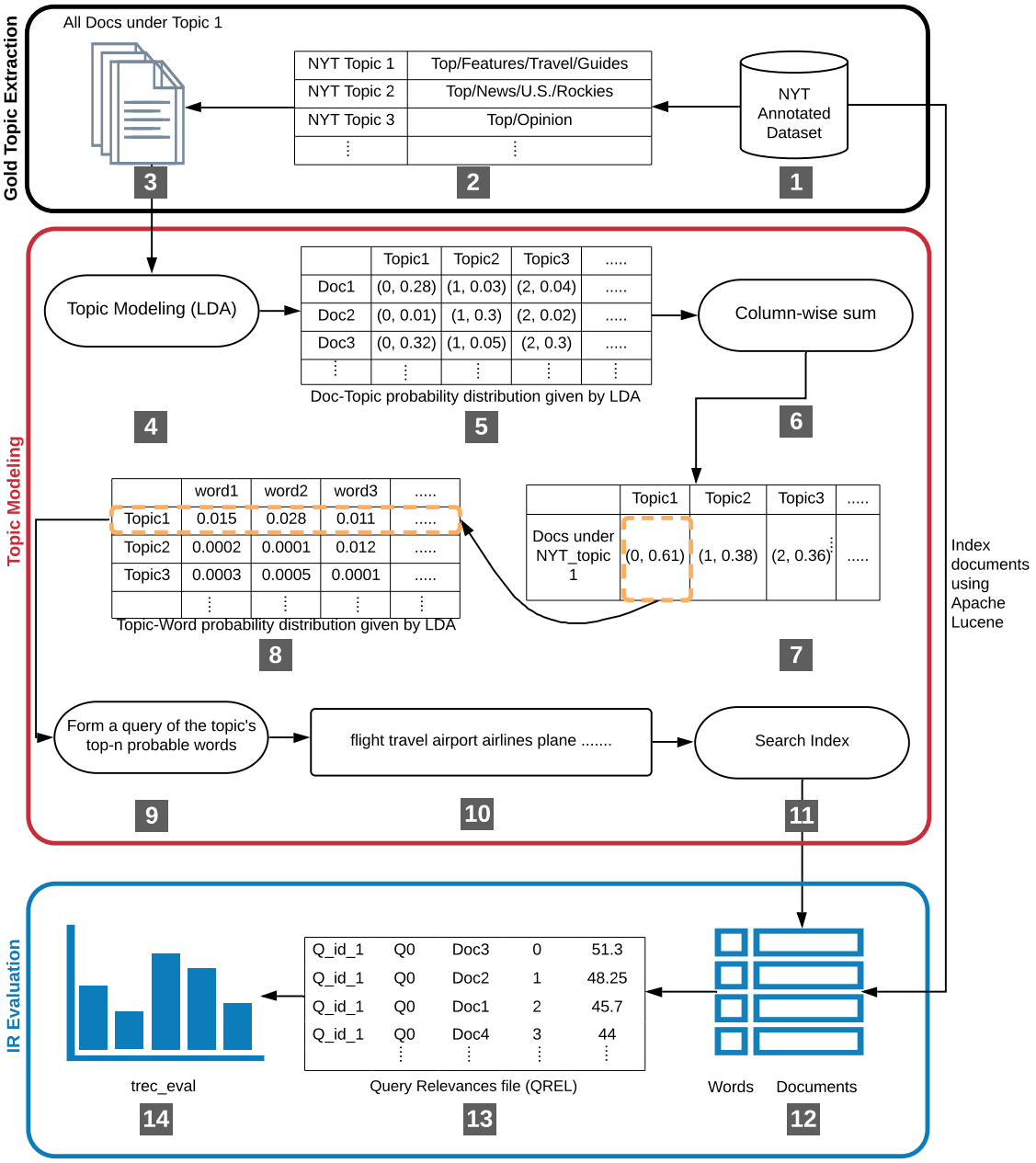


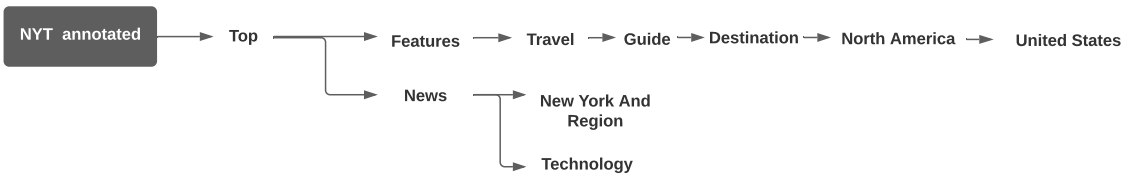

Figure 3.1: Illustration of the IR-based evaluation approach used to measure the quality of the re-ranked topics.

### 3.5.2 Datasets

The evaluation was conducted using datasets that represented documents collected from a wide range of domains: news articles, scientific literature and online reviews.

#### New York Times

A subset of the New York Times (NYT) annotated dataset[6] consisting of approximately 39,000 articles was used in this experiment. This collection contains news articles from the *New York Times* labelled with 1,746 topics which were used as gold standard labels. These labels, which we refer to as *NYT_topics*, belong to a controlled set of topic categories. They have been manually verified by the production staff of NYTimes.com. Each article has at least one NYT_topic, and the articles are organised into a topic hierarchy. Examples of NYT_topics include the following:

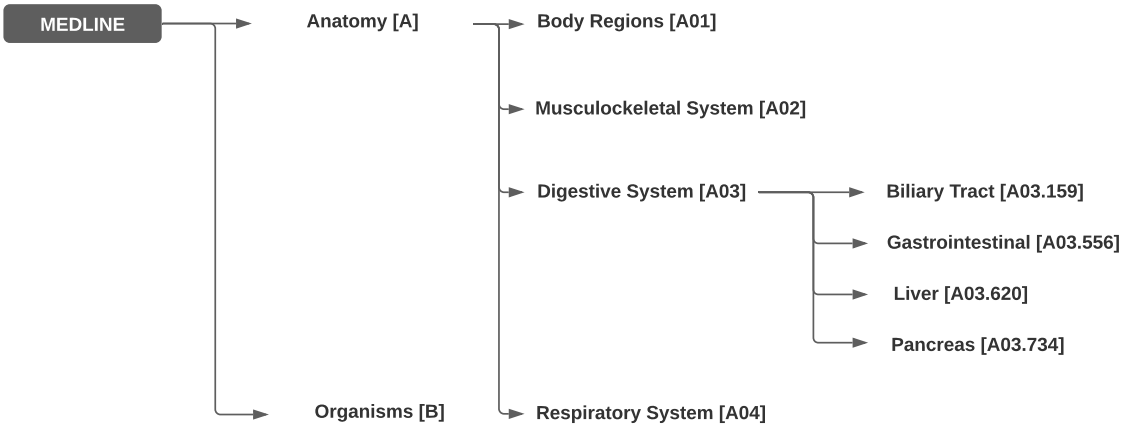



Because the hierarchy into which the topics are organised is quite deep in some places, each topic is truncated to the top four levels of the hierarchy to control the number of topics. For example, the topic *Top/Features/Travel/Guides/Destinations/North America/United States* was truncated to *Top/Features/Travel/Guides*. This process produced a total of 132 truncated NYT_topics. The number of articles associated with each of the 132 NYT_topics ranged from 1 to 18,489. To avoid NYT_topics that were associated with small numbers of documents, the 50 NYT_topics that were associated with the most documents were used, which resulted in NYT_topics, each of which were associated with at least 560 documents.

[6] https://catalog.ldc.upenn.edu/LDC2008T19

### MEDLINE

MEDLINE[7] is a primary medical literature resource that includes more than 25 million records of publications in medicine and related fields from 1805 to the present. MEDLINE uses more than 19,000 Medical Subject Headings (MeSH) to index and catalogue publications in a hierarchical structure. Each publication is associated with a set of MeSH codes that describe the content of the publication. A subset of the MeSH hierarchy is shown below:




In a subset of MEDLINE, the 50 most frequently used MeSH codes with the greatest number publications from 2017 were extracted. This set of codes was referred to as *MeSH_topics*.

### Amazon Product Reviews

The Amazon Product Reviews dataset (McAuley et al., 2015)[8] contains reviews of products purchased from the Amazon website. The reviews are organised into 24 top-level categories, each of which is divided into sub-categories. The number of sub-categories ranges from 1 to approximately 1,900. A sample of the categories and their sub-categories is shown in Figure 3.2.

[7] https://www.nlm.nih.gov/databases/download/pubmed_medline.html
[8] http://jmcauley.ucsd.edu/data/amazon

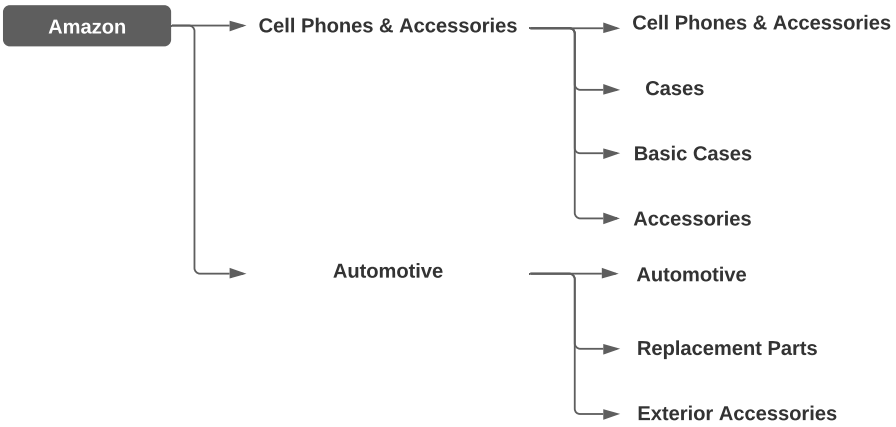



Figure 3.2: Sample categories from Amazon.

Eight main categories were selected (*Cell Phones & Accessories, Electronics, Movies & TV, Musical Instrument, Office Products, Pet Supplies, Tools & Home Improvement, and Automotive*) and from each, 10 sub-categories with the highest number of reviews where extracted, which yielded 76 distinct sub-categories. This set of categories were referred to as *AMZ_topics*. Five thousands product reviews were extracted from each main category. Each review must have belonged to at least one category in the 50 most frequent in the AMZ_topics, which resulted in 40,000 reviews.

### 3.5.3 Experimental Settings

Each of the datasets was indexed using Apache Lucene[9]. The articles were tokenised, and stop words were removed. Words occurring in fewer than five or more than half of the documents were also removed to control for rare and common words, respectively. The statistics of the datasets are shown in Table 3.3.

In each dataset, LDA was used to generate topics, and the number of topics for each dataset was set based on optimising for coherence, which yielded 35 in NYT, 45 in MEDLINE and 35 in Amazon. The automatically generated topics that was the most strongly associated with each of the gold topics (i.e., 35 NYT_topics, 45

[9] http://lucene.apache.org/

Table 3.3: Datasets statistics.

| Dataset | Documents | Distinct Words |
|---|---|---|
| NYT Annotated | 39,218 | 60,339 |
| MEDLINE | 23,640 | 18,571 |
| Amazon | 40,000 | 24,943 |

MeSH_topics and 35 AMZ_topics) were identified by applying the process described in section 3.5.1. The top 5, 10 and 20 words on this topic were used to form a query that was submitted to Lucene. The BM25 retrieval model (Robertson, 2004) was used to measure the similarity between the document and a given query. The documents that were retrieved by applying these queries were compared with the entire set of documents labelled with the dataset topics (i.e., NYT_topic, MeSH_topics, or AMZ_topics) by computing Mean Average Precision (MAP)[10], which is commonly used as a single metric to summarise IR system performance.

## 3.6 Results

### 3.6.1 Corpus-based Ranking Results

Queries were created using the top 5, 10, and 20 topic words and applying $R_{Orig}$, $R_{Norm}$, $R_{TFIDF}$, and $R_{IDF}$ re-rankings and applied to each of the three datasets (section 3.5.2). The results are shown in Table 3.4.

Re-ranking words using $R_{TFIDF}$ and $R_{IDF}$ consistently enhanced retrieval performance compared with the default ranking ($R_{Orig}$). $R_{TFIDF}$ produced the best results in most of the configurations, except when five words were used in the Amazon corpus, where $R_{IDF}$ outperformed the other re-ranking methods. Re-ranking using $R_{Norm}$ was less effective than all the other rankings, including the default ranking. The relative performance of the four approaches was generally stable when the number of words used to form the query was varied across the three datasets, which represented the different genres of text used in this experiment. These results

[10]MAP is computed using the *trec_eval* tool: http://trec.nist.gov/trec_eval/

Table 3.4: Results of the experiment in which the top 5, 10 and 20 ranked words were used to form a query. The words were ranked using the methods described in section 3.3. The **bold** font denotes the highest score among the ranking metrics in a particular cardinality and dataset.

| Dataset | New York Times | | | MEDLINE | | | Amazon | | |
|---|---|---|---|---|---|---|---|---|---|
| #Words | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 |
| $R_{Orig}$ | 0.095 | 0.116 | 0.126 | 0.142 | 0.152 | 0.150 | 0.023 | 0.020 | 0.026 |
| $R_{Norm}$ | 0.046 | 0.061 | 0.072 | 0.020 | 0.029 | 0.037 | 0.020 | 0.015 | 0.014 |
| $R_{TFIDF}$ | **0.136** | **0.142** | **0.139** | **0.174** | **0.161** | **0.166** | 0.021 | **0.024** | **0.028** |
| $R_{IDF}$ | 0.119 | 0.129 | 0.132 | 0.158 | 0.158 | 0.164 | **0.024** | 0.024 | 0.027 |

demonstrated that topic-word re-ranking produced words that were effective in discriminating documents that described a particular topic from those that did not.

### 3.6.2 Embedding-based Ranking Results

The queries in this section were created in the same way as described in the previous section. However, the embedding-based re-ranking methods (section 3.4) were applied. The MAP scores achieved by each method are presented in Table 3.5.

Table 3.5 shows that re-ranking the topic words using the similarity metrics, $R_{Pair}$ or $R_{Cent}$, as an individual metric did not improve the topics. This results occurred in all cardinalities and datasets. The similarity metrics reported a higher scores when they were combined with the original weights given by the topic model ($\hat{\varphi}$). However, in most cases, it did not surpass the scores reported when using the original topic model weights were used on their own. Following the addition of $IDF$ to the ranking metric, the produced topics outperformed the baselines ranking in most of the cases. $R_{Pair}$ was shown to be more useful and a better measure of similarity between the terms compared with $R_{Cent}$. Table 3.5 also shows a discrepancy in the performance of the embedding models. Topics that were re-ranked using FastText reported higher scores than the topics re-ranked using Word2vec and Glove.

Table 3.5: Results of the experiment in which the top 5, 10 and 20 ranked words were used to form a query. The words were ranked using the methods described in section 3.4. **Bold** font denotes the highest score in an embedding model, and underlines denotes the highest score in a column across the metrics and embedding models.

| Dataset | New York Times | | | MEDLINE | | | Amazon | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Number of Words | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 | |
| $R_{Orig}$ | 0.095 | 0.116 | 0.126 | 0.142 | 0.152 | 0.150 | 0.023 | 0.020 | 0.026 | |
| $R_{Pair}$ | 0.088 | 0.097 | 0.030 | 0.091 | 0.111 | 0.132 | 0.006 | 0.009 | 0.023 | Word2vec |
| $R_{Pair} + \hat{\varphi}$ | 0.093 | 0.108 | 0.118 | 0.153 | 0.153 | 0.145 | 0.023 | 0.020 | 0.027 | Word2vec |
| $R_{Pair} + \hat{\varphi} + IDF$ | **0.101** | **0.116** | **0.120** | **0.155** | **0.160** | **0.152** | **0.023** | 0.026 | 0.026 | Word2vec |
| $R_{Cent}$ | 0.058 | 0.082 | 0.105 | 0.093 | 0.106 | 0.123 | 0.014 | 0.020 | 0.025 | Word2vec |
| $R_{Cent} + \hat{\varphi}$ | 0.080 | 0.102 | 0.111 | 0.153 | 0.147 | 0.137 | 0.019 | 0.018 | 0.027 | Word2vec |
| $R_{Cent} + \hat{\varphi} + IDF$ | 0.090 | 0.104 | 0.115 | 0.150 | 0.152 | 0.137 | 0.020 | **0.027** | **0.029** | Word2vec |
| $R_{Pair}$ | 0.060 | 0.069 | 0.087 | 0.035 | 0.054 | 0.087 | 0.003 | 0.004 | 0.010 | Glove |
| $R_{Pair} + \hat{\varphi}$ | 0.096 | 0.106 | 0.120 | 0.125 | 0.141 | 0.146 | 0.020 | 0.020 | 0.021 | Glove |
| $R_{Pair} + \hat{\varphi} + IDF$ | **0.109** | **0.122** | 0.122 | 0.133 | **0.150** | **0.156** | **0.024** | 0.021 | **0.024** | Glove |
| $R_{Cent}$ | 0.028 | 0.040 | 0.073 | **0.036** | 0.045 | 0.073 | 0.003 | 0.005 | 0.009 | Glove |
| $R_{Cent} + \hat{\varphi}$ | 0.093 | 0.102 | 0.114 | 0.117 | 0.122 | 0.132 | 0.019 | 0.018 | 0.018 | Glove |
| $R_{Cent} + \hat{\varphi} + IDF$ | 0.098 | 0.110 | **0.124** | 0.127 | 0.136 | 0.145 | 0.023 | **0.022** | 0.022 | Glove |
| $R_{Pair}$ | 0.088 | 0.101 | 0.112 | 0.107 | 0.121 | 0.126 | 0.006 | 0.008 | 0.018 | FastText |
| $R_{Pair} + \hat{\varphi}$ | 0.097 | 0.114 | 0.125 | 0.153 | 0.155 | 0.155 | 0.021 | 0.023 | **0.029** | FastText |
| $R_{Pair} + \hat{\varphi} + IDF$ | **0.107** | **0.123** | **0.127** | 0.160 | **0.161** | **0.161** | **0.023** | 0.023 | 0.026 | FastText |
| $R_{Cent}$ | 0.074 | 0.096 | 0.108 | 0.088 | 0.112 | 0.125 | 0.004 | 0.010 | 0.020 | FastText |
| $R_{Cent} + \hat{\varphi}$ | 0.096 | 0.110 | 0.123 | 0.157 | 0.153 | 0.147 | 0.022 | 0.019 | 0.025 | FastText |
| $R_{Cent} + \hat{\varphi} + IDF$ | 0.101 | 0.120 | 0.124 | **0.161** | 0.160 | 0.152 | 0.021 | **0.025** | 0.026 | FastText |

## 3.7 Discussion

Over all performance of both metrics, corpus-based and embedding-based, is shown in Figure 3.3. The figure combines the results already shown in Table 3.4 and Table 3.5, and reports the results averaged for topics with 5, 10 and 20 words. In most cases, the embedding-based ranks improved over the original topics that were ranked using $R_{Orig}$. However, they rarely outperformed the corpus-based metrics $R_{TFIDF}$ and $R_{IDF}$.

Rankings that used FastText showed stable performances across the three cardinalities which indicated that the metric brought the most useful words to the start of the topic. The metrics that used Word2vec and Glove fluctuated between the cardinalities and showed fewer stable performances.

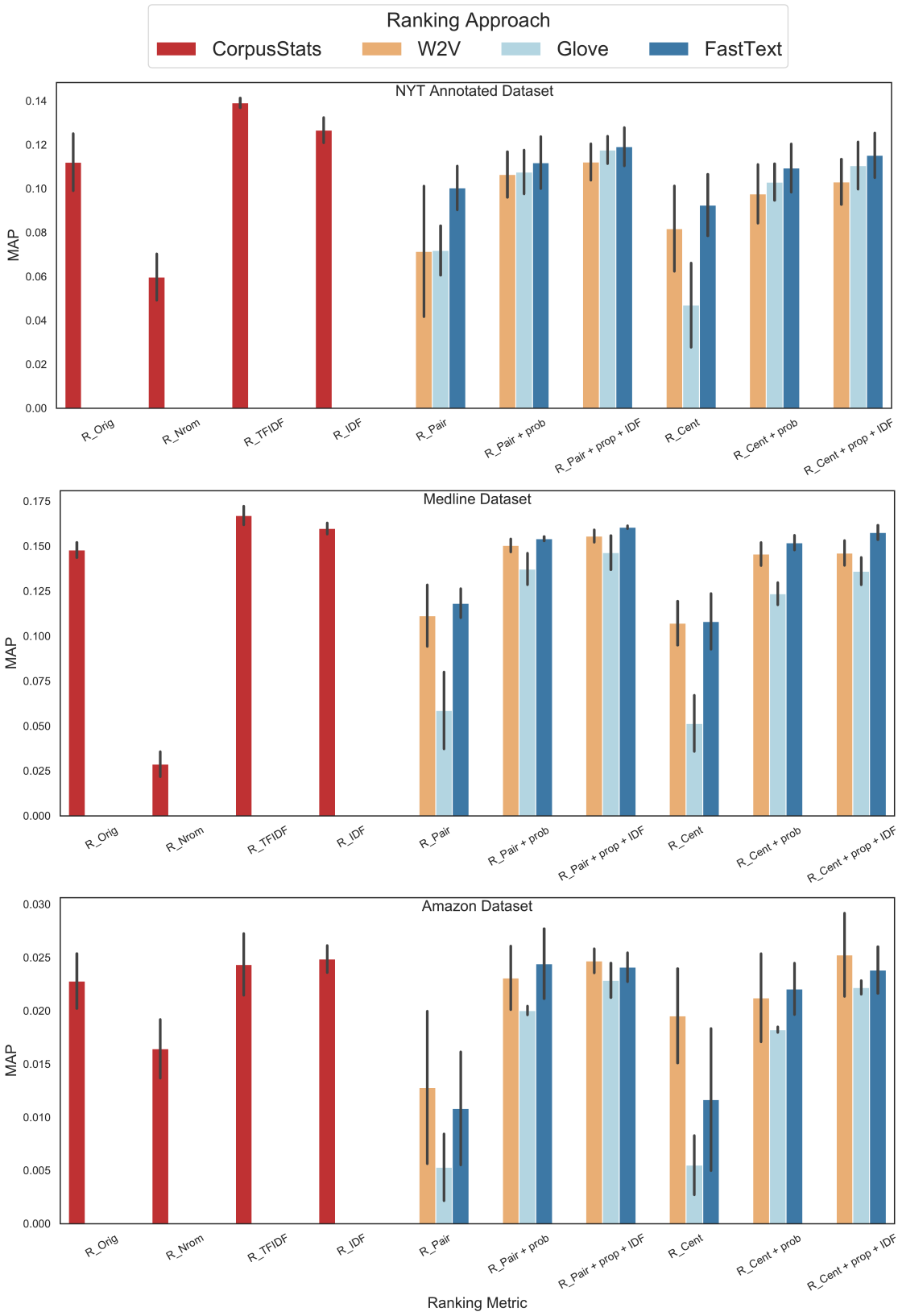



Figure 3.3: The MAP scores for the IR system when given different queries. The MAP scores were averaged for topics with 5, 10 and 20 words. The black bar represents the standard deviation.

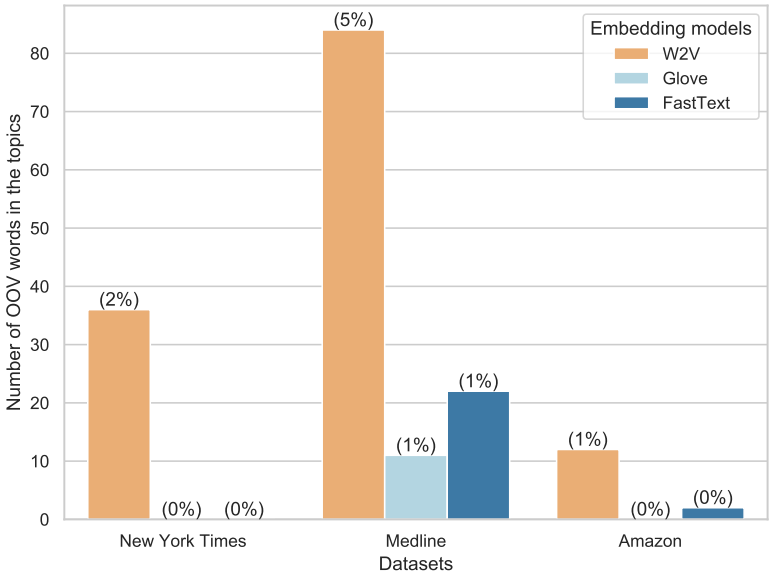



Figure 3.4: The total number of out-of-vocabulary (OOV) words in the topics for each of the embedding models under the three datasets. The percentage of the OOV words to the total number of topics' words' are shown above each bar.

Embedding-based metrics have not performed as anticipated. The reason for this is not clear, but it may be related to the problem of OOV. Therefore, the further analysis of the availability of the topic words in the embedding models is presented in Figure 3.4. The scores shown in the figure are the total number of OOV words found within the topics' top 50 words. Although, the Glove embedding model contained most of the vocabulary, the performance of its vectors in the ranking metric was less useful than those given by Word2vec and FastText. Word2vec showed the highest count of OOV among the embeddings models.

The automatic evaluation approach using IR allowed for the exploration of various ranking metrics and helped to identify metrics with superior performance. A human study at the same scale would not be possible given the resources required for it, such as the time needed to complete the human study and monetary incentives. However, because the aim is to create interpretable topics for humans, ranking metrics should be evaluated using an approach that measures the ranking metric that

produces topics that humans find the most interpretable. Therefore, a human-centred approach is introduced in Chapter 4. Only the metrics presented in section 3.3 will be evaluated further in the next chapter because their performance was superior to those using the embeddings approach described in section 3.4.

## 3.8 Conclusion

This chapter presented a study on word re-ranking methods designed to improve topic interpretability. Ten methods were presented and assessed using an automated evaluation approach based on a document retrieval task. Re-ranking the topic words was found to produce words that were more useful in discriminating documents that described a particular topic from those that did not. The most effective re-ranking schemes were those based on information from the corpus, particularly those that incorporated information about the importance of words, both within topics and their relative frequency in the entire corpus.

The next chapter evaluates the interpretability of the re-ranked topics in a human study. The study consists of a task in which performance indicates the usefulness of the ranking metrics.

# 4

# HUMAN EVALUATION OF TOPIC INTERPRETABILITY

## 4.1 Introduction

The previous chapter explored several re-ranking metrics and compared them using an IR-based approach. This chapter introduces an alternative way of evaluating topic representations. Further analyses are conducted of the best metrics derived in the previous chapter using an approach based on a crowdsourcing task. This study focuses on evaluating corpus-based re-ranking metrics. It does not include the embedding-based methods because they did not perform well in the previous chapter; moreover, human evaluation is time-consuming and expensive. This chapter also explored the effects of different topic visual representations, such as word list and word cloud, which are also ranked using various methods.

This thesis aims to improve the interpretability of topics by humans. Therefore, it evaluates topic representations that can be addressed by performing a human study. Previous works have evaluated topics using various approaches, including a

human study, to estimate the coherence of topics (Chang et al., 2009) and evaluate various visual representations (Aletras et al., 2017, Castella and Sutton, 2014, Smith et al., 2017). The study conducted in this chapter is formulated as a task that incorporates topic representation. The participants' success in performing the task indicates the representation's usefulness compared with other representations. The study was performed through crowdsourcing the task, which allowed for reaching a large number of participants and the fast completion of the task.

Various ways of representation have been presented in section 2.3.1, each of which aided users in interpreting topics to a certain degree. Choosing a representation (or a set of representations) depends strongly on the application and the potential targeted users of the application. This study compares the standard representation of a topic, as a list of the top $n$ terms, with representing the topic as a word cloud. Word clouds are appealing, and they are usually preferred in a wide range of visualisations (e.g., websites, leaflets, and brochures). Therefore, the human study presented in this chapter compares the effectiveness of different topic representations (i.e., word re-rankings) and visual representations (i.e., a list and word cloud) by asking humans to choose the correct topic for a given document. The hypothesis is that humans are able to find the correct topic more easily when the representation is more interpretable. Figure 4.1 shows a topic with the various representations that are explored in this study. The exceptions are different representations through re-ranking, which are not shown in this figure.

The rest of this chapter is organised as follows: the crowdsourcing task that was used to evaluate the topic representations is described in section 4.2, followed by the results of the study in section 4.3. Section 4.4, includes a discussion of evaluating topic representations. The conclusion of this chapter is provided in section 4.5.

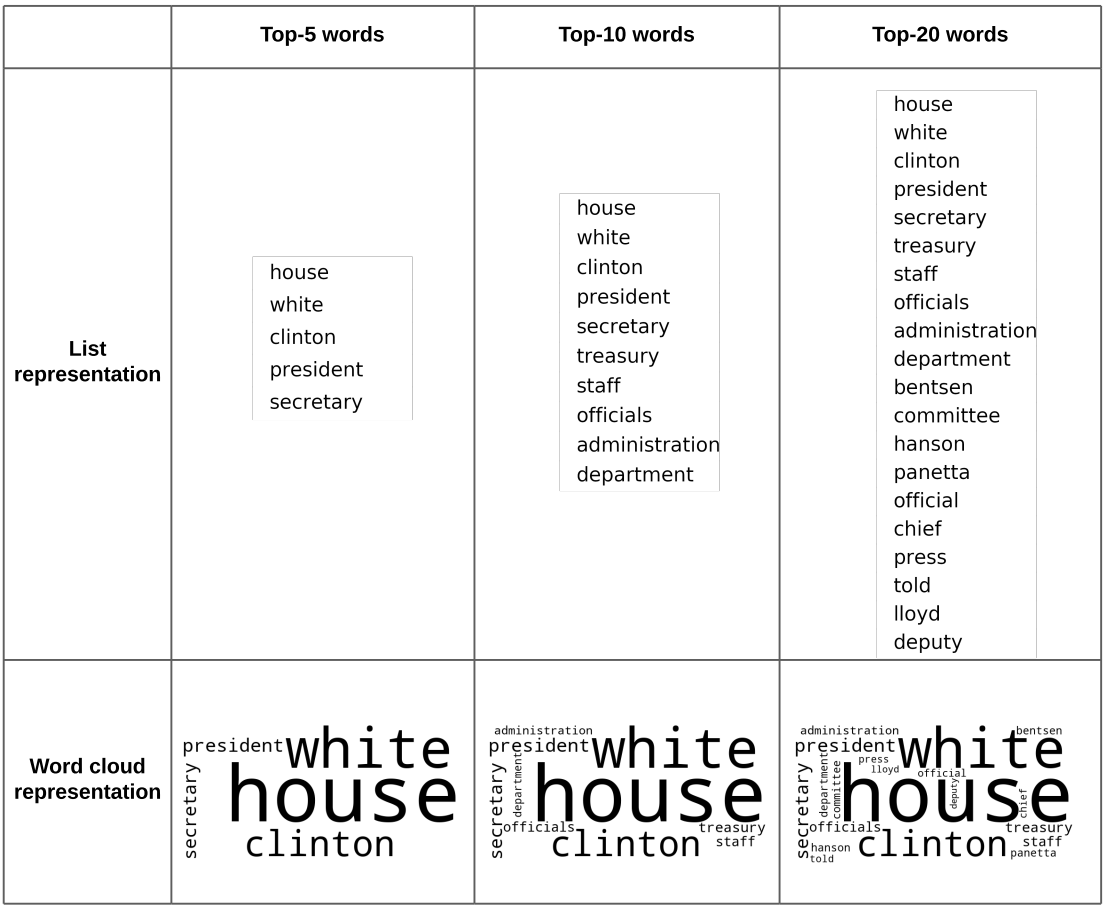



Figure 4.1: A topic represented in various ways by changing the number of words and the visual form of representation.

## 4.2 Crowdsourcing Task

A job was created on the Amazon Mechanical Turk (MTurk) crowdsourcing platform. The participants were presented with a micro-tasks consisting of a brief task overview and a text followed by six topics that were represented by either a list or a word cloud containing the topic's top $n$ words, which were selected using one of the re-ranking methods presented in section 3.3. The participants were asked to select the topic that was the most closely associated with the text. Figure 4.2 shows an example of the micro-task presented to the participants, in which they were asked to choose one of the topics.

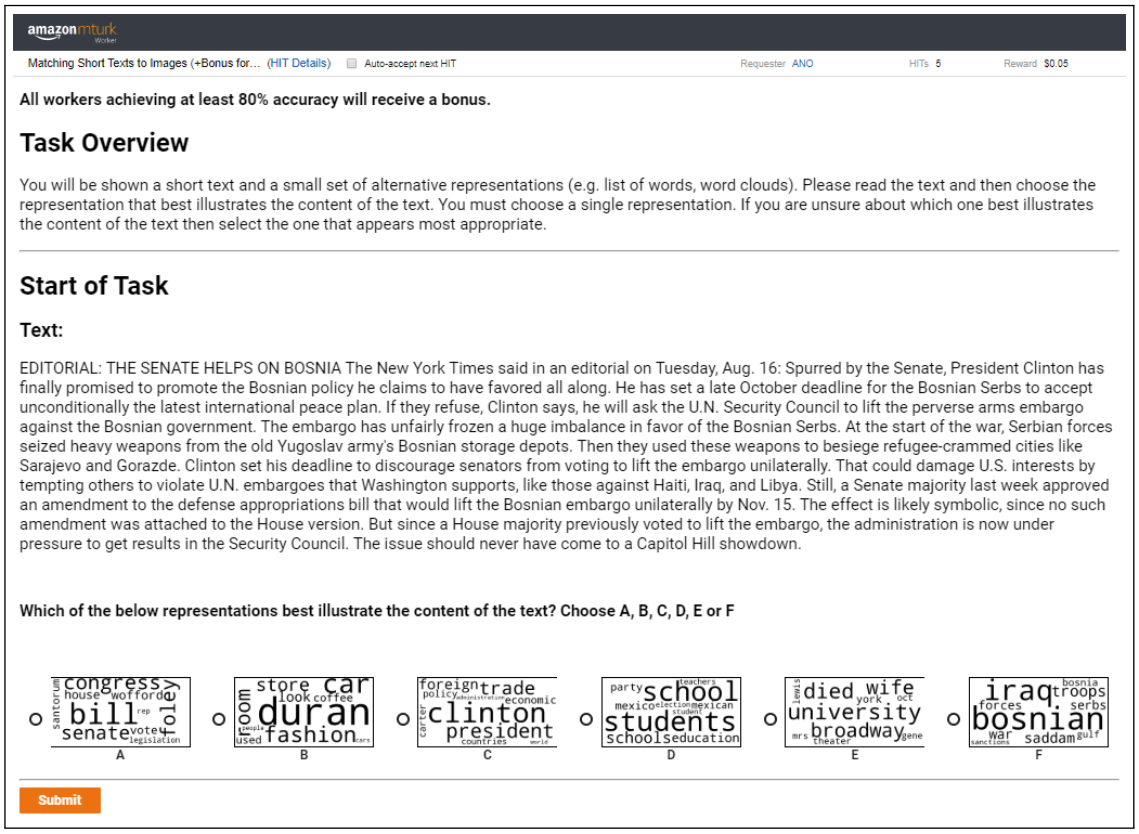



Figure 4.2: Example of the crowdsourcing micro-task interface.

Micro-tasks were created using 48 New York Times articles that were extracted randomly and satisfied the following criteria:

- The articles should be at least 25 words long, which produced documents that were long enough for the workers to comprehend.
- The article should have a topic that has at probability of at least 0.8.
- The distractor topics should have a low probability of less than 0.3.

The correct answer was the topic with the highest probability given the text and the distractor topics that had low probability. The probability of the correct topic was at least 0.8, and the probability of the five distractor topics was lower than 0.3. The distractor topics were picked in order, with the highest probability topic has a probability of less than 0.3. The topics after extraction are randomly ordered for each of the micro-tasks.[1] Micro-tasks were created for each article using

[1] Various values for these parameters were explored but it was found that lowering the probability of the correct answer and/or raising the probability of the distractors made the task too difficult.

each of the four ranking methods (section 3.3) that were generated by topics created using three cardinalities (5, 10, and 20 words) and were represented by two visual representations (i.e., list and word cloud). Five assessments were obtained for each micro-task; consequently, 120 judgements were obtained for each article.[2]

The recruiting approach was "low payment and bonus", which is usually used to encourage workers to submit quality work. This approach offers low payments, but workers who maintain a high accuracy will gain bonus payments. Therefore, \$0.05 was paid per micro-task, and a bonus of \$0.05 was paid for each micro-task submitted with an accuracy of 80%.

### 4.2.1 Dataset and Pre-processing

Approximately 33,000 news articles were randomly sampled from the New York Times included in the fifth edition of the English GigaWord corpus[3]. The same pre-processing steps used in Chapter 3 were applied to the dataset. Articles were tokenised and stop words were removed. Rare and common words were removed by filtering words that occurred in fewer than five or more than half of the articles. The size of the resulting vocabulary was approximately 52,000 words.

### 4.2.2 Topic Generation

Topics were generated using LDA's implementation in Gensim[4] fitted with online variational Bayes (Hoffman et al., 2010). The most important tuning parameter in LDA models is the number of topics. This parameter was set to 50 after experimenting with a varying number of topics and manually examining the resulting topics combined with a coherence analysis. To assess the quality of the resulting LDA models, topic coherence was computed[5] using the following: (1) $C_V$ (Röder et al., 2015); (2) $C_{UCI}$

[2] 4 (ranking methods) × 3 (cardinalities) × 2 (visual representations) × 5 (judgements per article)

[3] https://catalog.ldc.upenn.edu/LDC2011T07

[4] https://radimrehurek.com/gensim

[5] The implementations available in Gensim were used.

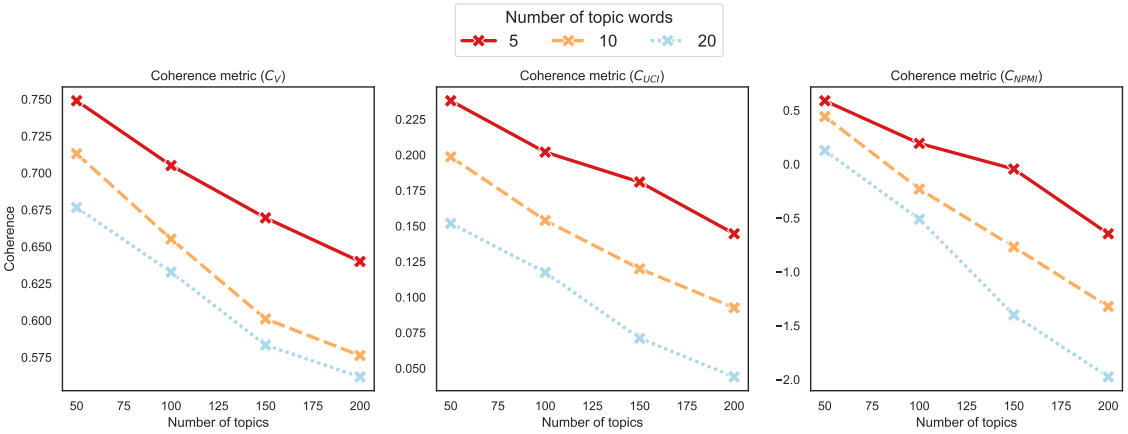



Figure 4.3: Line plot showing the coherence scores for different choices of the number of topics. Coherence was measured via three coherence metrics ($C_V$, $C_{UCI}$, and $C_{NPMI}$). Coherence was degraded when the number of topics was increased.

(Newman et al., 2010); and (3) $C_{NPMI}$ (Bouma, 2009). Figure 4.3 shows the coherence scores for the models using different numbers of topics $K \in \{50, 100, 150, 200\}$. These metrics measure the coherence between the words in a topic by measuring the degree of their semantic similarity. It returns a higher score for topics with words that occur in a similar context within a secondary data source (e.g., Wikipedia). The figure shows that increasing the number of topics produces topics with lower coherence between their words. The topics were also examined manually by inspecting the topics' words to confirm the results provided by the coherence metrics.

### 4.2.3 Quality Measurements of Crowdsourcing Tasks

This section describes the constraints that were used to address the risk of "bots" when software was used to perform the task (Mason and Suri, 2012) or when "cheater workers" found a way to finish the task quickly. MTurk provides system qualifications that are based on the workers' history and are *task-independent*. MTurk also allows for customised qualifications that are designed by the requester and can be *task-dependent*.

Requesters can set up qualifications that only allow workers who pass those qualifications to work on the task. Below are the qualifications that were used, which

were categorised into qualifications based on the worker's information or history on MTurk (**A** and **B**) and qualifications that required action by the user (**C** and **D**).

**A. Location constraint** Because the dataset used in this study was in English, this constraint allowed the task to be performed by workers in countries where English is the native language: Australia, Canada, Ireland, New Zealand, the United Kingdom, and the United States.

**B. Performance history constraint** This constraint controlled access to the task by experienced workers. For example, workers with micro-task approval rates above 90% and had more than 100 approved micro-tasks.[6]

**C. Worker qualification questions** This is a customised qualification, and each worker has to pass a qualification test before proceeding to perform the tasks. Each test contained five questions that were created by randomly extracting text that satisfied the same constraints presented in section 4.2. Workers were granted this qualification if they answered four of five questions correctly. This qualification test ensured that workers were familiar with the task before starting it, thereby eliminating random answers, which increased the reliability of the study (Kazai, 2011).

**D. Consent constraint** Workers were provided with a detailed description of the task as well as an information sheet about the study. They were expected to read the provided information and then click "I agree" to give their consent, which qualified them and allowed them to proceed to the task. A screenshot of the consent page is shown in Figure 4.4.

A total of 477 participants were recruited through MTurk. The participants could only participate once in any configuration to avoid being exposed to the same articles and topics and memorising the answers. Hence, each configuration was set up as a standalone job, and only one job was run at one time. Following the completion of a

[6]Submitted and approved micro-tasks are different, submitted micro-tasks become approved by the requester after further examination of quality.

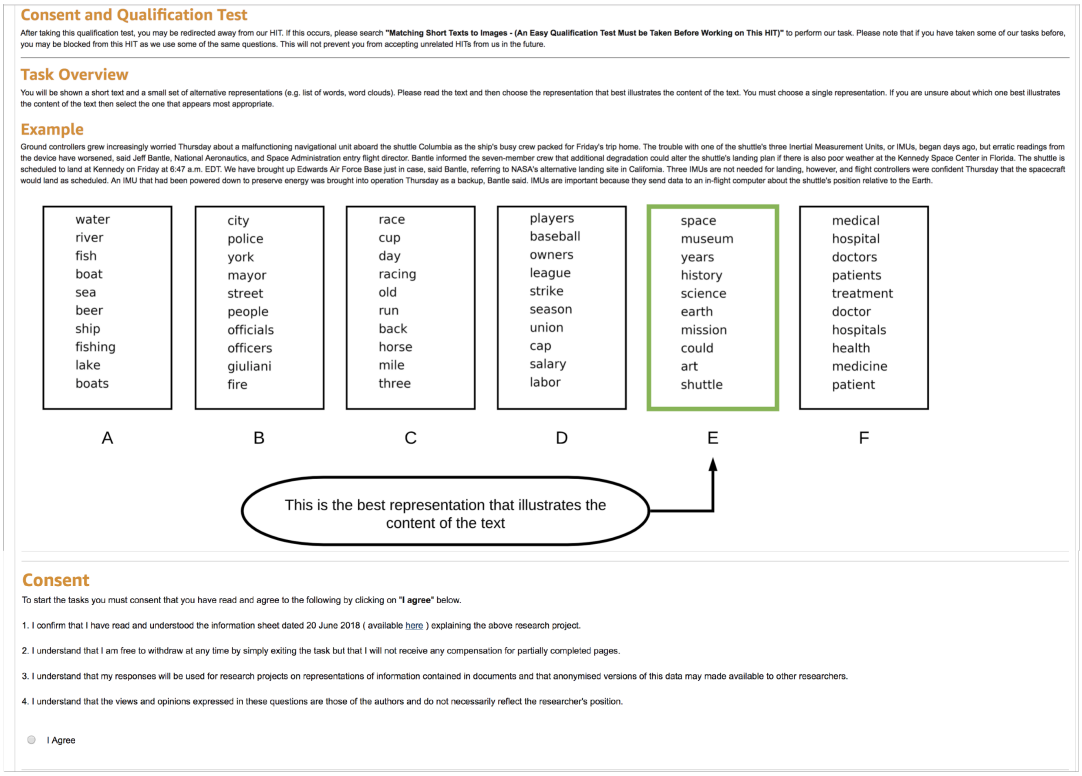



Figure 4.4: Consent page for the study on MTurk.

job, the participants were eliminated from future jobs, which removed the potential for redundant participants and memorised answers.

This crowdsourcing study received ethical approval from the ethics committee of the University of Sheffield. Appendix A includes the ethics application form and the information sheet provided to the participants.

## 4.3 Results and Discussion

### 4.3.1 Performance Accuracy

The results for $R_{Orig}$, $R_{Norm}$, $R_{TFIDF}$ and $R_{IDF}$ when topics were represented by the 5, 10 and 20 highest scoring words are shown in Table 4.1. *Accuracy* represented the percentage of questions in which participants were able to identify the correct topic (i.e., topics with the highest probability given the article). *Time/task* was the

Table 4.1: Results of the experiment comparing re-ranking methods using crowdsourcing (section 4.2). Topics are represented by with their top 5, 10 or 20 probable words. The results are for both topics are represented as lists and word clouds. **Bold** font denotes best values among the ranking metrics.

| #words | | Ranking Methods | | | |
|---|---|---|---|---|---|
| | | $R_{Orig}$ | $R_{Norm}$ | $R_{TFIDF}$ | $R_{IDF}$ |
| 5 | Accuracy (%) | 68.125 | 67.5 | 72.083 | **72.708** |
| | Time/task (Minutes) | 1.141 | 0.815 | 0.831 | **0.715** |
| | Coherence (NPMI) | 0.092 | 0.035 | **0.112** | 0.100 |
| 10 | Accuracy (%) | 70.208 | 63.125 | **76.458** | 75.208 |
| | Time/task (Minutes) | 0.988 | 1.03 | **0.79** | 0.813 |
| | Coherence (NPMI) | 0.072 | 0.038 | **0.091** | 0.084 |
| 20 | Accuracy (%) | 72.5 | 65.625 | **78.958** | 75.417 |
| | Time/task (Minutes) | 0.915 | 1.055 | **0.845** | 1.683 |
| | Coherence (NPMI) | 0.050 | 0.029 | **0.071** | 0.062 |

mean amount of time required for the participants to complete a single task (Figure 4.2). *Coherence* was the average coherence of the topics, which was computed using NPMI (Aletras and Stevenson, 2013b)[7].

The results showed variations in performance, which indicated that re-ranking the topics' words affected the individual's ability to interpret the topics. When the words were ranked using $R_{TFIDF}$ and $R_{IDF}$ the default ranking ($R_{Orig}$) was outperformed. When the words were ranked using $R_{Norm}$, the performance was considerably lower than in words re-ranked using the other methods, in terms of both accuracy and the amount of time taken to complete the task.

These results showed that the improvement obtained by using $R_{TFIDF}$ and $R_{IDF}$ was consistent when the number of words in the representation was varied. The results of using $R_{Orig}$ improved as the number of words increased but did not demonstrate the same performance as the re-ranking methods (except $R_{Norm}$), even when 20 words were included. These results demonstrated that choosing the most appropriate words to represent a topic is more useful than simply increasing the number of words

[7]The implementation provided in https://github.com/jhlau/topic_interpretability was used.

shown to the user. In fact, increasing the number of words shown appears to have slowed the time taken for a user to interpret the topic. The same increase in task completion time was not observed in $R_{Orig}$.

The $R_{TFIDF}$ and $R_{IDF}$ approaches combined information about the word's importance within an individual topic and across the entire document collection, which resulted in more effective rankings than the $R_{Orig}$ approach achieved. In contrast, $R_{Norm}$ considered only the relative importance of a word across all topics. Hence, it would be possible for a word with a relatively low probability based on the topic to be ranked highly if that word also had low probability across all the other topics.

The results were in contrast to those reported by Song et al. (2009), which concluded that $R_{Norm}$ was more effective in word re-ranking than $R_{Orig}$ and $R_{TFIDF}$ (see section 2.4.1). However, in their evaluation methodology, a single annotator was asked per-task to judge whether the words included within topic representations were important or not. The crowdsourcing task presented in this chapter measured the participants ability to directly interpret topic representations using multiple annotations. The low results for $R_{Norm}$ suggested that the crowdworkers were simply unable to interpret many of the topics and, in these cases, their judgements about which words were important were not likely to be reliable.

Overall $R_{TFIDF}$ appeared to be the most effective among the re-ranking approaches evaluated. This method achieved the best performance with 10 and 20 words, although not as well as $R_{IDF}$ for 5 words.

Table 4.2 shows the results for $R_{Orig}$, $R_{Norm}$, $R_{TFIDF}$ and $R_{IDF}$ when the topics were visually represented by either lists or word clouds. Table 4.2 clearly shows that the crowdworkers were more successful in answering the questions when the topics were in a list representation compared with when they were represented by word clouds. Regarding the time taken to complete the task, surprisingly, the crowdworkers who were shown questions in which topics were represented by word

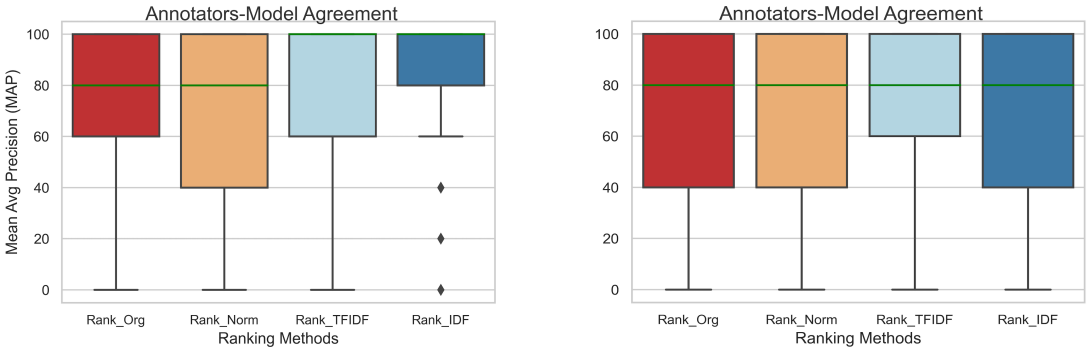
Table 4.2: Comparison of re-ranking methods using the crowdsourcing task (section 4.2). Topics are represented with their top 5, 10 or 20 probable words in lists or word clouds. The results were averaged across three cardinalities. **Bold** font denotes the best value among the ranking metrics, and underlines denote the best score for the list and word cloud.

| Representation | | Ranking Methods | | | |
|---|---|---|---|---|---|
| | | $R_{Orig}$ | $R_{Norm}$ | $R_{TFIDF}$ | $R_{IDF}$ |
| List | Accuracy (%) | 71.94 | 66.67 | 77.5 | **79.03** |
| | Time/task (Minutes) | 1.06 | **0.96** | 1.01 | 1.24 |
| Word cloud | Accuracy (%) | 68.61 | 64.17 | **74.17** | 69.86 |
| | Time/task (Minutes) | 1.03 | 0.92 | **0.73** | 0.86 |

clouds submitted their answers in less time than those who received the topics in the list representation, regardless of the accuracy. The ranking methods maintained the same behaviour presented previously when crowdworkers who were shown the re-ranked topics using $R_{TFIDF}$ and $R_{IDF}$ performed the task more successfully than others did.

### 4.3.2 Inter-annotator and Annotator-model Agreement

Figure 4.5 shows the distribution of data based on the agreement between annotators (i.e., the crowdworkers) and the model (i.e., accuracy). The box plot shows how tightly the data were grouped and indicates any outliers. In Figure 4.5, the boxes in the figure represent the upper quartile (3rd quartile) to the lower quartile (2nd quartile) of the correct choice of topic. The green line in the middle of the box represents the median of the data. The whiskers represent the start and end of the data range, and the diamond shapes stand for outliers data points. Figure 4.5a shows the agreement between the crowdworkers and the model when the topics were represented by lists. The crowdworkers agreed the least with the model when they were shown topics that were re-ranked using $R_{Norm}$ and on the contrast when they where shown topics that were re-ranked using $R_{IDF}$. The crowdworkers tended to agree with the model when the $R_{TFIDF}$ ranking method was used compared with the




(a) List Representation (b) Word cloud Representation

Figure 4.5: Box plot showing human agreement with the model in each of the ranking method when the topics were represented by lists of words or word clouds.

Table 4.3: Agreement between workers and the model computed using Krippendorff's alpha.

| **Ranking Methods** | **Agreement Alpha** | |
|---|---|---|
| | **List** | **Word Cloud** |
| $R_{Orig}$ | **0.669** | 0.661 |
| $R_{Norm}$ | **0.553** | 0.468 |
| $R_{TFIDF}$ | **0.735** | 0.692 |
| $R_{IDF}$ | **0.743** | 0.661 |
| Overall | **0.675** | 0.62 |

default ranking $R_{Org}$. Figure 4.5b shows the topics presented to the crowdworkers using word clouds. The crowdworkers had lower agreement with the model in all ranking methods compared with the topics represented in lists, except $R_{Norm}$ where the same agreement level was maintained.

Furthermore, the agreement among crowdworkers was computed using Krippendorff's alpha (Krippendorff, 2011). Table 4.3 reports the Krippendorff's alpha scores computed for the two visual representations. The results showed that the crowdworkers had higher inter-agreements in all rankings in a list representation than when topics were represented by word clouds. The ranking methods also maintained the same behaviour described in section 4.3.1. $R_{IDF}$ showed the highest agreement among the crowdworkers.

## 4.4 Evaluating Topic Representations

This chapter presented a method for evaluating topic representations using a crowdsourcing experiment that relied on human judgements. The previous chapter (Chapter 3), presented an automated evaluation of topics based on an IR task. Although there were differences between results of the two methods, the relative performances of the re-ranking methods explored in the two chapters were similar. The correlations between the results of the crowdsourcing experiment and the IR evaluations were statistically significant in all three datasets (Pearson's $r$ varied between 0.82 and 0.89, $p < 0.005$). These results suggest that the automated evaluation approach presented in the previous chapter could be a useful tool for assessing the effectiveness of word re-ranking methods because of the advantage that results are obtained more rapidly than in methods that require human judgements. However, human judgements are recommended when performances are similar. Moreover, automated evaluation should not be relied upon to make fine-grained distinctions between approaches, which is common in some tasks (e.g., Machine Translation (Papineni et al., 2002)).

## 4.5 Conclusion

A study was conducted on word re-ranking methods that were designed to improve topic interpretability. Four methods were assessed through a crowdsourcing experiment in which the participants were asked to associate articles with related topics.

Re-ranking the topic words was found to improve the interpretability of the topics. Therefore, re-ranking should be used as a post-processing step to improve topic representation. The results indicated that a complex visual representation, such as a word cloud, did not improve the interpretability of topics compared with a simple representation in the form of a list.

The next chapter addresses topic interpretability through topic labelling. Topic labelling improves the topic's interpretability by providing descriptive phrases that indicate the topics' subject.

# 5

# A NEURAL APPROACH TO AUTOMATICALLY LABELLING TOPICS

## 5.1 Introduction

The previous chapter presented the details of applying a number of ranking metrics to the topic terms. It has been shown that topics tend to have less than perfect terms within their top $n$ terms such as the most frequent terms in a corpus (Chang et al., 2009, Lau et al., 2010). Re-ranking the topic terms re-evaluates the importance of a term to the topic and allows for the identification of informative words that were ranked lower initially, thereby improving the interpretability of the topics to users.

Even after re-ranking, the user has to read all/part of the topic terms to understand the topic's idea or subject, which relies on the user's interpretation of the topic in addition to the user's knowledge of the terms (Lau et al., 2011). This chapter includes the proposition of assigning a short phrase to the topics automatically. Such

a phrase summarises the ideas of the topic and therefore allows for fast interpretation by the users (Aletras and Mittal, 2017, Aletras and Stevenson, 2014, Aletras et al., 2017). For example, consider a topic with the following terms ⟨*pain, disorder, symptom, depression, anxiety, patient, chronic, depressive, study, psychiatric*⟩. The terms belong to a psychology domain and may be more easily interpreted if they were labelled with ⟨*mental health*⟩. More formally, the topic labelling task is the process of assigning a latent topic, $t$, represented by its top $n$ words, $w_*$, such that $t = \{w_1, w_2, \ldots, w_n\}$ with a phrase , $l$, of length $m$ that is semantically related to the topic and conveys the topic's concept (i.e., a label $l = \{w_1, w_2, \ldots, w_m\}$).

Section 2.4.2 presented previous work in topic labelling, which mainly followed a two-stage approach where: (1) candidate labels are retrieved from a large pool (e.g., Wikipedia article titles); and (2) ranked based on their semantic similarity to the topic terms to identify the most suitable label (Aletras and Stevenson, 2014, Aletras et al., 2017, Bhatia et al., 2016, Lau et al., 2011, Mei et al., 2007). A limitation of these extractive approaches to label generation is that they are restricted to assigning labels that are found within the set of candidates. This chapter presents the use of a neural-based approach that does not suffer from this limitation when generating labels. The model generates labels for topics in one step given the topics' top $n$ terms, instead of retrieving and ranking. The labels are generated using generative models conditioned on the entire source label words. Therefore, labels generated using this approach can include novel words that do not appear in the topics.

The rest of this chapter is divided into four sections. Section 5.2 presents the proposed approach followed by the neural labelling models employed to generate topic labels are detailed in sections 5.3 and 5.4. Finally, a summary of the chapter is presented in section 5.5.

## 5.2 Proposed Approach

The proposed approach is based on a sequence-to-sequence model (Seq2Seq) (Cho et al., 2014, Sutskever Google et al., 2014) that takes a sequence of words as input and generates another sequence of words as output. For example, a model can take a sequence of words in French and generates its translation as a sequence of words in English.

Seq2Seq models consist of two neural networks, one of which acts as an encoder and the other as a decoder. In general, the encoder takes as input a sequence of values $x = (x_1, \ldots, x_n)$ and transforms them into hidden representations $z = (z_1, ..., z_n)$ which are passed to the decoder. The decoder generates the output one symbol at a time with each symbol generated being conditioned by the hidden state and the symbols generated previously, i.e., symbol $y_t$ is predicted as $P\left(y_i | \left\{y_1, \cdots, y_{i-1}\right\}, x\right)$.

Various neural networks including convolutional, recurrent and attention-based networks are used to create the labelling model. The neural labelling models are described next in section 5.3.

At the start of this chapter, in section 5.1, it has been stated how the proposed labelling method follows a different approach than those previously proposed, most of which retrieve candidate labels and rank them to find the most appropriate one semantically. Besides, the proposed approach differs from previous neural-based approaches. Aletras et al. (2017) used neural networks to estimate the relevance of a given topic and an image label. Similarly, Sorodoc et al. (2017) proposed an approach that predicts an appropriateness score between a topic and a textual label.

## 5.3 Topic Labellers

### 5.3.1 Convolutional-based Model

As shown in section 2.5.1, CNNs are networks that use convolution, a linear operation between the input and a filter (i.e., kernel) to produce a feature map for the input which is followed by a max-pooling operation to reduce the size of the feature map. A model derived from (Gehring et al., 2017) is used where a multi-layer CNN learns hierarchical representations over the input sequence. The model architecture is shown in in Figure 5.1, it consists of an encoder and a decoder. At the encoder network, a topic sequence with n terms $x = (x_1, \ldots, x_n)$ is passed through an embedding layer where the sequence is embedded in distributional space as $w = (w_1, \ldots, w_n)$, where $w_j \in \mathbb{R}^d$ is a column in an embedding matrix $\mathcal{M} \in \mathbb{R}^{V \times d}$. The sequence is also combined with a positional embedding which encodes the sequence's order as $p = (p_1, \ldots, p_n)$, where $p_j \in \mathbb{R}^d$. $w$ and $p$ are combined by element-wise sum to obtain the input embedding representation with information about the token and its position in the sequence $e_{enc} = (w_1 + p_1, \ldots, w_n + p_n)$. Then, $e_{enc}$ is passed through a linear layer and a number of convolutional blocks where the output of the $l^{th}$ block is denoted as ${Z_{enc}}^l = \left(z^l_{enc,1}, \ldots, z^l_{enc,n}\right)$. This chapter's embeddings were learned with the model, unlike in section 3.4.1, where pre-trained embeddings were adopted.

Each convolutional block contains one convolutional layer with a single filter sliding over the input embeddings within a sequence. The size of the resulting hidden representation is twice the size of the input because a spacial activation function called gated linear units (GLU (Dauphin et al., 2016)) is used which uses a gating mechanism similar to those used in LSTM and GRU that controls which parts of the convolution outputs to keep. After the GLU and before passing the block's output to the next block, a residual connections process is performed from the input of each convolutional block with the output of the same block (He et al., 2016).

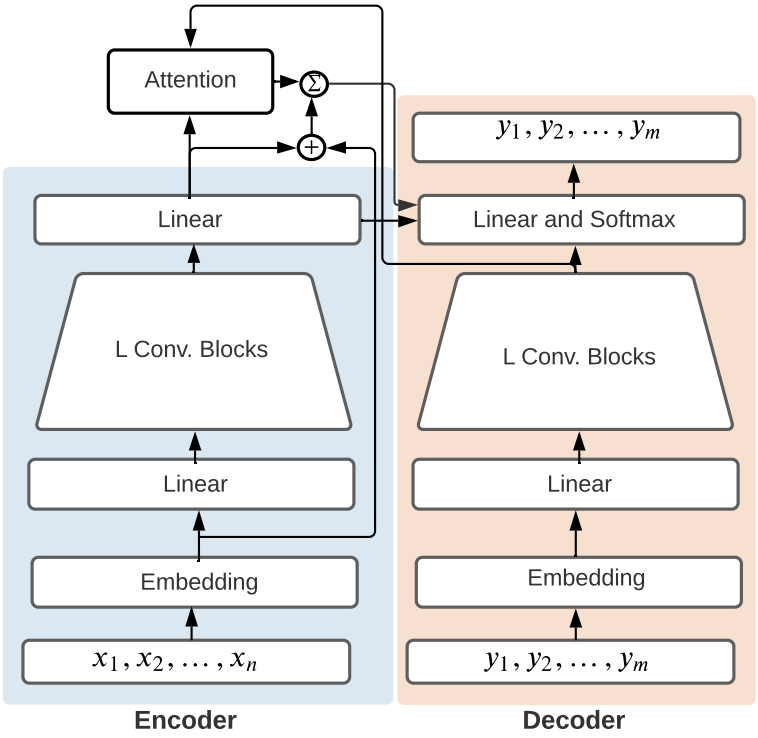



Figure 5.1: Seq2Seq topic labelling model using CNNs.

At the decoder network, the label words are passed as inputs which are processed in a similar way to that described for the encoder's inputs. The decoder's inputs are transformed into an embedding and combined with positional information giving $e_{dec} = (y_1 + p_1, \ldots, y_m + p_m)$. Padding is essential to make sure that the decoder does not observe future information. The input is padded by $filter\ size - 1$ elements before being passes to the convolutional blocks where the output of the $l^{th}$ block at the decoder is denoted as ${Z_{dec}}^l = \left(z^l_{dec,1}, \ldots, z^l_{dec,m}\right)$.

An attention mechanism (Sukhbaatar et al., 2015) is used to allow the decoder to learn which parts of the encoder influences the prediction at each step. The attention context vector $c^l_i$ at step $i$ for decoder layer $l$ is computed as follows. First, the decoder's state summary $d^l_i$ is computed given the current decoder state ${z_{dec,i}}^l$ and the embedding of the previous target word $e_{dec,i}$,

$$d^l_i = W^l_d {z_{dec,i}}^l + b^l_d + e_{dec,i} \tag{5.1}$$

The attention $a_{ij}^l$ of state $i$ and source element $j$ in layer $l$ is computed as the dot-product between the resulting $d_i^l$ with each output of the last encoder layer ${Z_{enc}}^u$ as shown in Eq. 5.2. The conditional input $c_i^l$ for the current decoder layer is computed using Eq. 5.3 as a weighted sum between the encoder outputs and inputs. Adding the inputs again to the encoder outputs have been found to be helpful since considering the encoder outputs $z_{enc}$ represent input contexts, and the inputs $e_{enc}$ provides point information about a specific input element (Gehring et al., 2017).

$$a_{ij}^l = \frac{\exp\left(d_i^l \cdot {z_{enc,j}}^u\right)}{\sum_{i=1}^{n} \exp\left(d_i^l \cdot {z_{enc,i}}^u\right)} \tag{5.2}$$

$$c_i^l = \sum_{j=1}^{n} a_{ij}^l \left({z_{enc,j}}^u + e_{enc,j}\right) \tag{5.3}$$

Finally, $Z_{dec,i}^l, c_i^l, W_o$ and $b_o$ are passed through an FC with softmax activation function to compute the distribution over the $V$ possible target words to identify the next word as follows:

$$p\left(y_{i+1} \mid y_1, \ldots, y_i, x\right) = \text{softmax}\left(W_o(Z_{dec,i}^l + c_i^l) + b_o\right) \tag{5.4}$$

### 5.3.2 Recurrent Models

The recurrent-based labelling model consists of a bidirectional RNN encoder and an RNN decoder (two RNN labelling models were created: one with GRU and the other one with an LSTM. The two models are separate and denoted as **BiGRU** and **BiLSTM**, respectively). An attention mechanism (Bahdanau et al., 2015) was used to allow the decoder to pay specific attention to parts of the encoder states at each decoding step. In the proposed approach the encoder takes the topic terms as input and it passes them to an embedding layer that maps them into a low-dimensional embedding followed by a bidirectional RNN. The forward RNN reads the input in its original order $(x_1, \ldots, x_n)$, whereas the backward RNN reads it in the reverse order $(x_n, \ldots, x_1)$, thereby encoding information from the preceding and following

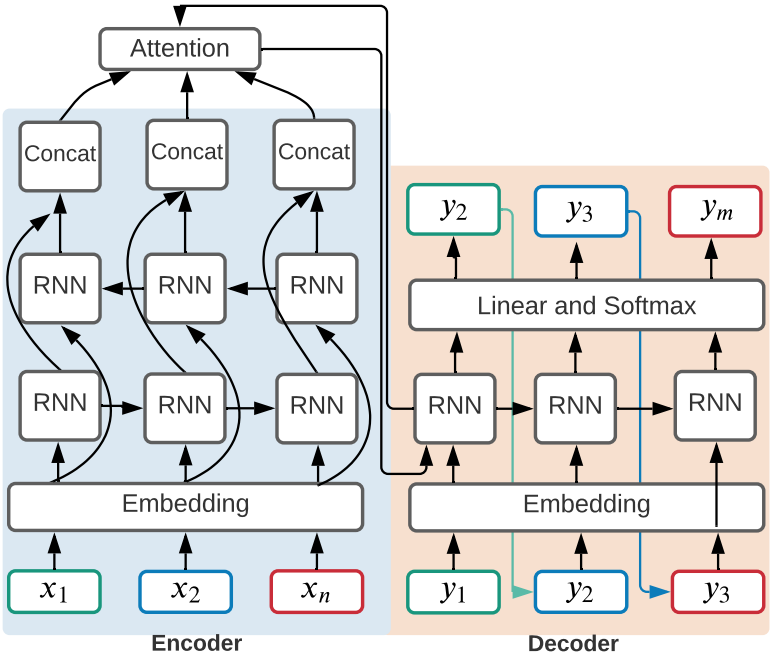



Figure 5.2: Seq2Seq topic labelling model using RNNs.

words. The RNN's forward output at step $i$, $zf_i$, and backward output, $zb_i$, are concatenated giving the hidden state $z_{enc,i}$ of $x_i$.

$$\begin{aligned} zf_i &= \text{RNN}(x_i, z_{i-1}) \\ zb_i &= \text{RNN}(x_i, z_{i-1}) \\ z_{enc,i} &= [zf_t; zb_i] \end{aligned} \tag{5.5}$$

During decoding, labels are predicted word by word. At timestep $i$, the decoder computes the hidden state $z_{dec,i}$ as follows

$$z_{dec,i} = \text{RNN}\left(y_{i-1}, z_{dec,i-1}, c_i\right) \tag{5.6}$$

where $y_{i-1}$ is the previous prediction that gets fed back to predict the next word and $z_{dec,i-1}$ is the previous hidden state. Notice here that $c_i$ is a context vector computed for each target word. This approach is different from traditional encoder-decoder architectures were the last hidden state of the encoder is used to compute $C$, a

context vector which is used by the decoder at every time step. The context vector $c_i$ is computed as the weighted sum over all encoder hidden states and weights $\alpha_i$ using an attention mechanism (Bahdanau et al., 2015):

$$\begin{aligned} d_{ij} &= a\left(z_{dec,i-1}, z_{enc,j}\right) \\ \alpha_{ij} &= \frac{\exp\left(d_{ij}\right)}{\sum_{k=1}^{n} \exp\left(d_{ik}\right)} \\ c_i &= \sum_{j=1}^{n} \alpha_{ij} z_{enc,j} \end{aligned} \tag{5.7}$$

where $a$ is a FC learned with the rest of the model. The weights $\alpha_i$ sum to 1 and give higher weight to a specific state, which allows the decoder to focus on this state among others.

The decoder's hidden state $z_{dec,i}$ is used to generate the output probability over all possible vocabulary items for the labels by passing it to a FC with a softmax activation function (similar to Eq. 5.4 ). Finally, the probability distribution resulting from the previous step is used to choose the word with the highest probability as the prediction $y_i$,

$$y_i = argmax\big(p(y_i|\{y_1, \cdots, y_{i-1}\}, x)\big) \tag{5.8}$$

Figure 5.2 shows the high level architecture for the RNN-based labelling model.

### 5.3.3 Combined model

It has been shown in section 2.5.1 that CNNs extract features from data and therefore they have been used as a feature extraction mechanism within another model where the feature maps they produce are passed as input to another model. For example, an image captioning model that takes an image as an input and extracts its feature map, which is then used to initialise an RNN at the decoder side that generates the image caption one word at a time (Vinyals et al., 2015). Given that the input in the topic labelling task is a list of terms that ignores positional information, a CNN

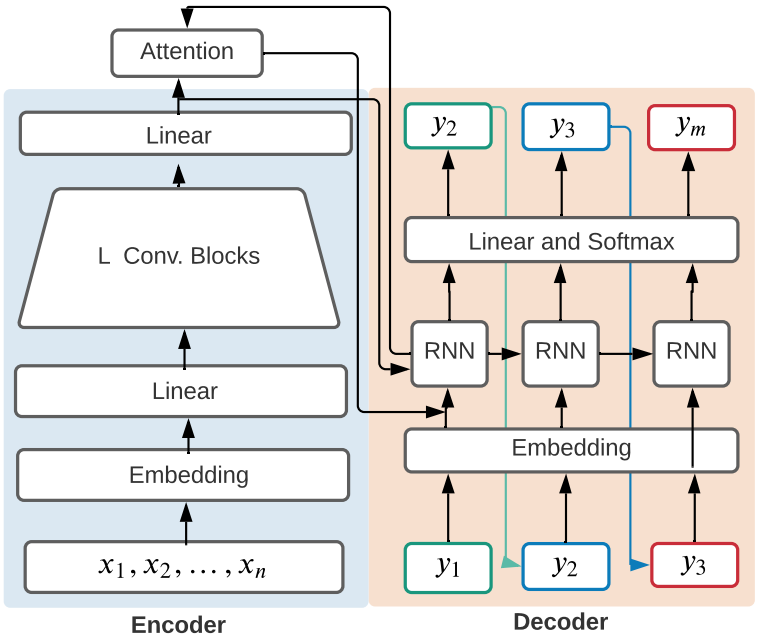



Figure 5.3: Seq2Seq topic labelling model using a CNN-based encoder and an RNN-based decoder.

encoder is used. Since the output is a phrase where word order is important, an RNN decoder is used. As in section 5.3.2, two models are created, one using a GRU (**CNN-GRU**) and one using an LSTM (**CNN-LSTM**).

The encoder model in this architecture is the one proposed in section 5.3.1. The output of the encoder contains the features extracted from the input sequence which is passed in turn to the decoder. The decoder side is a recurrent network that takes as input: the encoder's feature map, the decoder's previous state, and the decoder's input. The encoder's extracted features are also used to compute attention in a similar approach to the one described in section 5.3.2. A high level overview of the model is shown in Figure 5.3.

### 5.3.4 Transformer-based model

As shown in section 2.5.3, a Transformer block is composed of linear layers, attention mechanisms and normalisation processes. The transformer-based model for generating labels consists of an encoder-decoder architecture similar to those used in earlier

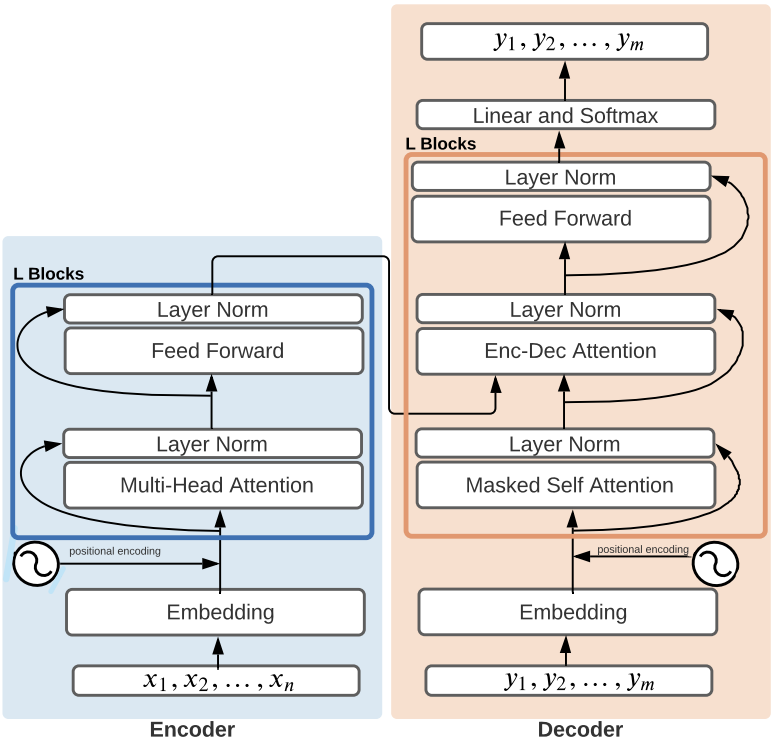



Figure 5.4: Seq2Seq topic labelling model using a transformers architecture.

models. The encoder and decoder include a stack of $L$ transformer blocks each with a self-attention and FC networks. The decoder has an additional layer of multi-head attention over the output of the encoder stack and the output of the masked self-attention layer. An overview of the architecture can be found in Figure 5.4.

First, the encoder receives a sequence $x = (x_1, x_2, \ldots, x_n)$ which is passed through an embedding layer to convert it to vectors of dimension $d$. The embedding vector gets combined with a positional embedding to encode the order in the sequence. The resulting embedding is a combination of tokens and positions $\mathbf{e}_{enc} = (e_1, e_2, \ldots, e_n)$. The combined embedding is passed through $L$ transformer blocks to get the output of the encoder $Z_{enc}$ which in turn gets passed to the decoder.

The decoder has additional components to the encoder: masked multi-head attention, encoder-decoder attention, linear layer, and softmax. A similar process is applied at the decoder where a sequence is passed to an embedding layer and combined with positional information. It must be remembered that the decoder at

position $p$ is not supposed to have access to future information, in other words, the decoder only has access to words from the start of the sequence until $p$ and the rest is masked (i.e., set to zero temporarily).

The masked sequence is passed to the first attention component, the *masked self-attention layer*, which yields the result $Z_{dec}$, then it is passed to the encoder-decoder attention layer with the outputs of the encoder $Z_{enc}$. An FC network receives the attention output and produces the final decoder output.

The predictions in a Transformer are generated in a similar fashion as in the previous neural networks. The final hidden representations (i.e., the output of the last Transformer block in the decoder) are passed through a linear layer to map them to a dimension equal to the number of words in the output vocabulary $V$. Then, a softmax function is used to transform it to a probability distribution and the word associated with the index with highest probability is the prediction $\hat{y}$.

## 5.4 Candidate Search

The labelling model generates a probability distribution across the possible vocabulary and at each time step the decoder chooses from this pool of vocabulary. The final layer in the model is FC layer that uses a softmax activation function which, for each word in the vocabulary, produces the likelihood of it being the next word. The decoding algorithm samples the output from the probability distribution either by: (1) a greedy approach where the word with the highest probability is predicted at each step or (2) a beam search approach where multiple possible predictions are maintained at each time step.

Greedy approaches are fast and effective but not optimal as picking the highest probability word at each time step does not guarantee a final output with the highest probability in total. For greedy decoding, an argmax function is used to select the index with the largest value and thereby select the associated word. The models

in this thesis use a greedy sampling approach and leave beam search sampling for future work.

## 5.5 Conclusion

The start of this chapter included motivation for assigning labels to topics and a formal definition of the topic labelling task. The proposed labelling approach was described in section 5.2 followed by the details of the proposed models for topic labelling in section 5.3 and the candidate search used was stated in section 5.4.

The next chapter describes the approach used to create the training datasets and presents details of the datasets used in testing. It also covers the experimental settings for implementing the proposed models including: hardware specification, hyper parameters, evaluation approaches and evaluation metrics. An analysis of the results is presented, in addition to qualitative content analysis that was used to examine the quality of the generated labels.

# 6

# EVALUATION OF NEURAL APPROACHES

## 6.1 Introduction

In chapter 5, a new approach to generating labels for topics was proposed. The new approach formulates the topic labelling task as a Seq2Seq task where topic terms are provided as input and another set of words (i.e., label) is produced as output. A Seq2Seq model consists of two networks: an encoder and a decoder. A number of ANN variants were proposed as encoders and decoders including: CNN, RNN, and Transformer.

This chapter is organised as follows: the data created and used for training the labelling models are described in section 6.2. Section 6.3 includes the experimental details for implementing the proposed models including hardware/software specifications and the models' hyperparameters. Section 6.4, describes and formulates the evaluation approaches used and evaluation metrics. The results and discussion are included in section 6.5, followed by a summary in section 6.6.

Table 6.1: Sample topics from (Bhatia et al., 2016).

| Domain | Topic terms | Candidate labels | Average rating |
|---|---|---|---|
| Blogs | school, student, university, college, teacher, class, education, learn, high, program | primary education,<br>graduate school,<br>pre-medical | 2.0<br>2.25<br>0.71 |
| Blogs | vote, house, election, poll, bill, republican, party, voter, candidate, senate | general election,<br>state senator,<br>incumbent | 2.43<br>1.75<br>0.66 |

## 6.2 Data

### 6.2.1 Training Data

A set of topics represented by lists of terms and their associated labels are required to train the proposed labelling models. However, the current available datasets are too small to train large neural networks. For example, Bhatia et al. (2016) released a dataset that contains 228 topics with 19 labels for each topic. A sample of the topics with their associated labels and ratings is shown in Table 6.1. Therefore, a *distant supervision* (Craven and Kumlien, 1999) approach was followed which generates training data automatically by using existing knowledge base to extract examples for the labelling task. Using distant supervision approach, two different datasets consisting of pairs of topics and labels were created:

- **ds_wiki_tfidf** was created by selecting pairs of titles and articles from Wikipedia[1]. The article titles are treated as the *labels*, and the top 30 words from each article ranked by TFIDF are treated as synthetic *topic terms*.
- **ds_wiki_sent** is a variation of **ds_wiki_tfidf**. Rather than extracting the top 30 words using TFIDF, the *first* 30 words from the article were used as *topic terms*.

Using this approach, just over 300,000 pairs of topics and labels were collected.

[1]Using the dump enwiki-2019201-pages-articles1.xml-p10p30302

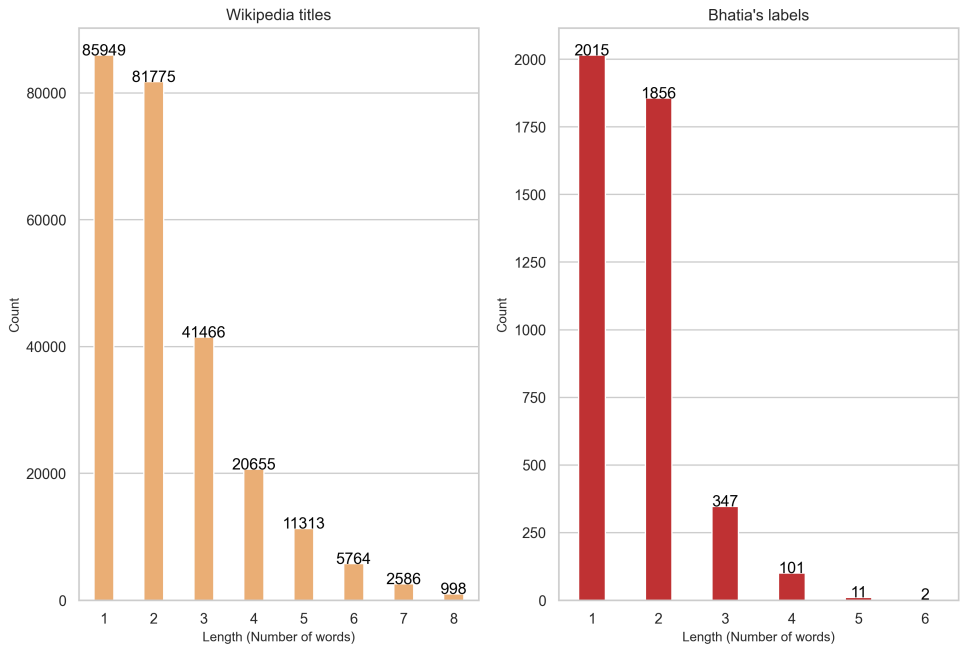



Figure 6.1: Bar-plot showing the number of words distribution in Wikipedia titles and the labels generated by Bhatia et al. (2016).

Standard pre-processing steps were applied to clean and tokenise the datasets including the removal of numbers, special characters, rare terms and stop words[2]. Titles with more than eight words or those that contained duplicate words were filtered. Refer to the left bar-plot in Figure 6.1 that shows the frequency of words in the titles after filtering. As illustrated, most titles consist of one or two words. Pre-processing resulted in over 250,000 pairs of topics and labels. The pairs were divided randomly into three sets: train, validate, and test sets consisting of 226,282, 12,424 and 11,800 pairs, respectively. The article titles (i.e., labels) in both datasets contain 13,947 unique words while the articles' terms contain 181,793 in **ds_wiki_tfidf** and 87,446 in **ds_wiki_sent**. Table 6.2 contains samples from both datasets.

[2]Stop words were not removed from headlines.

### 6.2.2 Test Data

Labels generated by the proposed models were evaluated by comparing them against gold-standard labels from two datasets. The first, described by Bhatia et al. (2016) (**topics_bhatia**), contains 228 topics from four different domains (blogs, books, news and PubMed) that were generated by Lau et al. (2011). Bhatia et al. (2016) associated each topic with 19 candidate labels by matching the topic's top 10 terms with Wikipedia titles using neural embedding. Human ratings for those candidate labels were collected by formulating a crowdsourcing task on MTurk. Annotators (i.e., crowdworkers) gave ratings for the labels between 0 and 3, where 3 is the highest rating. Only labels that received a high average rating (of 2 and above) were used for the dataset, resulting in 219 topics and 1156 pairs (instead of 4332, i.e., 228 topics $\times$ 19 labels). Figure 6.1 shows the length of labels in this dataset.

The second dataset, **topics_tfidf**, is an extended version of **topics_bhatia** that includes 20 additional terms for each topic. These additional terms were added to the 10 terms from **topics_bhatia** so that each topic consists of 30 terms, matching the encoder length. The additional terms were identified by finding documents associated with each topic and choosing the 20 terms with the highest TFIDF scores. Unfortunately the topic-document distributions are not available for **topics_bhatia**. Consequently suitable documents were identified by computing cosine similarity between the topic terms and documents using word embedding. While the lack of information about the topic-document distributions is far from ideal, we chose to use **topics_bhatia** since it provides ratings for labels and these are expensive to obtain. Samples of **topics_bhatia** and **topics_tfidf** are shown in Table 6.2.

Summary of all datasets for training and testing are shown in Table 6.3.

Table 6.2: Samples of topics and labels from the datasets described in section 6.2. Additional terms added to the topic are shown in blue.

| Dataset | Topic terms/Article | Label/Title |
|---|---|---|
| **topics_bhatia** | oil energy gas water power fuel global price plant natural | biofuel |
| **topics_tfidf** | oil energy gas water power fuel global price plant natural lng regasification plants cold gasification turbine exhaust viable floating fluid usage conventional temperature joule acceptability argon utilisation byproducts urea cryogenic | biofuel |
| **ds_wiki_tfidf** | uruguay uruguayan immigration spaniards immigrants amerindians european th argentina backbone italians background society syrian fructuoso countries matanza paraguayans bolivians uruguayans peruvians venezuelans americans colonial multiethnic del amerindian brazil people | immigration to uruguay |
| **ds_wiki_sent** | immigration uruguay started arrival spanish settlers colonial period known banda oriental immigration uruguay similar towards immigration argentina throughout history uruguay known gain massive waves immigration around world specifically european immigration | immigration to uruguay |

Table 6.3: Statistics of the datasets described in section 6.2.

| Dataset | Subset | Size | Vocabulary Size | |
|---|---|---|---|---|
| | | | Input | Output |
| **topics_bhatia** | - | 1,156 | 1,274 | 1,101 |
| **topics_tfidf** | - | 1,156 | 6,872 | 1,101 |
| **ds_wiki_tfidf** | Train<br>Validate<br>Test | 226,282<br>12,424<br>11,800 | 181,793 | 13,947 |
| **ds_wiki_sent** | Train<br>Validate<br>Test | 226,282<br>12,424<br>11,800 | 87,446 | 13,947 |

## 6.3 Experimental Setup

### 6.3.1 Hardware and Software Specifications

The models were implemented with Python and using TensorFlow and PyTorch. Most of the model training was carried out using NVIDIA Tesla P100 GPU, while the inference was mostly done locally using a CPU i7. The models were trained for a maximum of 10 epochs and to avoid overfitting an early stopping option was used to monitor the loss on the validation set.

### 6.3.2 Model Hyperparameters

Model hyperparameters were tuned by randomly sampling combinations including: learning rates, layer sizes, number of layers, and filter sizes for CNNs. The combination that produced the smallest loss was used. The hyperparamters that were tested for each model architecture are shown in Table 6.4.

### 6.3.3 Baselines

The labels generated by the proposed models were compared with two baselines obtained through truncating the topics' words to the top n with the highest probability using the topic model's original word ordering. The first baseline label consists of the top two words (**Top-2 label**), in terms of highest marginal probabilities ($\hat{\varphi}_{w,t}$). While the second baseline label consists of the top three words (**Top-3 label**). The intuition behind these baselines is that for each topic, the highest-ranked words given by the original ordering from the topic model account for most of the information describing the topic's subject and therefore they are suitable as a label for the topic (Lau et al., 2010).

Table 6.4: The hyperparameters explored for training the proposed models in Chapter 5. In case of multiple values, **bold** font denotes the value used in training the final model.

| | **CNN** | **BiGRU & BiLSTM** | **CNN-GRU & CNN-LSTM** | **Transformer** |
|---|---|---|---|---|
| **optimizer** | adam | **adam**, rmsprop | adam | adam |
| **learning rate** | 1e-3 | **1e-3**, 1e-4, 1e-5 | 1e-3 | 5e-4 |
| **embedding dimensions** | **128**, 256 | 200, **300**, 400 | 128, **256** | 128, **256**, 512 |
| **hidden dimensions** | **256**, 512 | **200**, 300, 400 | 256, **512** | **512**, 1024 |
| **encoder layers** | 5, 7, **10**, 20 | **1**, 2 | 3, **5**, 7 | **3**, 6 |
| **decoder layers** | 5, 7, **10**, 20 | **1**, 2 | 1 | **3**, 6 |
| **dropout** | **0.2**, 0.5 | **0.1**, 0.2, 0.4 | 0.2 | **0.1**, 0.2, 0.4 |
| **clip** | 0.1, **0.2**, 0.5 | - | **0.1**, 0.2 | 1 |
| **positional encoding** | without, **with** | - | **without**, with | with |
| **kernel** | 3 | - | 3 | - |
| **filter** | 512 | - | 1024 | - |
| **attention heads** | - | - | - | 8 |

## 6.4 Topic Labels Quality Estimation

### 6.4.1 Comparison of Generated Labels with Gold Labels

The generated labels are compared with human-rated ones. For that purpose, the dataset by Bhatia et al. (2016) is used which contains pairs of topics and labels together with human scores of relevance. The similarity is computed using BERTScore (Zhang et al., 2019)[3], a text generation evaluation metric that uses contextual embeddings to match terms from the reference sentence with terms from the candidate sentence using cosine similarity and return the overall similarity between the two sentences. BERTScore has been shown to have high correlation with human judgements.

The BERTScore metric starts by aligning each word in the reference sentence with its most similar one from the candidate sentence. Alignments are based on a pairwise cosine similarity between the words' embeddings. The embeddings are generated using BERT (Devlin et al., 2019) that learns word representations via a transformer encoder network (Vaswani et al., 2017). The final similarity score is the sum of the maximum alignment between the words. Figure 6.2 shows the process of alignment and matching between two sentences and the final score computation.

Formally, given a reference sentence $y = (y_1, y_2, \ldots, y_m)$ and a candidate sentence $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$, BERTScore$(y, \hat{y})$ is computed as follows. First, generate contextual embeddings for $y$ and $\hat{y}$ using BERT, resulting in a sequence of vectors $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$ and $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_n)$. Then, calculate the similarity between the vectors via cosine similarity. For example the cosine similarity between reference word $\mathbf{y}_i$ and candidate word $\hat{\mathbf{y}}_j$ is computed as

$$\text{sim}(\mathbf{y}_i, \hat{\mathbf{y}}_j) = \mathbf{y}_i^\top \hat{\mathbf{y}}_j \tag{6.1}$$

[3]Results were generated using the reference implementation: https://github.com/Tiiiger/bert_score

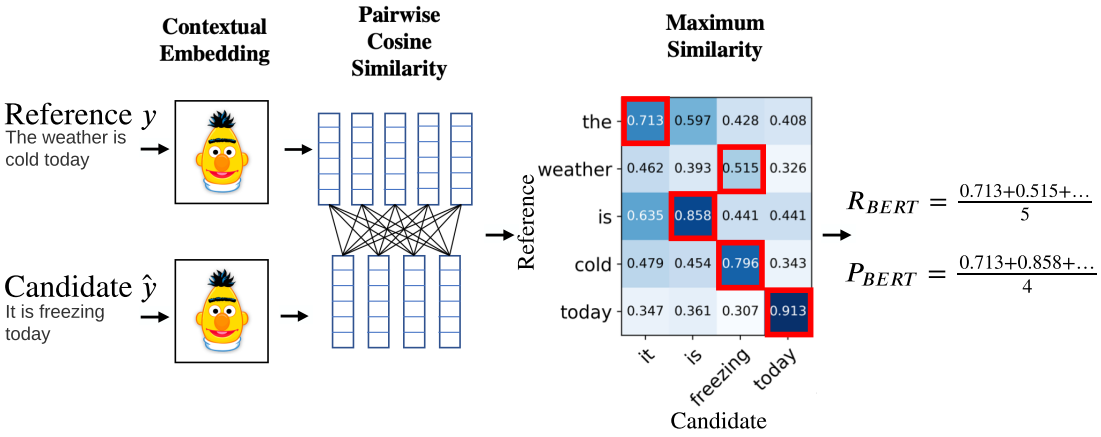



Figure 6.2: The process of computing pair-wise similarity and matching words between the reference sentence $y$ and candidate sentence $\hat{y}$ in BERTScore (Zhang et al., 2019). In this figure, two scores are computed, recall ($R_{\text{BERT}}$) and precision ($P_{\text{BERT}}$).

Finally, greedy matching between the reference vectors and candidate vectors is performed to compute recall ($R_{\text{BERT}}$), precision ($P_{\text{BERT}}$) and F1 ($F_{\text{BERT}}$) scores as follows:

$$R_{\text{BERT}} = \frac{1}{|y|} \sum_{y_i \in y} \max_{\hat{y}_j \in \hat{y}} \text{sim}(\mathbf{y}_i, \hat{\mathbf{y}}_j) \tag{6.2}$$

$$P_{\text{BERT}} = \frac{1}{|\hat{y}|} \sum_{\hat{y}_j \in \hat{y}} \max_{y_i \in y} \text{sim}(\mathbf{y}_i, \hat{\mathbf{y}}_j) \tag{6.3}$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{6.4}$$

Since BERTScore does not rely on exact string matches between candidate and reference labels, it is able to match appropriate label words with reference words even if label word does not appear in the reference label. The test datasets (topics_bhatia and topics_tfidf) have multiple appropriate reference labels for each topic (see Table 6.1) and therefore a pairwise BERTScore is performed. The pairwise BERTScores for topic $t_i$ between the reference labels $l_i = (l_{i,1}, ..., l_{i,m})$ and the candidate label $\hat{l}_i$

is computed as follows:

$$\text{score\_topic}_i = \max_{q=[1,\ldots,m]} \text{BERTScore}(l_{i,q}, \hat{l}_i)$$

The model's overall score is the mean score over all topics:

$$score\_model = \frac{1}{K}\sum_{i=1}^{K} score\_topic_i$$

### 6.4.2 Comparison of Generated Labels with Topics

Useful labels should also meet additional criteria, such as being highly related to the topic (Mei et al., 2007, Wan and Wang, 2016). A number of metrics are used to further estimate the quality of the labels:

**Relevance** Labels are expected to be relevant to the topic. Adopted from (Mei et al., 2007, Wan and Wang, 2016), the relevance is measured as the semantic similarity between the generated labels and the topics using BERTScore. This is a similar approach to the one described earlier (section 6.4.1), but instead, the similarity is measured between the candidate label and the topic. For example, given topic $t_i = (w_1, \ldots, w_P)$ and candidate label $\hat{l}_i = (w_1, \ldots, w_Q)$, relevance is computed as

$$\text{Relevance}(t_i, \hat{l}_i) = \text{BERTScore}(t_i, \hat{l}_i) \tag{6.5}$$

**Discrimination** Labels that have high relevance to multiple topics are not useful since they make interpretability harder for the user. This metric is also adopted from a metric proposed by (Mei et al., 2007). Discrimination is computed by subtracting the label's overall relevance to all topics from its relevance to its topic. The discrimination score for $\hat{l}_i$ given the topics $T$ is computed as follows:

$$\text{Discrimination}(T, \hat{l}_i) = \text{Relevance}(t_i, \hat{l}_i) - \frac{1}{K-1}\sum_{k=1}^{K_{-i}} \text{Relevance}(t_k, \hat{l}_i) \tag{6.6}$$

where $K_{-i}$ denotes all elements of $K$ except $i$.

**Coverage** Labels that cover more information from their topics are considered more useful and representative of the topic (Mei et al., 2007). The coverage of a label is measured as the fraction of a topic's terms that are also in the generated label:

$$\text{Coverage}(t_i, \hat{l}_i) = \frac{\sum_{q=1}^{Q} \mathbf{1}(\hat{l}_{i,q} \in t_i)}{|t_i|}, \tag{6.7}$$

where $\mathbf{1}$(condition) denotes 1 if the condition is true, 0 otherwise; and $|.|$ denotes length.

**Repetition** Generative approaches tend to produce redundant terms and labels with repetitive terms are not favoured. Repetition is computed as the mean of the number of repeated terms in each label,

$$\text{Repetition}(\hat{l}_i) = \frac{\sum_{q=1}^{Q} \#(\hat{l}_{i,q} \in \hat{l}_i)}{|\{\hat{l}_{i,q}\}_{q \in \{1,\ldots,Q\}}|}, \tag{6.8}$$

where $\#(.)$ denotes count; $\{\}$ denotes a unique set of words.

The two metrics, relevance and discrimination, use BERTScore, which employs contextualised word representation returned from BERT. Contextual word representations have different representations for the same word based on its surrounding words. This raises the question of the appropriateness of BERTScore in measuring the relevance and discrimination between labels and topics since topic's constitute words that do not have any order. A study has been performed to explore the effect of shuffling the topic's word on the stability of BERTScore. Appendix C includes bar plots that show the relevance and discrimination for the models under different configurations. Results show that the metrics maintained the relationships between the models and delivered the same conclusion even after shuffling. This can be used to describe BERTScore as appropriate metrics for the topics even when the order is not considered.

Table 6.5: BERTScore F1 measure between the generated labels and human-rated labels. **Bold** numbers indicate the highest score in a column.

| **Train data** | | ds_wiki_tfidf | | ds_wiki_sent | |
|---|---|---|---|---|---|
| **Test data** | | topics_bhatia | topics_tfidf | topics_bhatia | topics_tfidf |
| **Baselines** | | | | | |
| Top-2 label | 0.902 | - | - | - | - |
| Top-3 label | 0.882 | - | - | - | - |
| **Proposed models** | | | | | |
| BiGRU | - | 0.922 | 0.925 | 0.919 | 0.929 |
| BiLSTM | - | **0.926** | 0.924 | **0.936** | 0.933 |
| CNN | - | 0.916 | 0.927 | 0.903 | 0.927 |
| CNN-GRU | - | 0.848 | 0.913 | 0.905 | 0.912 |
| CNN-LSTM | - | 0.866 | 0.928 | 0.887 | 0.925 |
| Transformer | - | 0.915 | **0.931** | 0.916 | **0.937** |

## 6.5 Results and Discussion

### 6.5.1 Evaluation using Gold Labels

Table 6.5 shows the BERTScore between gold labels and the labels generated by the proposed models (see section 5.3 for the proposed models and section 6.4.1 for details about BERTScore). Most of the proposed models' performance was superior to the baselines (Top-2 label and Top-3 label), except for CNN and CNN-LSTM, when the model was trained with *ds_wiki_tfidf* and the labels generated using *topics_bhatia* (these topics consist of 10 terms only).

BiLSTM and Transformer are the best performing models. Performance of BiLSTM exceeds the baselines models under all cases but it does decay when using more words per topic (*topics_tfidf*). However, the Transformer model achieves the best results when *topics_tfidf* are used. This might be caused by the LSTM failing to handle long sequences while the Transformer is better able to retain information from earlier states in long sequences. In general, labels are scored higher when generated given 30 terms (*topics_tfidf*) compared to limiting the topic to 10 terms (*topics_bhatia*). This behaviour is witnessed under all models with the exception of BiLSTM.

Another conclusion that can be drawn is that training models on *ds_wiki_tfidf* does not make a substantial difference to performance, and in most cases using the raw sequences from Wikipedia (*ds_wiki_sent*) leads to better performance. The largest BERTScore of 0.937 was reported with the Transformer model that was trained on *ds_wiki_sent* while using the topics with 30 terms (*topics_tfidf*).

### 6.5.2 Label Relevance to Topics

Table 6.6 reports the scores for the four metrics described in section 6.4.2 that compare the generated labels with the topics themselves. Doing so is important since the main aim of the labelling task is to produce informative labels that represent the topics. It is worth noting that the baselines (**Top-2 label** and **Top-3 label**) are not included in this evaluation given that they are extracted from the topics' terms and this would skew the scores in their favour.

The highest relevance for the topics is observed for the CNN-LSTM followed by the Transformer model. The CNN model obtained the highest discrimination score, which means its generated labels are different across the topics. It has also produced labels that contain more topic terms than the other models (i.e., it has better coverage). The BiGRU model generates a smaller number of repetitive terms within the label which is to be expected as its labels tend to be short (see Tables 6.7 and Table B.1 for samples of the labels generated by the BiGRU model). Similarly, the Transformer labels contain minimal repetition. On the other hand, the CNN-based models (CNN, CNN-GRU, and CNN-LSTM) demonstrated the highest repetition within their labels. Overall, these results indicate that the Transformer model consistently performs well under all metrics and ranks second (except under discrimination where CNN-LSTM outperformed it by a small margin).

Table 6.6: Metrics to assess the quality of the generated labels in relation to the topics. † indicates scores are based on cosine similarity using BERTScore, while ‡ indicates scores are based on exact matches and only considers unigrams. An up arrow ↑ indicates that higher values are better and down arrow ↓ that lower values are better. **Bold** values denote the highest score across the models and underlined scores denote second highest. All reported scores are statistically significant using Wilcoxon signed-rank test with $p < 0.05$.

| Metric / Model | Relevance † ↑ | Discrimination † ↑ | Coverage ‡ ↑ | Repetition ‡ ↓ |
|---|---|---|---|---|
| **BiGRU** | 0.819 ± 0.01 | 0.016 ± 0.01 | 0.092 ± 0.05 | **0.011** ± 0.01 |
| **BiLSTM** | 0.823 ± 0.01 | 0.020 ± 0.01 | 0.112 ± 0.04 | 0.038 ± 0.02 |
| **CNN** | 0.822 ± 0.02 | **0.025** ± 0.01 | **0.144** ± 0.06 | 0.146 ± 0.06 |
| **CNN-GRU** | 0.785 ± 0.03 | 0.011 ± 0.01 | 0.077 ± 0.01 | 0.394 ± 0.32 |
| **CNN-LSTM** | **0.827** ± 0.01 | 0.024 ± 0.01 | 0.120 ± 0.05 | 0.117 ± 0.14 |
| **Transformer** | 0.824 ± 0.01 | 0.021 ± 0.01 | 0.121 ± 0.04 | 0.026 ± 0.02 |

### 6.5.3 Qualitative Analysis

Table 6.7 shows sample topics with their gold labels and the generated automatic labels (Additional samples can be found in Appendix B.1). The first topic is about sports, in particular football which is indicated by the terms {*football, nfl*}. The BiGRU model generated a generic label that could be related to any sport. The BiLSTM model produced a highly specific label {*new york yankees season*} which is relatively coherent but misleading since it refers to baseball rather than football. The labels generated by the CNN and CNN-LSTM models are related to the topic but generic and contain repetitive terms, while the CNN-GRU model generated *knight* which can be a football team name or an honour. Specific labels such as this one require additional knowledge to make the desired association. On the other hand, the Transformer's label is coherent, specific and representative of the topic.

The second topic is about the 2008 Olympic games in China. BiGRU also produced a generic label in this example, whereas the BiLSTM and Transformer return coherent and informative labels. CNN and CNN-LSTM maintained the same behaviour as in topic 1, where their labels have unnecessary repetitive terms. Labels generated with (CNN, CNN-GRU and CNN-LSTM) models are from the correct

domain, but they are less coherent when compared to the other models.

An error analysis was carried out to examine cases where the models produced sub-optimal labels. For example, the third topic in Table 6.7 was not assigned an appropriate label by most of the models which may be due to the topic being incoherent and having no obvious theme. Topic four is another example where the models failed to produce appropriate labels. The topic is from a medical domain and contains domain-specific words which are not in the vocabulary that the models were trained on, since Wikipedia is a general domain resource.

## 6.6 Conclusion

This chapter presented the implementation details for an automatic topic label generation approach using several neural network models. The chapter also included automatic evaluation approaches to estimate the quality of the generated labels in addition to qualitative analysis that highlights the coherence and relevance of the labels to the topics. The proposed labelling models have shown promising results and the Transformer-based model stands out as being able to produce coherent labels.

| | |
|---|---|
| **Topic 1** | game team play season yard coach quarterback nfl run football |
| **Gold labels** | playoffs, head coach, national football league |
| **BiGRU** | season |
| **BiLSTM** | new york yankees season |
| **CNN** | football football team |
| **CNN-GRU** | knights |
| **CNN-LSTM** | football football football football game |
| **Transformer** | nfl game |
| **Topic 2** | china chinese olympics gold olympic team win beijing medal sport |
| **Gold labels** | olympic gold, summer olympic games, winter olympic games |
| **BiGRU** | china |
| **BiLSTM** | china at the summer paraympics |
| **CNN** | china chinese olympics of the olympics |
| **CNN-GRU** | gold at the summer olympics |
| **CNN-LSTM** | china china at the asia |
| **Transformer** | china at the summer olympics |
| **Topic 3** | mr mrs young lady look friend tell mother miss father |
| **Gold labels** | aunt, wife |
| **BiGRU** | the |
| **BiLSTM** | the |
| **CNN** | tell |
| **CNN-GRU** | young |
| **CNN-LSTM** | mr |
| **Transformer** | the devil is |
| **Topic 4** | artery vascular coronary stent vein vessel carotid aortic aneurysm arterial |
| **Gold labels** | pulmonary artery, ascending aorta, aneurysm, blood vessel, aortic stenosis, stenosis, aorta, aortic valve, aortic aneurysm |
| **BiGRU** | hatun |
| **BiLSTM** | gaius |
| **CNN** | effects of |
| **CNN-GRU** | syndrome |
| **CNN-LSTM** | combodia |
| **Transformer** | ma |

Table 6.7: Labelling samples from the proposed models.

# 7

# CONCLUSIONS AND FUTURE WORK

This thesis introduced several approaches for improving the interpretability of topics generated automatically using topic models. This chapter presents a summary of the findings and contributions of the thesis and proposes directions for future research.

## 7.1 Summary of Thesis Contributions

As stated in Chapter 1, the main aim of this thesis is to propose ways of improving the interpretability of topics by humans. This aim was approached by addressing three sub-problems: (1) creating alternative topic representations that are more comprehensible; (2) evaluating the usefulness of topic representation; and (3) summarising the main subject of topics by assigning them short phrases.

Chapter 2 introduced topic models as a method of summarising document collections, and it presented various approaches to evaluating topic models. The chapter also discussed several extensions to topic models to accommodate a specific task or

address a drawback. The available approaches to evaluating the topic model's output were surveyed. Various ways of representing topic model outputs were presented in addition to the approaches employed to improve representations to promote their easier comprehension by humans.

Chapter 3 presented an approach to improving the output of topic models by re-ranking topics' words. The re-ranking methods were evaluated by a proposed approach based on an IR task to retrieve relevant documents. The re-ranking methods were found to be useful in identifying informative words and in discriminating relevant documents from non-relevant ones. In this evaluation approach, the most useful re-ranking methods were based on information in the corpus, which incorporated information about the importance of words, both within topics and in their relative frequency in the entire corpus.

Chapter 4 further evaluated the set of re-ranking metrics in the previous chapter. Four methods were assessed through an alternative evaluation approach based on a crowdsourcing experiment in which the participants were asked to associate articles with related topics. The crowdsourcing study was designed to assist the usefulness and interpretability of the topics produced by re-ranking. Re-ranking the topic words was found to improve the interpretability of the topics by the participants. Based on this finding, it could be used as a post-processing step to improve topic representation. The study included an investigation of the effects of using a simple word list representation and a word cloud representation on a topic's interpretability. The findings showed that word cloud representations were not advantageous to the interpretability of the topics by humans. Therefore, using a simple representation, such as a list of words, was found to be superior.

Chapter 5 introduced a new approach to topic labelling through neural networks. Various neural networks were explored, such as CNNs, RNNs, and Transformers. Labelling a topic was defined as a Seq2Seq problem, in which the topic's terms were applied to the model, and another set was produced as labels. The produced labels

reflected the subject covered and contained by the topic's terms.

Chapter 6 presented the datasets used to train and test the neural-based labelling model proposed in the previous chapter. The training dataset was created following a distant-supervision approach that can be used to resolve the issue of limited availability of data. On the other hand, the used testing data was publicly released by Bhatia et al. (2016), it contained pairs of topics and labels where the labels appropriateness for the topics was rated by humans. It also described in detail the implementation of the neural-based label generation approach. The generated labels were automatically evaluated by comparing them with a set of gold labels that were rated by humans. The generated labels were also compared with the topics' terms to ensure that an appropriate label was also representative of the topic. The proposed neural-based approach showed promising results and produced appropriate labels in most cases. The findings also showed that labels created using the transformer-based model were more coherent than those produced by the other models.

## 7.2 Future Work

The methods proposed in this thesis could be extended and improved as follows:

- **Topic Ranking**
  - The methods proposed for re-ranking the topic terms were based on information from the corpus (e.g., IDF) and semantic similarity between the topic terms using vector neural representations (e.g., Word2vec embeddings). However, the embeddings employed were context-independent, which meant that a single word had one vector representation regardless of its meaning in the context. For example, the word "cat" in a topic about animals refers to the species of a small carnivorous mammal, yet the same word within a medical topic about diseases and diagnosis refers to a diagnostic scan device called computerised axial tomography (CAT).

In both topics, the word has the same vector representation; therefore, misleading similarities between the topic's words would be inferred. A possible alternative ranking approach could make use of context-aware word embeddings (e.g., ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019)). Such word representations were generated using large language models and therefore learned separate vector representations of a word depending on its surroundings.

- **Topic Labelling**
  - In Chapter 5 and 6, the topics used in the labelling task were in the original order given by the topic model. However, the effect of re-ranking the topic's words on the resulting label was not explored. Because re-ranking has been shown to improve the topics' interpretability, it is possible that it would also have a positive effect on the generated labels.
  - The hypothesis search in the neural labelling model followed an approach in which the maximum probability word was chosen in each step (i.e., the greedy approach). Following this approach, the decoder always prefers easier words, and multiple common words are predicted more often than rare words (Ippolito et al., 2019). However, a broader hypothetical space for decoding can be used (i.e., the beam search approach), which has been shown to generate better sequences (Ippolito et al., 2019). At each time step, $n$ possible hypotheses are considered, which allows for a wide variety of potential sequences. Finally, the sequence with the largest product of probability is the chosen prediction.
  - The labelling model has an embedding layer that is learned word embeddings with the model. However, previous NLP work using neural networks has shown that using pre-trained embeddings is effective and provides additional useful information to the model without the extra cost of training it (Liu et al., 2015).

# BIBLIOGRAPHY


Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. Syntactically supervised Transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*, pages 1269–1281, Florence, Italy, 2019.

Nikolaos Aletras and Arpit Mittal. Labeling topics with images using neural networks. In *Proceedings of the European Conference on Information Retrieval (ECIR '17)*, pages 500–505, Aberdeen, UK, 2017.

Nikolaos Aletras and Mark Stevenson. 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 '13)*, pages 158–167, Atlanta, Georgia, 2013a.

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on Computational Semantics (IWCS '13)*, pages 13–22, , Potsdam, Germany, 2013b.

Nikolaos Aletras and Mark Stevenson. Labelling topics using unsupervised graph-based methods. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14)*, pages 631–636, Baltimore, Maryland, 2014.

Nikolaos Aletras, Timothy Baldwin, Jey Lau, and Mark Stevenson. Representing topics labels for exploring digital libraries. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '14)*, pages 239–248, London, United Kingdom, 2014.

Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. Evaluating topic representations for exploring document collections. *Journal of the Association for Information Science and Technology*, 68(1):154–167, 1 2017.

Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. *Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2009.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceeding of the 3rd International Conference on Learning Representations (ICLR '15)*. 2015.

Lukas Barth, Stephen Kobourov, and Sergey Pupyrev. Experimental Comparison of Semantic Word Clouds. In *Proceedings of the 13th International Symposium on Experimental Algorithms (SEA '14)*, pages 247–258, Copenhagen, Denmark, 2014.

André Bergholz, Jeong-Ho Chang, Gerhard Paass, Frank Reichartz, and Siehyun Strobel. Improved phishing detection using model-based features. In *Proceedings of the Conference on Email and Anti-Spam (CEAS '08)*, pages 1–10, Mountain View, CA, 2008.

Shraey Bhatia, Jey Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING '16)*, page 953–963, Osaka, Japan, 2016.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. An automatic approach for document-level topic model evaluation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL '17)*, pages 206–215, Vancouver, Canada, 6 2017.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Topic intrusion for automatic topic model evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 18)*, pages 844–849, Belgium, Brussles, 2018.

Jonathan Bischof and Edoardo Airoldi. Summarizing topical content with word frequency and exclusivity. In *Proceedings of the 29 th International Conference on Machine Learning (ICML '12)*, page 201–208, Edinburgh, Scotland, UK, 2012.

David Blei and John Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*, 18:147–154, 2006.

David Blei and John Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

David Blei and John Lafferty. Topic models. *Text mining: Classification, Clustering, and Applications*, 10(71):34, 2009a.

David Blei and John Lafferty. Visualizing topics with multi-word expressions. *arXiv preprint*, arXiv:0907, 2009b.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint*, 7 2016.

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL '09)*, pages 31–40, Potsdam, Germany, 2009.

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

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 4 1998.

Guoray Cai, Feng Sun, and Yongzhong Sha. Interactive visualization for topic model curation. In *Proceeding of Exploratory Search and Interactive Data Analytics (ESIDA '18)*, Tokyo, Japan, 2018.

Amparo Elizabeth Cano Basave, Yulan He, and Ruifeng Xu. Automatic labelling of topic models learned from twitter by summarisation. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics (ACL '14)*, page 618–624, Baltimore, USA, 2014.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proceeding of the 2020 European Conference on Computer Vision (ECCV '20)*, pages 213–229, Glasgow, Scotland, 2020.

Quim Castella and Charles Sutton. Word storms: Multiples of word clouds for visual comparison of documents. In *Proceedings of the 23rd international conference on World wide web (IW3C2 '14)*, pages 665–676, Seoul, Korea, 2014.

Yougchul Cha and Junghoo Cho. Social-network analysis using topic models. In *Proceedings of the 35th international ACM SIGIR conference on Research and*

*development in information retrieval (SIGIR '12)*, pages 565–574, Portland, Oregon, 2012.

Allison Chaney and David Blei. Visualizing topic models. In *Proceedings of the 6th International AAAI Conference Weblogs and Social Media (AAAI ICWSM '12)*, pages 419–422, Dublin, Ireland, 2012.

Jonathan Chang and David Blei. Relational topic models for document networks. In *Proceedings of the 12th International Confe-rence on Artificial Intelligence and Statistics (AISTATS '09)*, pages 81–88, Clearwater Beach, Florida, 2009.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS '09)*, pages 288–296, Columbia, Canada, 2009.

Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*, pages 1724–1734, Doha, Qatar, 2014.

Jason Chuang, Christopher Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceeding of the International Working Conference on Advanced Visual Interfaces (AVI '12)*, pages 74–77, Capri Island, Italy, 2012.

Mark Craven and Johan Kumlien. Constructing bological knowledge bases by extracting information from text sources. In *Proceeding of the 7th Inter-national Conference on Intelligent Systems for Molecular Biology (ISMB '99)*, pages 77–86, Heidelberg, Germany, 1999.

Yann Dauphin, Angela Fan, Michael Auli, and David. Grangier. Language modeling with gated linear units. *arXiv preprint*, 2016.

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

Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceeding of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '19)*, pages 4171–4186, Minneapolis, Minnesota, 2019.

Adji Dieng, Francisco Ruiz, and David Blei. Topic Modeling in Embedding Spaces. *Transactions of the Association for Computational Linguistics (TACL)*, 8:439–453, 2020.

Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-aware neural topic modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (ACL '18)*, pages 830–836, Melbourne, Australia, 2018.

Jacob Eisenstein, Duen Horng, Aniket Kittur, and Eric P Xing. Topicviz: Interactive topic exploration in document collections. In *Proceeding of Extended Abstracts on Human Factors in Computing Systems (CHI '12)*, pages 2177–2182, Austin, TX, 2012.

Thomas Fruchterman and Edward Reingold. Graph Drawing by Force-directed Placement. Technical Report 1, 1991.

Matthew Gardner, Joshua Lutes, Jeff Lund, Josh Hansen, Dan Walker, Eric Ringger, and Kevin Seppi. The topic browser: An interactive tool for browsing topic models. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*, pages 1–9, Vancouver, Canada, 2010.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning (PMLR '17)*, pages 1243–1252, Sydney, Australia, 2017.

Yoav Goldberg and Graeme Hirst. *Neural network methods in natural language processing*. Morgan & Claypool Publishers, 2017.

Sujatha Gollapalli and Xiao-li Li. Using PageRank for Characterizing Topic Quality in LDA. In *Proceedings of the ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '18)*, pages 115–122, Tianjin, China, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning book*, volume 521. 2016.

Thomas Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–35, 2004.

Thomas Griffiths, Mark Steyvers, David Blei, and Joshua Tenenbaum. Integrating topics and syntax. Technical report, 2005.

Thomas Griffiths, Mark Steyvers, and Joshua Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211, 2007.

Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. Neural topic model with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*, pages 3478–3483, Hong Kong, China, 2019.

Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009.

Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09)*, pages 362–370, Boulder, Colorado, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR '16)*, pages 770–778, Las Vegas, NV, 2016.

Yulan He, Chenghua Lin, and Harith Alani. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL '11)*, pages 123–131, Portland, Oregon, 2011.

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

Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent dirichlet allocation. In *Proceedings of the 23rd Advances in Neural Information Processing Systems (NIPS '10)*, pages 856–864, Vancouver Canada, 2010.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99)*, pages 50–57, Berkeley, CA, 1999.

Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. Technical report, 2001.

Diane Hu and Lawrence Saul. A probabilistic topic model for music analysis. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS '09)*, Vancouver, Canada, 2009.

Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM '13)*, pages 465–474, Rome, Italy, 2013.

Daphne Ippolito, Reno Kriz, Maria Kustikova, João Sedoc, and Chris Callison-Burch. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL '19)*, pages 3752–3762, Florence, Italy, 2019.

Shoaib Jameel and Wai Lam. An unsupervised topic segmentation model incorporating word order. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13)*, pages 203–212, Dublin, Ireland, 2013.

Shoaib Jameel, Bullet Wai Lam, Bullet Lidong Bing, Wai Lam, and Lidong Bing. Supervised topic models with word order structure for document classification and retrieval learning. *Information Retrieval Journal*, 18(4):283–330, 2015.

Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM Conference on Knowledge Discovery and Data Mining (KDD '06)*, page 217–226, Philadelphia, PA, 2006.

Daniel Jurafsky and James Martin. *Speech and language processing*. Pearson London, 2014.

Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR '11)*, pages 165–176, Berlin, Heidelberg, 2011.

Klaus Krippendorff. Computing krippendorff ' s alpha-reliability part of the communication commons. Technical report, 2011.

Simon Lacoste-Julien, Fei Sha, and Michael Jordan. DiscldA: Discriminative learning for dimensionality reduction and classification. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS 08)*, pages 897–904, Vancouver, Canada, 2008.

Thomas Landauer. Applications of latent semantic analysis. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, 2002.

Thomas Landauer and Susan Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

Jey Lau and Timothy Baldwin. The sensitivity of topic coherence evaluation to topic cardinality. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '16)*, pages 483–487, San Diego, California, 2016.

Jey Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*, pages 605–613, Beijing, China, 2010.

Jey Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '11)*, pages 1536–1545, Portland, Oregon, 2011.

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

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

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Jimmy Lei Ba, Jamie Kiros, and Geoffrey Hinton. Layer normalization. *arXiv preprint*, 7 2016.

Wei Li and Andrew Mccallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pages 577–584, Pittsburgh, Pennsylvania, 2006.

Marie Liénou, Henri Maître, and Mihai Datcu. Semantic annotation of satellite images using latent dirichlet allocation. *IEEE Geoscience and Remote Sensing Letters*, 7(1):28–32, 1 2010.

Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*, pages 17–21, Lisbon, Portugal, 2015.

Hsin Lu, Chih Wei, and Fei Hsiao. Modeling healthcare data using multiple-channel latent dirichlet allocation. *Journal of biomedical informatics*, 60:210–223, 2016.

Yue Lu, Qiaozhu Mei, and ChengXiang Zha. Investigating task performance of probabilistic topicmodels: An empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203, 2011.

Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic labeling of topics. In *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA '09)*, pages 1227–1232, Pisa, Italy, 2009.

Winter Mason and Siddharth Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 3 2012.

Julian McAuley, Christopher Targett, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*, pages 43–52, Shanghai, China, 2015.

Jon Mcauliffe and David Blei. Supervised topic models. *Advances in neural information processing systems*, pages 121–128, 2008.

Andrew Mccallum, Andrés Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 786–791, Edinburgh, Scotland, 2005.

Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text. In *Proceeding of the eleventh ACM SIGKDD international conference on*

*Knowledge discovery in data mining (KDD '05)*, pages 198–207, Chicago, Illinois, 2005.

Qiaozhu Mei and ChengXiang Zhai. A mixture model for contextual text mining. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06)*, pages 649–655, Philadelphia, PA, 2006.

Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, pages 533–542, Edinburgh, Scotland, 2006.

Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07)*, pages 490–499, San Jose, California, 2007.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceesing of the 1st International Conference on Learning Representations, Workshop Track (ICLR '13)*, page 500–509, Scottsdale, Arizona, 1 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2013)*, pages 3111–3119, Lake Tahoe, Nevada, 2013b.

David Mimno and Andrew Mccallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07)*, San Jose, California, 2007.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew Mccallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 262–272, Edinburgh, United Kingdom, 2011.

Thomas M Mitchell. *Machine learning*. McGraw-Hill, Inc., USA, 1 edition, 1997.

John Mohr and Petko Bogdanov. Introduction-topic models: What they are and why they matter. *Poetics*, 41(6):545–569, 2013.

Kevin Murphy. *Machine learning: A probabilistic perspective*. MIT press, 2012.

Claudiu Musat, Julien Velcin, Stefan Trausan-Matu, and Marian-Andrei Rizoiu. Improving topic evaluation using conceptual knowledge. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI '11)*, pages 1866–1871, Barcelona, Spain, 2011.

Naresh Nagwani. Summarizing large text collection using topic modeling and clustering based on mapreduce framework. *Journal of Big Data*, 2, 2015.

Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pages 807–814, Haifa, Israel, 2010.

David Newman, Han Lau, Karl Grieser, and Timothy Baldwin. 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'10)*, pages 100–108, Los Angeles, California, 2010.

Viet Nguyen, Yuening Hu, Jordan Boyd-Graber, and Philip Resnik. Argviz: Interactive visualization of topic dynamics in multi-party conversations. In *Proceedings of the 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*, pages 36–39, Atlanta, Georgia, 2013.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 311–318, Philadelphia, Pennsylvania, 2002.

Michael Paul. Interpretable machine learning: Lessons from topic modeling. In *Proceedings of Human-Computer Interaction Workshop on Human-Centered Machine Learning (CHI HCML '16)*, San Jose, California, 2016.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*, pages 1532–1543, Doha, Qatar, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 16th North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '18)*, pages 2227–2237, New Orleans, Louisiana, 2 2018.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08)*, pages 569–577, Las Vegas, Nevada, 2008.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 conference on empirical methods in natural language processing (EMNLP '09)*, pages 248–256, Suntec, Singapore, 2009.

Daniel Rampage, Susan Dumais, and Daniel Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM '10)*, pages 338–349, Washington, DC, 2010.

Yafeng Ren, Ruimin Wang, and Donghong Ji. A topic-enhanced word embedding for Twitter sentiment classification. *Information Sciences*, 369:188–198, 11 2016.

Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 10 2004.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eight ACM International Conference on Web Search and Data Mining (WSDM '15)*, pages 399–408, Shanghai, China, 2015.

David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning representations by back-propagating errors. *Nature*, 323(19):533–536, 1986.

Pedro Santos, Lisa Beinborn, and Iryna Gurevych. A domain-agnostic approach for opinion prediction on speech. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 163–172, Osaka, Japan, 2016.

Ferdinand Saussure. *Course in general linguistic*. McGraw-Hill, 1959.

Stefano Sbalchiero and Maciej Eder. Topic modeling, long texts and the best number of topics. Some Problems and solutions. *Quality and Quantity*, 54(4):1095–1108, 8 2020.

Alexandra Schofield, Måns Magnusson, and David Mimno. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*, volume 2, pages 432–436, Valencia, Spain, 2017.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Preceedings of the Workshop on Interactive Learning Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, 2014.

Alison Smith, Jason Chuang, Yuening Hu, Jordan Boyd-Graber, and Leah Findlater. Concurrent visualization of relationships between words and topics in topic models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 79–82, Baltimore, Maryland, 2014a.

Alison Smith, Timothy Hawes, and Myers Myers. Hiérarchie: Interactive visualization for hierarchical topic models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces,*, pages 71–78, Baltimore, Maryland, 2014b.

Alison Smith, Sana Malik, and Ben Shneiderman. Visual analysis of topical evolution in unstructured text: Design and evaluation of topicflow. In *Applications of Social Media and Social Network Analysis*, pages 159–175. Springer, 2015.

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

Justin Snyder, Rebecca Knowles, Mark Dredze, Matthew Gormley, and Travis Wolfe. Topic models and metadata for visualizing text corpora. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '13)*, pages 5–9, Atlanta, Georgia, 2013.

Yangqiu Song, Shimei Pan, Shixia Liu, Michelle Zhou, and Weihong Qian. Topic and keyword re-ranking for LDA-based topic modeling. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pages 1757–1760, Hong Kong, China, 2009.

Ionut Sorodoc, Jey Han Lau, Nikolaos Aletras, and Timothy Baldwin. Multimodal topic labelling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*, volume 2, pages 701–706, 2017.

Asbjørn Steinskog, Jonas Therkelsen, and Björn Gambäck. Twitter topic modeling by Tweet aggregation. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NACL '17)*, pages 77–86, Gothenburg, Sweden, 2017.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. *Empirical Methods in Natural Language Processing*, 20:952–961, 2012.

Mark Steyvers and Tom Griffiths. *Probabilistic topic models*. 2007.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Proceedings of the 28th Internation Conference on Neural Information Processing Systems (NIPS '15)*, pages 2440–2448, Montreal, Canada, 2015.

Ilya Sutskever Google, Oriol Vinyals Google, and Quoc V Le Google. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS '14)*, pages 3104–3112, Montreal, Canada, 2014.

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12): 2295–2329, 2017.

Matthew Taddy. On estimation and selection for topic models. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS '12)*, page 1184–1193, La Palma, Canary Islands, 2012.

Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):566–1581, 2006.

Ivan Titov and Ryan Mcdonald. A joint model of text and aspect ratings for sentiment summarization. In *Preceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '08)*, pages 308–316, Columbus, Ohio, 2008.

Kristina Toutanova and Mark Johnson. A bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS '08)*, pages 1521–1528, Vancouver, Canada, 2008.

Edward Tufte and Glenn Schmieg. *The visual display of quantitative information.* American Association of Physics Teachers, 1985.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 11–16, Uppsala, Sweden, 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention as all you need. In *Preceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*, pages 5998–6008, Long Beach, CA, 2017.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pages 3156–3164, Boston, MA, 2015.

Hanna Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning (ICML '06)*, pages 977–984, Pittsburgh, Pennsylvania, 2006.

Hanna Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pages 1105–1112, Montreal, Canada, 2009.

Xiaojun Wan and Tianming Wang. Automatic labeling of topic models using text summaries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16)*, pages 2297–2305, Berlin, Germany, 2016.

Chong Wang, David Blei, and Li Fei-Fei. Simultaneous image classification and annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pages 1903–1910, Miami, FL, 2009.

Xuerui Wang and Andrew Mccallum. 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 (KDD '06)*, pages 424–433, Philadelphia, PA, 2006.

Xuerui Wang, Andrew Mccallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM '07)*, pages 697–702, Omaha, NE, 2007.

Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. Technical report, 2006.

Pengtao Xie and Eric P. Xing. Integrating document clustering and topic modeling. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI '13)*, pages 694–703, Bellevue, Washington, 9 2013.

Linzi Xing and Michael J Paul. Diagnosing and improving topic models by analyzing posterior variability. In *Proceedings of the Advancement of Artificial Intelligence (AAAI '18)*, pages 6005–6012, New Orleans, Louisiana, 2018.

Michael Yandex and Natalia Loukachevitch. A method of accounting bigrams in topic models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '15)*, pages 1–9, Denver, Colorado, 2015.

Hui Yang. Constructing task-specific taxonomies for document collection browsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*, pages 1278–1289, Jeju Island, Korea, 2012.

Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. In *Proceedings of 31th European Conference on Information Retrieval (ECIR '09)*, pages 29–41, Toulouse, France, 2009.

Jia Zeng, William K. Cheung, and Jiming Liu. Learning topic models by belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1121–1134, 2013.

Cheng Zhang, Carl Ek, Xavi Gratal, Florian Pokorny, and Hedvig Kjellstrom. Supervised hierarchical dirichlet processes with variational inference. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW '13)*, pages 254–261, Sydney, Australia, 2013.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations (ICLR '20)*, volume abs/1904.0, Addis Ababa, Ethiopia, 4 2019.

Jun Zhu, Amr Ahmed, and Eric Xing. MedldA: Maximum margin supervised topic models. *Journal of Machine Learning Research*, 13:2237–2278, 2012.

Jun Zhu, Xun Zheng, and Bo Zhang. Improved Bayesian logistic supervised topic models with data augmentation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, pages 187–195, Sofia, Bulgaria, 2013a.

Jun Zhu, Xun Zheng, Li Zhou, and Bo Zhang. Scalable inference in max-margin topic models. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD '13)*, pages 964–972, Chicago, Illinois, 2013b.

# A

# ETHICAL APPROVAL

The University Of Sheffield.

# Application 013744

## Section A: Applicant details

Date application started:
Wed 12 April 2017 at 14:22

First name:
Areej Nasser A

Last name:
Alokaili

Email:
areej.okaili@sheffield.ac.uk

Programme name:
PhD/Computer Sci

Module name:
N/A

Last updated:
11/07/2019

Department:
Computer Science

Applying as:
Postgraduate research

Research project title:
Visual Representations of Documents

Has your research project undergone academic review, in accordance with the appropriate process?
No

Similar applications:
*- not entered -*

## Section B: Basic information

### Supervisor

| Name | Email |
|---|---|
| Mark Stevenson | mark.stevenson@sheffield.ac.uk |

### Proposed project duration

Start date (of data collection):
Tue 25 July 2017

Anticipated end date (of project)
Fri 5 February 2021

### 3: Project code (where applicable)

Project code
*- not entered -*

### Suitability

Takes place outside UK?
No

Involves NHS?
No

Health and/or social care human-interventional study?
No

ESRC funded?
No

Likely to lead to publication in a peer-reviewed journal?
Yes

Led by another UK institution?
No

Involves human tissue?
No

Clinical trial or a medical device study?
No

Involves social care services provided by a local authority?
No

Involves adults who lack the capacity to consent?
No

Involves research on groups that are on the Home Office list of 'Proscribed terrorist groups or organisations?
*- not entered -*

### Indicators of risk

Involves potentially vulnerable participants?
No

Involves potentially highly sensitive topics?
No

## Section C: Summary of research

### 1. Aims & Objectives

The project aims to assess and compare the usefulness of a range of visual representations for documents (or sets of documents). A wide range of such representations has been developed and used within document browsing interfaces, examples include word clouds and lists of key terms. However, they have not been systematically compared and evaluated. Many of these representations have been generated for the output of topic models, statistical algorithms that aim to identify the underlying themes from a collection of documents. The aim of the study is to evaluate representations for topic models by determining whether individuals can identify the one that is most appropriate for a particular document.

### 2. Methodology

We will create a job on a standard crowdsourcing platform (CrowdFlower) in which participants are presented with a document and representations for a small number of topics. Participants will be asked to select the topic that is most suitable for that document (An example can be seen in Figure 1 in the Consent form document).

Topics will be created from a standard Natural Language Processing corpus (e.g. 20 Newsgroups, Reuters Corpus or a Wikipedia dump) and manually checked to ensure that none contain potentially offensive content. Questions will then be automatically generated by randomly selecting a document from the corpus and identifying one topic that is closely associated with the document (using the topic model's document-topic probability distribution) and three that are not. These questions will also be manually checked.

3. Personal Safety

Have you completed your departmental risk assessment procedures, if appropriate?

*- not entered -*

Raises personal safety issues?

No

*- not entered -*

## Section D: About the participants

1. Potential Participants

The main route for attracting participants will be through the CrowdFlower crowdsourcing website. We would also like to keep open the option of using the University of Sheffield volunteers list.

2. Recruiting Potential Participants

The participants will be approached through CrowdFlower's platform.

2.1. Advertising methods

Will the study be advertised using the volunteer lists for staff or students maintained by CiCS? Yes

We will initially offer tasks via CrowdFlower's interface without advertising on the volunteer's list. We would like to retain the option to distribute it on the list to attract additional participants if required, although we may not need to do so.

3. Consent

Will informed consent be obtained from the participants? (i.e. the proposed process) Yes

Each user will be shown a consent form and will be unable to start the task until they have indicated that they have read it and agree with its contents.

4. Payment

Will financial/in kind payments be offered to participants? Yes

A small payment will be offered to compensate volunteers for their time. This will ensure that results can be gathered quickly. The payment will be made through the CrowdFlower platform.

We anticipate compensating volunteers 10 US cents per page of five judgments (CrowdFlower's suggested payment). We may vary this amount to control the number of annotators the task attracts, but not by a substantial amount.

5. Potential Harm to Participants

What is the potential for physical and/or psychological harm/distress to the participants?

No potential for harm to participants is anticipated. Volunteers are only asked to complete an online survey consisting of a number of small tasks. The questions will be manually checked to ensure that no documents or topics with potential to cause distress are included.

How will this be managed to ensure appropriate protection and well-being of the participants?

Participants must actively volunteer to participate in the study and are free to withdraw at any point. Crowdflower's terms and conditions state that users of the site must be at least 18 years of age.

## Section E: About the data

1. Data Confidentiality Measures

The only information that will be stored will be the anonymized responses to tasks. All responses are anonymized in the sense that we will only receive from CrowdFlower the participant's answers to the questions paired with their IDs and we have no ability to use their IDs to personally identify participants. No other personal information will be stored. The Crowdflower platform collects personal information, but this is not available to us.

2. Data Storage

The Primary analysis will take place on Areej Alokaili's computer. Some additional analysis may also take place on Mark Stevenson's computers. No encryption will be used given the nature of the data.

We expect the data to be made available for future research projects and will retain copies. If the data proves to be interesting and useful enough we may make it available to other researchers (e.g. via a URL published in a research paper).

## Section F: Supporting documentation

Information & Consent

Participant information sheets relevant to project?
Yes

| | |
|---|---|
| Document 1032590 (Version 1) | All versions |
| Document 1046626 (Version 1) | All versions |

Consent forms relevant to project?
Yes

| | |
|---|---|
| Document 1032591 (Version 2) | All versions |

Additional Documentation

| | |
|---|---|
| Document 1079157 (Version 1)<br>Lead Ethics Reviewer Amendment Approval | All versions |

External Documentation

*- not entered -*

## Section G: Declaration

Signed by:
Areej Alokaili
Date signed:
Fri 14 July 2017 at 22:49

## Offical notes

Additional info sheet uploaded 20/06/2018
'infoSheetAfterMay.docx. in line with GDPR.
---

Amendment 11/07/2019
We're planning to move the experiments carried out under this ethics approval from the Crowdflower platform to Amazon Mechanical Turk. We'll essentially be just moving to a different crowd souring platform and the change has been effectively forced on us as due to a change in Crowdflower's business model.

Approved by Lead reviewer 11/07/19 with only comment to check that the date is updated/provided on the GDPR compliant info sheet.


**20/06/2018**

***Participant Information Sheet***

This study forms part of a research project. The aim of this information sheet is to allow you to understand the purpose of the project, its aims and your role in it.

New data protection legislation comes into effect across the EU, including the UK on 25 May 2018; this means that we need to provide you with some further information relating to how your personal information will be used and managed within this research project. This is in addition to the details provided within the information sheet that has already been given to you.

The University of Sheffield will act as the Data Controller for this study. This means that the University is responsible for looking after your information and using it properly.

In order to collect and use your personal information as part of this research project, we must have a basis in law to do so. The basis that we are using is that the research is 'a task in the public interest'.

Further information, including details about how and why the University processes your personal information, how we keep your information secure, and your legal rights (including how to complain if you feel that your personal information has not been handled correctly), can be found in the University's Privacy Notice https://www.sheffield.ac.uk/govern/data-protection/privacy/general

Research Project:
The project assesses and compares visual representations of documents. These representations are created from the output of a computer algorithm (called a topic model) which automatically analyses collections of documents to identify the underlying themes contained within them. The aim of the project is to determine which representations provide users with the most information about the contents of document collections.

This project is carried out at the University of Sheffield by Areej Alokaili (areej.okaili@sheffield.ac.uk) under the supervision of Dr Mark Stevenson (mark.stevenson@sheffield.ac.uk). This experiment has been approved by the University of Sheffield's Ethics Committee.

Voluntary Participation:
You will be asked to read and acknowledge a consent form to start the experiment. You can withdraw at any point by simply exiting the task.

Data Confidentiality and Storage:
Your responses are anonymised and will not be personally linked to you. The Primary analysis will take place on Areej Alokaili's computer. Some additional analysis may also take place on Mark Stevenson's computers. No encryption will be used given the nature of the data. We expect the data to be made available for future research projects and will retain copies and we may make it available to others.

# B

# SAMPLE TOPIC LABELS

Table B.1: Additional labelling samples to those presented in Table 6.7.

| | |
|---|---|
| **Topic 6** | church cathedral century english bishop england chapel style abbey early |
| **Gold labels** | cathedral, church architecture, gothic architecture, romanesque architecture, hereford cathedral, priory church |
| **BiGRU** | cathedral of the apostle cathedral |
| **BiLSTM** | holy trinity church |
| **CNN** | cathedral of english church |
| **CNN-GRU** | church of st mary |
| **CNN-LSTM** | holy trinity church |
| **Transformer** | holy trinity church |
| **Topic 7** | party campaign candidate political mccain election vote republican democratic voter |
| **Gold labels** | election, primary election, political party, presidential nominee, independent democrat, candidate, state senator, democratic party (united states), voter turnout, general election |
| **BiGRU*** | party |
| **BiLSTM** | party |
| **CNN** | united republican election |
| **CNN-GRU** | party presidential election |
| **CNN-LSTM** | iberal party party party |
| **Transformer** | united states presidential election in the united states |
| **Topic 8** | stock fund market investor share trading firm investment exchange bond |
| **Gold labels** | financial services, investment fund, stock exchange, investment company, investor, capital market, stock market, investment |
| **BiGRU*** | inc |
| **BiLSTM** | international stock exchange |
| **CNN** | stock market investment fund |
| **CNN-GRU** | deposit |
| **CNN-LSTM** | investment investment |
| **Transformer** | uk |
| **Topic 9** | san los california city angeles arizona mile mexico valley land |
| **Gold labels** | los angeles, southern california, san fernando valley, baja california |
| **BiGRU*** | history of the city |
| **BiLSTM** | san francisco |
| **CNN** | los angeles california los angeles |
| **CNN-GRU** | city san |
| **CNN-LSTM** | los angeles san los angeles |
| **Transformer** | san francisco |

Table B.1: (Continue) Additional labelling samples to those presented in Table 6.7.

| | |
|---|---|
| **Topic 10** | kosovo war nato milosevic albanian serb india refugee yugoslavia force |
| **Gold labels** | kosovo liberation army, yugoslav wars, kosovo serbs, breakup of yugoslavia, kosovo albanians, kosovo, kosovo war |
| **BiGRU*** | russian brazilians |
| **BiLSTM** | movement |
| **CNN** | kosovo |
| **CNN-GRU** | indians india |
| **CNN-LSTM** | war war war |
| **Transformer** | national war of |
| **Topic 11** | government political country leader president power party minister democracy protest |
| **Gold labels** | national unity government , democratic party (united states) , political party, prime minister, politics, government, social democracy, head of state |
| **BiGRU*** | people party |
| **BiLSTM** | people |
| **CNN** | political government of presidential party |
| **CNN-GRU** | party |
| **CNN-LSTM** | ministry of the |
| **Transformer** | democratic party of the united states |
| **Topic 12** | god church jesus christian faith lord christ catholic give prayer |
| **Gold labels** | son of god, catholicism, holy spirit (christianity), baptism, faith, christian theology, sacrament, god the father, christian church, christianity |
| **BiGRU*** | of the of the of the of the of the |
| **BiLSTM** | church of our lady of god |
| **CNN** | church of church of church |
| **CNN-GRU** | church of christ |
| **CNN-LSTM** | christian christian christian christians christian church |
| **Transformer** | god of jesus |
| **Topic 13** | baseball league game player season team home hit play fan |
| **Gold labels** | game |
| **BiGRU*** | baseball league game |
| **BiLSTM** | baseball league |
| **CNN** | baseball league game player season |
| **CNN-GRU** | league |
| **CNN-LSTM** | baseball league |
| **Transformer** | baseball |

# C

# ORDER EFFECT ON BERTSCORE

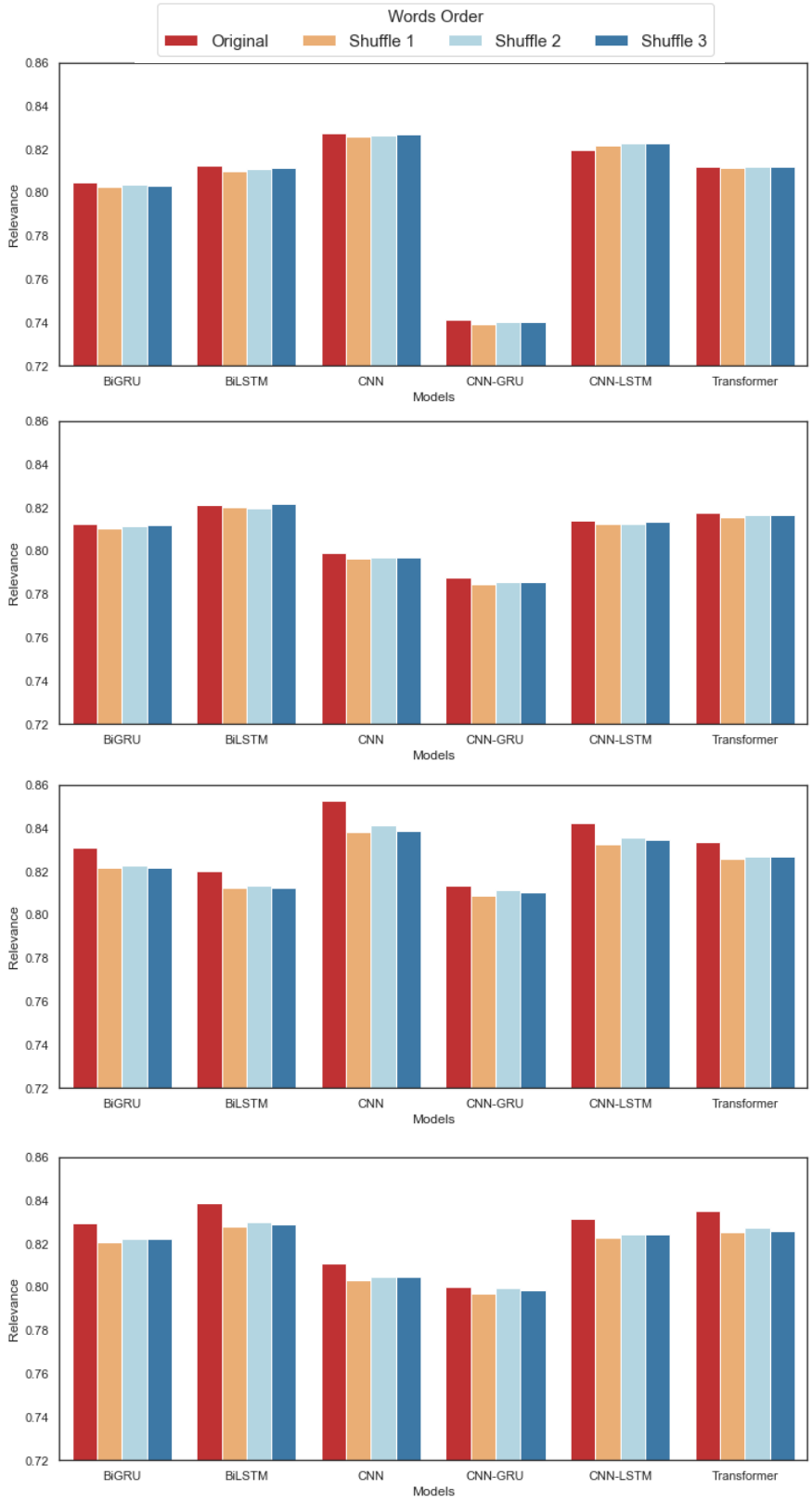



Figure C.1: Bar-plots show the effect of changing the topics' words order on BERTScore, which is used in computing the relevance metric. The relevance metric is described in section 6.4.2.

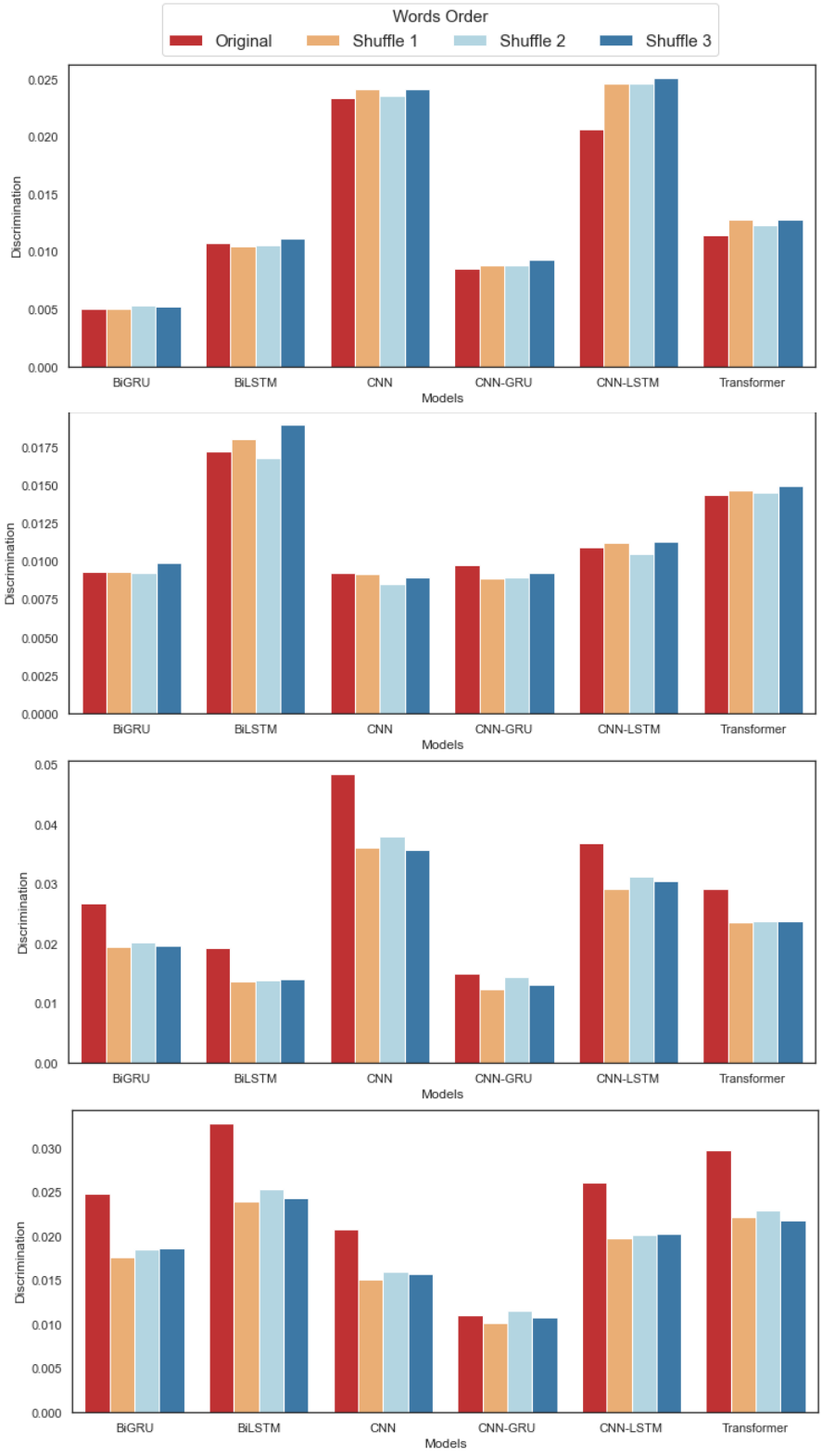



Figure C.2: Bar-plots show the effect of changing the topics' words order on BERTScore, which is used in computing the discrimination metric. The discrimination metric is described in section 6.4.2.