NEURAL TOPIC MODELLING WITH DEEP GENERATIVE MODELS

by **AMIT KUMAR**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Principal Supervisor: Prof. Massimo Piccardi

Co-Supervisor: Dr. Nazanin Esmaili

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

June 26, 2023

Certificate of Authorship/Originality

I, Amit Kumar, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is funding from Food Agility Cooperative Research Centre (CRC) Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaborations between industry, researchers and the community. Moreover, this research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date: June 26, 2023

Abstract

Topic modelling is a popular task of natural language processing (NLP) aimed to automatically discover the main, shared topics of a given collection of documents. In addition, topic modelling is able to determine the topic proportions of each individual document in the collection, which can help with their categorization and organization. Over the years, topic models have found application and proved useful for a broad variety of fields including business, finance, healthcare, education, the media industry, social media, digital agriculture and many others. Like many other applications of NLP and machine learning, in recent times topic models have substantially improved their effectiveness thanks to the integration with deep learning—and deep generative models in particular—which has gained them the collective appellation of neural topic models. However, many improvements are still possible and needed, and this thesis has aimed to make significant contributions in this direction. As a first contribution, we have explored the use of reinforcement learning for refining the training of the models. To this aim, we have proposed novel training objectives based on the policy gradient theorem and contemporary gradient estimators such as REINFORCE with baseline, the Gumbel-Softmax and REBAR. The experimental results over several topic modelling datasets have invariably shown the improved performance of the models. As a second contribution, we have explored how to integrate the powerful, contextualized document representations (i.e., Transformer-based embeddings) in the training objective of the model. This, too, has led to marked performance improvements over probing datasets. Eventually, we have extended the investigation to dynamic topic models, which are models capable of analyzing time-stamped document collections and extracting sets of topics that adapt over

time. For these models, we have proposed a modification of the topic distributions which allows controlling their sparsity, thus adjusting to the characteristics of the collection to be analyzed. Once more, the experimental results have given evidence to the effectiveness of the proposed approach.

Acknowledgements

I am very thankful to some wonderful persons I came across during my whole journey of my PhD. First, I am very grateful to my PhD supervisor Professor Massimo Piccardi and my co-supervisor Dr. Nazanin Esmaili as I would not have succeeded without your support. You all have been a perfect mentor who guided me on each and every step of my PhD career whether it was in my research or non-research matters. I must admit, I enjoyed a great flexibility to explore and pursue my research interests. I owe my research career to both of you. Your valuable perception, brilliant guidance and constructive feedback have made an invaluable difference in the success of my PhD career.

Secondly, I would like to dedicate my success to my family and my friends, my mother, sister and brother, and especially to my wife Kalpana who has been very supportive helping me to elevate moments of cheerfulness and calmness in those days of despair and sitting next to me in those days of ups and downs. Again, I would not have been able to complete this journey without her.

Finally, I would like to thank the Food Agility Co-operative Research Centre Ltd. in selecting me as one of their researchers and financially supporting my research.

Amit Kumar

June 26, 2023

Sydney, Australia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) is a major field of Artificial Intelligence that enables machines to read, understand and derive meaning from human language to some extent. It is also the backbone of an increasingly natural interaction between humans and computers, which are nowadays able to generate meaningful and engaging text (in dialogue systems, virtual assistants, chatbots and so forth). Thanks to the increased availability of data and better algorithms, NLP has fully evolved from a simple keyword- and rule-based technology (the old-fashioned, "mechanical" way) to a sophisticated technology that can deal with the meaning and nuances of human language (the "semantic" way).

One of the most successful applications of NLP to date is known as *topic modelling* [1]. A topic model analyses large collections, or corpora, of documents, and automatically discovers the *topics* of the whole collection and topic proportions of the individual documents. Such an analysis can prove very useful to gain an overall understanding of the contents of a given collection, and also to categorise, group and cluster the individual documents. Topic modelling is a form of unsupervised (and, predominantly, probabilistic) learning since it uses machine learning and NLP techniques to discover the topics from large amounts of unannotated text. A topic is typically represented as a probability distribution over a given vocabulary, assuming that each specific topic would exhibit characteristic frequencies in the use of words

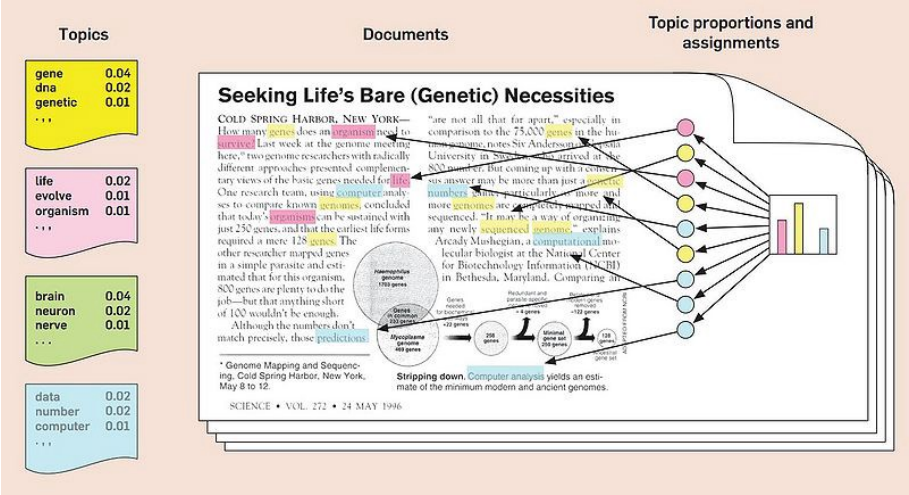

Figure 1.1: An illustration of a probabilistic topic model (from David M. Blei: Probabilistic topic models. Commun. ACM 55(4): 77-84, 2012). The identified topics are on the left (note that they do not have explicit names, but can be named post-hoc) while the parts of the document that come from each topic are highlighted with the corresponding colour.

(for instance, words such as "stumps", "bails" and "wickets" in a topic on cricket). At their turn, documents can be hard-clustered or soft-clustered based on their similarity to the topics. In essence, a topic model can take a possibly huge collection of documents as input, discover all its "template" distributions over the words (i.e. the topics) and simultaneously cluster all the documents into topics based on various notions of similarity. Such an analysis has been validated as useful by many end users in the most diverse fields [2]–[9]. To illustrate the key concepts, Figure 1.1 shows a schematic of topic modelling reproduced from [1]. On the left, one can see the topics, which are probability distributions over each word in the vocabulary. The probabilities have been ranked in descending order, and only the top few words, which characterise each topic, are displayed. Based on such top words, one can also attempt to label each topic with a name. For instance, the topmost topic could be called "genetics", the second could be called "biology" or "life science", and so forth. In the middle, the figure displays one of the documents, where the individual words have been tagged with the topic that they most belong to. Finally, to the right, the figure displays the histogram with the total counts of the words in each topic, i.e. the "topic proportions" for the document.

Topic modelling first came into the picture around 1990 as a text mining and information retrieval technique. Initially, an algorithm called Latent Semantic Analysis (LSA), also equally known as Latent Semantic Indexing (LSI), was presented by Deerwester et al. [10] in 1990. This algorithm clusters a given corpus of documents based on the assumption that documents with similar frequencies of words should be clustered together. As framework, it makes use of a matrix decomposition technique, the singular value decomposition (SVD). Later, another algorithm called Probabilistic Latent Semantic Indexing/analysis (pLSI/pLSA) was proposed by Hoffmann in 1999 [11]. This algorithm introduced probabilistic assumptions in the decomposition, and proved to give better results than LSA/LSI in many cases. In turn, a generalization of pLSA called Latent Dirichlet Allocation (LDA) was introduced by Blei, Ng and Jordan in 2002 [12]. LDA is a probabilistic modelling technique to simultaneously measure the probability distribution of the topics for each document and the probability distributions of the words for each topic under Dirichlet priors assumptions. A huge number of variants have been proposed over the years such as the hierarchical latent tree analysis (HLTA) [13] developed by Liu, Zhang and Chen, where word occurrence using a tree of latent variables is used to discover meaningful topics and form soft clusters of documents. In more recent years, *neural* topic models have come onto the scene by combining deep neural networks and LDA. Deep generative models such as the Variational Autoencoder (VAE) [14][15] [16] have proved to be promising for the automatic discovery of the latent structure in the corpus. Given the tremendous momentum still experienced by deep learning, the area of neural topic modelling is in continuous evolution.

1.1 Research Objectives and Questions

Topic modelling is a mature technology and topic models have helped both researchers and data scientists to automatically extract useful information from unstructured textual data. They have been successfully applied in many domains such as finance, health, social media, agriculture etc. However, one can argue that there are still some standing limitations in this technology. One, for instance, is that con-

ventional topic models struggle to usefully model short documents such as personal messages and social media posts. Due to their general assumptions over the nature of the documents, conventional models reportedly tend to identify too many topics per document. Another limitation is that, in principle, updating the models to new data requires learning them afresh from scratch. This is certainly not suitable for massive-size corpora that receive continuous updates such as, for instance, collections of social media posts. A last limitation that we highlight is that most existing topic models are simply trained with a so-called maximum-likelihood objective. While this is effective to an extent, it misses on the features provided by other learning approaches such as reinforcement learning and exploration-exploitation trade-offs.

For these and other reasons, my thesis focuses on topic modelling and aims to provide significant improvements to existing, state-of-the-art neural topic models. As corpora, I have leveraged both widespread benchmarks such as the 20 Newsgroups and Wiki20K datasets, and more recent collections such as the CORD-19 dataset [17]. In addition, since my doctoral stipend has been provided by the Food Agility Cooperative Research Centre (FACRC), I have also partially explored the agrifood domain using the Amazon Fine Food Reviews dataset [18]. As a potential future application, all the developed models could be used to identify the main topics in agrifood by periodically browsing social media posts in chosen geographical areas. In addition, they could be used to provide a timely discovery of "new" trends, either by comparing topic models at different points in time or by using a dynamic topic model such as that we propose in Chapter 7. The sudden emergence of new trends in agrifood is not uncommon: for example, the demand for camel milk spiked suddenly in the EMEA countries around 2016-2017, and the market value of the "keto" diet has grown by more than a billion dollars in the last five years. As such, we speculate that the timely discovery of emerging food topics by topic modelling may be able to provide the partners of the FACRC and the Australian agrifood industry with a novel, AI-based competitive edge.

Research Questions

The research questions that I have addressed in my thesis are:

1. Can we improve the performance of neural topic models by suitably leveraging the framework of reinforcement learning? (mainly, Chapters 3 and 5)

2. Can we improve the performance of neural topic models by suitably controlling aspects of the model such as the sparsity of the topics and the topic proportions? (mainly, Chapters 4 and 7)

3. Can we improve the performance of neural topic models by better leveraging existing resources such as pre-trained language models? (mainly, Chapters 6)

The justification for my first research question mainly lies in the great success that reinforcement learning has experienced as a framework for improving the performance of machine learning models, including in NLP. The key strengths of reinforcement learning are its ability to leverage both differentiable and non-differentiable "rewards" to guide the training of the models, jointly with its use of sampling to increase the exploration of the parameter space. Despite the richness of the topic model literature, the field had made limited use of reinforcement learning and my research has aimed to fill this gap.

The justification for my second research question is that some aspects of the topic models, such as the higher or lower degree of sparsity of both the shared topic distributions and the topic proportions of the individual documents, are likely to have a substantial influence on the models' final performance. For this reason, throughout my experiments I have incorporated "temperature" parameters in the models to control the sparsity trade-off, and assessed their impact on the model's performance.

Eventually, my last research question stems from the impact that pre-trained language models have had in virtually every other field of NLP. To date, topics models have mostly leveraged the simple "bag-of-words" representation to represent the individual documents, while pre-trained language models permit alternative, much richer, "contextualized" representations that may be able to lead to significant per-

formance improvements.

Overall, I believe that the experimental results presented in this thesis answer all these questions in the affirmative.

1.2 Publications

The following are the publications completed during my PhD:

Journals:

- Topic-document inference with the Gumbel-Softmax distribution: Amit Kumar, N. Esmaili, M. Piccardi, IEEE Access (IF 4.640), vol. 9, pp. 1313-1320, 2021, doi: 10.1109/ACCESS.2020.3046607
- Neural Topic Model Training with the REBAR Gradient Estimator: Amit Kumar, N. Esmaili, M. Piccardi, The ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 21, no. 5, pp. 1-18, 2022, doi: 10.1145/3517336
- The Contextualized Regressive Topic Model: Amit Kumar, N. Esmaili, M. Piccardi, *to be submitted* to Computer Speech and Language, Elsevier (planned for February 2023)

Conference proceedings:

- A REINFORCEd Variational Autoencoder Topic Model: Amit Kumar, N. Esmaili, M. Piccardi, Proceedings of the 28th International Conference on Neural Information Processing (ICONIP 2021), CCIS vol. 1516, pp. 360–369, doi: 202110.1007/978-3-030-92307-5_42
- A Temperature-Modified Dynamic Embedded Topic Model: Amit Kumar, N. Esmaili, M. Piccardi, Proceedings of the 20th Australasian Data Mining Conference (AusDM 2022), CCIS vol. 1741, pp. 15-27, 2022, doi: 10.1007/978-981-19-8746-5_2

All the authors of the above publications have agreed that I am to be recognized as

their main author.

1.3 Thesis Structure

CHAPTER 1 provides an introduction to my work and presents my research objectives and questions, my publications and the thesis structure (this section).

CHAPTER 2 starts with a description of the topic modelling task and the key concepts of a topic model. This is followed by a description of the data and the preprocessing steps which are required to prepare the input for the model. The chapter continues with a presentation of the main topic models, with focus on probabilistic and neural approaches. This includes a review of Latent Semantic Indexing (LSI), probabilistic Latent Semantic Indexing (pLSI), Latent Dirichlet Allocation (LDA), and autoencoder-based neural topic models. To facilitate the comprehension, the chapter also briefly reviews deep generative models such as the Variational Autoencoder (VAE) and Generative Adversarial Networks (GANs). Finally, the datasets and the evaluation metrics used in the experiments are presented.

CHAPTER 3 introduces a neural topic model integrating a variational autoencoder (VAE) topic model and the REINFORCE gradient estimator. This unit of research leverages the neural topic model proposed by Srivastava and Sutton [14] which augmented LDA with a variational autoencoder and established state-of-the-art performance for the field. However, at the beginning of our research we noted that the field of topic modelling had made very limited use of the framework of *reinforcement learning*, which had instead proved beneficial for so many other fields. For this reason, in this chapter we propose a topic model that uses the policy gradient theorem and the REINFORCE algorithm [19] to learn an effective *policy* over the topics. Extensive experimental results show that the proposed model has been able to achieve a marked performance improvement.

In a similar vein, CHAPTER 4 integrates the Gumbel-Softmax distribution in a VAE topic model. In specific, we propose modelling the topic proportions of the individual documents using the Gumbel-Softmax distribution [20], [21] which is a

soft alternative to a standard categorical distribution. By using Gumbel-Softmax samples instead of categorical samples we can 1) further diversify the model during training and inference, and 2) influence the sparsity of the topic proportions by a hyperparameter called the pseudo-temperature which allows us to control the expected number of the topics for each document. In our model, the Gumbel-Softmax is integrated into a state-of-the-art topic model, the autoencoding variational inference for topic models (AVITM) of Srivastava and Sutton, outperforming its baseline in all metrics.

To explore reinforcement learning further, CHAPTER 5 integrates the recently-proposed REBAR gradient estimator in a VAE topic model. The approach is similar to that of the previous chapter, but is able to amend an intrinsic limitation of the Gumbel-Softmax, i.e. the biasedness of the gradient estimator. To remove the bias, in this chapter we leverage the REBAR gradient estimator [22], which is both unbiased and low-variance by design. The estimator is integrated in the state-of-the-art deep variational-autoencoder topic model of Srivastava and Sutton [14], once again displaying remarkable performance.

The conclusion of this chapter also briefly compares the three approaches proposed across chapters 3-5.

CHAPTER 6 takes the research in a different direction: exploring the potential of contextualized representations such as BERT embeddings for topic modelling. To this aim, the chapter presents a novel, contextualized regressive topic model which exploits BERT embeddings in the training objective. The experimental results show that this model has been able to outperform its strong baseline.

Finally, CHAPTER 7 explores extensions of topic modelling to corpora of timestamped documents. These extensions are known as *dynamic topic models* since they are able to extract *sequences of topics* from the successive timestamps, giving an idea of the temporal evolution of the topics in the corpus. In specific, the chapter presents the Dynamic Embedded Topic Model with Temperature (DETM-tau), which is an augmentation of the Dynamic Embedded Topic Model of Dieng et al. [23] with a temperature hyperparameter controlling the sharpness/smoothness

trade-off of the word distributions. The experimental results show that the proposed model has been able to achieve a remarkable performance over challenging datasets of timestamped documents.

For clarity, chapters 3 to 7 have been reproduced from the corresponding publications with minimal rewording. This means that some of the contents are repeated across the chapters, but we have preferred to keep them self-contained to permit reading in any order. The conclusion of chapter 5 also contains an unpublished table comparing the performance of models across chapters 3 to 5.

The thesis is eventually concluded by conclusions derived from the work conducted and suggestions of possible future extensions.

Chapter 2

Literature Review and Background

In this chapter, I concisely review all the main topics that form the background for my research. The chapter opens with a high-level framing of machine learning and then introduces the concept of topic model. After a review of the standard data preparation steps, it then covers key topic models such as LSI, pLSI and LDA. More advanced topic models based on variational autoencoders and generative adversarial networks are described next. Eventually, the chapter concludes with a review of popular datasets and evaluation metrics.

2.1 Machine Learning

Machine learning is a field of Artificial Intelligence (AI) concerned with automatically learning from and finding patterns in data [24]. Machine learning then uses the unhidden patterns to make predictions over future data and support decision processes, even under uncertainty. In contrast with conventional programming where data and rules are manually defined, machine learning uses data and algorithms to automatically infer sets of rules that work well for a chosen task. In the real world, the data generated from a variety of sources are predominantly unstructured (i.e., they don't fit neatly and spontaneously into organized databases), "messy",

and hard to manipulate. Yet, such data are said to represent approximately 80 percent of the data available in the world. Therefore, machine learning offers a natural fit to help identify structure in such data. As a learning approach, it can be broadly subdivided into three main styles: supervised learning, unsupervised learning and semi-supervised learning. In supervised learning the goal is to learn a predictive model based on a given training set of annotated input-output examples. In unsupervised learning the goal is to identify patterns and trends based solely on unannotated data. Semi-supervised learning is an intermediate case where the goal is to learn a model from a limited amount of annotated input-output examples alongside lots of other unannotated data. All these three styles have found extensive application in research and professional practice. Given that the scope of my thesis is to derive "topic models" from large collections of documents, unsupervised learning is its case of reference.

2.2 Topic Models

The motivation for topic modelling comes from the more general notion of *text mining* which refers to any process aimed to extract meaningful information from documents. Typically, text mining techniques are subdivided over the fields of Information Retrieval (IR) and Natural Language Processing (NLP), but it is common to have applications spanning both fields. In turn, NLP encompasses many, diverse tasks such as text classification, sentiment analysis, part-of-speech tagging, chunking, named-entity recognition, relation extraction, and many more all the way up to high-level tasks such as automated question answering and conversational agents. Among them, my thesis has focussed on topic modelling.

A topic model is an unsupervised machine learning approach applied heavily in NLP that parses and analyzes a document collection in order to 1) extract its main, shared topics and 2) map each individual document to the extracted topics. This can be useful for two fundamental reasons: to gain a quick, synoptic understanding of the entire collection; and to categorize and organize the individual documents according to their topic proportions. In order to provide this information, topic

models often make simplifying assumptions in the representation of the documents: for instance, they ignore the order of the documents' words as it is often irrelevant to the determination of the topic; likewise, they dismiss the tense of verbs, the number of nouns (singular vs plural) and so forth. The input to a topic model is typically a so-called *term-document matrix* which is a matrix where the rows correspond to the distinct words of a chosen vocabulary and the columns correspond to the documents in the given collection. Each element of this matrix typically just stores the count of the occurrences of a particular word in a specific document. For instance, if word "cat" has row index i and appears three times in the document of column index j, $\text{matrix}(i, j) = 3$. This representation for the individual documents is also known as "bag-of-words" (BoW) where "bag" refers to the dismissal of the word order information. Several variants also exist which mainly leverage weighting and normalization to emphasize the "informativeness" of specific words. Among them, the most popular is undoubtedly TF-IDF (term frequency/inverse document frequency) [25]. Often, the vocabulary used for the matrix is simply the list of the unique words appearing in the given collection, but it can also be externally provided. In general, the vocabulary has to abide by some size constraint to limit the computational complexity, so the least frequent terms may be omitted. Likewise, at times the most frequent terms are omitted if they appear to be overly generic and uninformative (unless preprocessing has already removed them). In terms of models, really many different topic models have been proposed to date and it is challenging to hint at them all. A first cut could be distinguishing between non-probabilistic and probabilistic topic models. Non-probabilistic topic models were the first to appear in the early 1990s and are mainly based on various matrix factorization approaches; champions are Latent Semantic Analysis (LSA) (also known as Latent Semantic *Indexing* (LSI) when employed for information retrieval) and Non-negative Matrix Factorization (NMF). Conversely, probabilistic topic models have appeared later to provide a more principled and versatile underpinning to the models. The most famous probabilistic topic models include Probabilistic LSI (pLSI) and Latent Dirichlet Allocation (LDA); and, in more recent years, all those based on the so-called deep *generative* models such as variational autoencoders (VAEs)

and generative adversarial networks (GANs).

To recap, the typical assumptions made by a large majority of the existing topic models are:

- A document collection with D documents is provided.

- Given a vocabulary of size V, we pre-compute the bag-of-word (BoW) representation of each document.

- We concatenate all such BoWs into a large V x D matrix (the *term-document matrix*).

- V is typically in the 2K-50K range, while D can go from a few thousand to a million or more.

2.3 Data Preparation

The overall aim of data preparation is to create a representation for the documents which can enable effective and efficient topic modelling. In this process, we typically aim to remove all the textual elements that do not convey topical information, while at the same time amending the noise, errors and missing values that typically affect real-world documents. As such, data preparation plays a key role in removing undesirable words, headers, footers, symbols, punctuation, suffixes, stopwords etc, and eventually convert the document into a suitable representation such as BoW or TF-IDF. Hereafter, we present a brief review of the most common text preprocessing and feature extraction techniques employed in the field.

Tokenization is the process of splitting a paragraph or sentence into words, characters or some meaningful text units known as tokens and is the founding step of virtually any NLP tasks. For example, a sentence such as "Topic modelling is one of

the most widely known techniques in NLP!" may be tokenized as: "Topic", "modelling"," is", "one", "of", "the", "most", "widely","known", "techniques", "in", "NLP", "!".

Normalization is the process of converting the tokens to some "standardized" form to reduce the size of the token space while retaining meaning and semantics. This helps both the effectiveness and the efficiency of the modelling stage. Using linguistic terminology, normalization can be defined as the process of converting a token to its base form by removing all its inflectional elements. *Stemming* and *lemmatization* are the two most common components of normalization.

Stemming is a rule-based process for removing inflection from a given token. For example, words such as playing, played, and plays will all become play after the stemming process. This can be useful for topic modelling since it can drastically reduce the size of the vocabulary while aggregating all the counts of words of equivalent topical value. However, stemming has at times to be used carefully to not introduce undesirable artifacts. As a paradox, if all ending "s" characters were to be blindly removed, a word such as "his" would turn into "hi" with a completely different meaning. Fortunately, robust tokenizers exist for English and many other language families.

Lemmatization is the process of removing the suffix of a given word to reduce it to its base form, known as lemma. For example, a verbal form such as "went" will be turned into "go". This is similar to stemming, but somehow more general. A lemmatization algorithm typically makes use of word structure, grammar, vocabulary and part-of-speech tagging. This allows it to distinguish it between cases such as the word 'working' used as a verb, which will will be turned into "work", and word 'working' used as a noun, which will remain unchanged.

Stopword removal consists of removing commonly occurring words such as "and", "is", "an", "the"," are" and so forth which are assumed to convey little or no meaning in topic modelling. All the popular NLP libraries such as NLTK or spaCy contain comprehensive built-in lists of stopwords; however, the list of stopwords can also be customized at will.

Document representation is the process of representing an individual document in a vector format so that it can be understandable by the machine. Popular representations used in topic models are the bag-of-words (BoW) and TF-IDF. The BoW representation only stores the number of occurrences of each word in each document, dismissing the sequential order. The intuition behind a simple representation such as BoW is that documents containing similar word counts are indeed similar in content. To discount the impact of the length of the document, the BoW counts can also be normalized by dividing them by the total number of the words in the given document, turning them into term frequencies (TFs). An alternative representation is TF-IDF which is formed by multiplying the TF (term frequency) by the IDF (inverse document frequency). The IDF calculates the importance of a word by checking in how many documents it appears. If the word appears in many documents, it is assumed to be uninformative and is assigned a low IDF. Conversely, if it appears in only one or a few documents it is assumed to be informative and is assigned a high IDF. Quantitatively, the IDF is defined as the logarithm of the ratio between the total number of the documents and the number of the documents in which the word appears at least once (NB: many variants have been proposed).

2.4 Latent Semantic Analysis/Indexing

Latent Semantic Analysis (LSA) or, interchangeably, Indexing (LSI) [10] is often cited as the first, actual topic model. This model learns the hidden topics by carrying out a matrix decomposition on the term-document matrix with the singular value decomposition (SVD). SVD is a dimensionality reduction technique which factorizes the input matrix in an approximate way to reduce the number of the columns while maintaining the similarity structure among the rows (or vice versa, depending on the input shape). To better understand this factorization, let us introduce the following notations: K is the number of topics in the whole corpus, V is the size of the vocabulary, D is the number of documents in the corpus, and W is the term-document matrix of size V x D. With these notations, the LSI factorization can be expressed as:

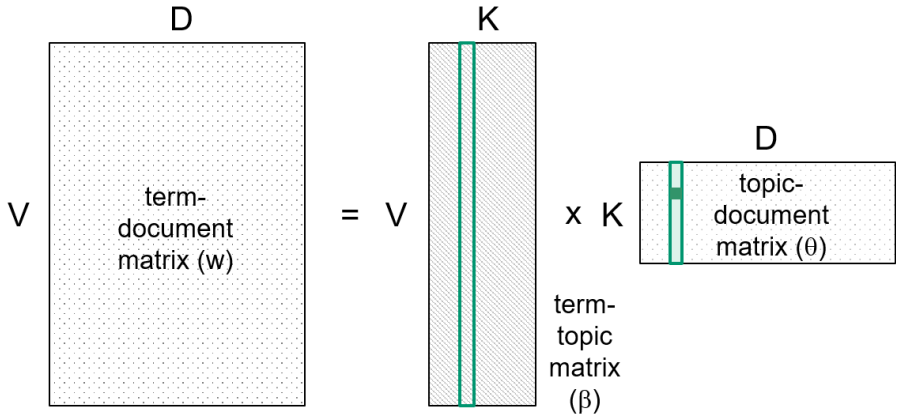

Figure 2.1: LSA as matrix factorization.

$$W \approx \beta\theta \tag{2.1}$$

where β represents the *term-topic matrix* of size V x K and θ represents the *topic-document matrix* of size K x D. Fig. 2.1 depicts the factorization. With this factorization, each column of matrix β represents one topic as a distinctive set of weights for each word in the vocabulary, while each column of matrix θ represents one document as a distinctive set of weights for each topic. Lastly, each row of matrix β can be seen as a low-dimensional "embedding" of a vocabulary word.

It is worth noting that the representations obtained from a topic model also permit measurements of similarity between any two documents, topics and words. To quantify the similarity one can use, for instance, the cosine similarity, where values close to 0 denote very different vectors while values close to 1 denote high similarity. This can be useful for comparisons, information retrieval, word and topic embedding, and several other tasks.

2.5 Probabilistic Latent Semantic Analysis

The earliest topic models (LSA/LSI, NMF etc) made use of classic matrix decomposition approaches, such as singular value decomposition and non-negative matrix factorization, to identify the hidden topics. However, these approaches were not able

to derive proper probability distributions that could, for instance, be composed by Bayes' theorem, marginalized or sampled. For this reason, the further development of this field has seen a dominance of probabilistic approaches. Probabilistic topic model are based on the following assumptions:

1. Each document is generated from an underlying distribution over the set of the latent topics.

2. In turn, the words in the document are generated from the topics, which are distributions over the words in the given vocabulary.

These two assumptions are also called a "generative model", in the sense that, given the parameters for all the involved distributions, they could be used to sample new, synthetic "documents" (i.e., their BoWs) that abide by those distributions. This is obviously different from what we are interested in real life, that is, given a collection of actual documents, to infer the optimal parameters for the distributions. However, it is a very useful formalization of the model. In terms of matrix representations, the basic idea is to generate the term-document matrix from a probabilistic model with latent topics such that for any document d and vocabulary word w, $p(w|d)$ is an element in the matrix. In turn, an observed term-document matrix is the basis for estimating the model's probability distributions. A probabilistic version of LSA, known as probabilistic latent semantic analysis (pLSA, or pLSI), was proposed by [11] to improve the LSA model. Compared to LSA, pLSA introduces the constraint that the elements of β and θ must each be $>= 0$ and $<= 1$ as for the probabilities of a categorical distribution, and that each column must add up to 1 (a constraint also known as the simplex constraint). By noting an index on the topics as z, it is easy to see that the columns of β can now be noted as $p(w|z)$ (the probability of the words in the vocabulary for a given topic) while the columns of θ can be noted as $p(z|d)$ (the probability of the topics for a given document).

Given the notations already introduced, pLSA can be expressed as:

$$p(w|d) = \sum_{z=1}^{K} p(w|z)p(z|d) \quad w = 1 \dots V, d = 1 \dots D \tag{2.2}$$

where $p(w|d)$ is the probability of word w in document d, $p(w|z)$ is the probability of word w in topic z, and $p(z|d)$ is the probability of topic z in document d. The equality is obtained by applying Bayes' rule and marginalizing variable z. A prior over the document index, $p(d)$, can also be added.

2.6 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [12] is a generalized version of pLSA that adds prior probabilities (in the form of Dirichlet distributions) to the columns of θ (the topic proportions per document) and optionally to the columns of β (the topics). LDA is widely regarded as the reference model in the field of topic modeling. We briefly describe it hereafter, also introducing the following notations:

- With $w_{d,n}$ we denote the n-th word in the d-th document in the given corpus. We use the term "word" to refer to a categorical value in a chosen vocabulary of size V (NB: the "surface form" of the word, i.e., its string of characters, is irrelevant).

- With w_d we note the set of all the words in document d.

- Each word, $w_{d,n}$, is assigned to a corresponding *topic*, $z_{d,n}$. Also the topics are categorical variables, and we note the set of their possible values simply as indexes $1 \dots K$.

LDA's main distributional assumptions are:

- The topic variables for a given document d are independently and identically distributed (i.i.d.) according to a multinomial distribution of parameter vector θ_d, $\text{Mult}(z_{d,n}|\theta_d)$.

- In turn, the parameter vector, θ_d, of the multinomial distribution is distributed according to a Dirichlet prior, $\text{Dir}(\theta_d|\alpha)$, parametrized by a K-dimensional integer vector, α, shared by the entire corpus.

- The model also includes a set of K multinomial distributions over the vocabu-

lary, one per topic. Each such distribution is parametrized by a V-dimensional probability vector, noted as $\beta_k, k \in [1 \dots K]$.

- Each word in a given document is distributed according to one of these distributions, indexed by its topic variable, as in $w_{d,n} \sim \text{Mult}(w_{d,n}|\beta_{z_{d,n}})$.

Variable $z_{d,n}$ can be marginalized analytically since both $w_{d,n}$ and $z_{d,n}$ are multinomially distributed. This allows us to rewrite the probability of $w_{d,n}$ as:

$$w_{d,n} \sim \text{Mult}(w_{d,n}|\beta\theta_d) \tag{2.3}$$

where matrix $\beta = [\beta_1 \dots \beta_K]$ is $V \times K$, and vectors θ_d and $\beta\theta_d$ are $K \times 1$. We can then express the joint probability of word $w_{d,n}$ and its topic vector, θ_d, as:

$$p(w_{d,n}, \theta_d|\alpha, \beta) = \text{Mult}(w_{d,n}|\beta\theta_d)\text{Dir}(\theta_d|\alpha), \tag{2.4}$$

and the probability of all the N words in document d and their topic vector as:

$$p(w_d, \theta_d|\alpha, \beta) = \prod_{n=1}^{N} p(w_{d,n}, \theta_d|\alpha, \beta) \tag{2.5}$$

The training goal for this model is to estimate θ_d, α and β that maximize (2.5) over the given collection. Typical training algorithms leverage variational approximations and Markov chain Monte Carlo [12]. Once the model is trained, the topic vectors for any given new document can be inferred by keeping parameters α and β fixed.

As said in the opening, LDA is widely regarded as the reference model for the field and as such has also been used as the base to build a large number of extensions and variants. A non-exhaustive list of these extensions include: class-supervised versions [26], sparse versions [27]–[31], sequential versions [32], hierarchical versions [13], [33], [34], and many more.

2.6.1 Parameter estimation in LDA

Parameter estimation in LDA can be performed with different approaches. Since in this thesis we limit ourselves to using existing approaches, we only briefly sketch the main in the following.

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm [35] and is one of the approaches that can be used to learn the parameters of this model. In a nutshell, it is a method for generating samples from a complex joint distribution when variables are actually sampled from conditional distributions. It runs through every document and assigns each word in the document to one of the K topics. Such an assignment provides a topic representation for all the documents and for the word distributions of all the topics. This initial representation is then improved upon iteratively.

Expectation-maximization (EM) is another approach used for estimating parameters when the model depends on latent variables. It is used to find the argmax of the model (i.e. the best parameters in a maximum-likelihood sense) when they cannot be found analytically in closed form. Calculating a likelihood solution involves taking a lower bound, then finding its gradient and optimizing upon it. EM consists of two steps: in the first step (E step) the expectation of the log likelihood is estimated based on the last parameters known, and in the second step (M step) the parameters are optimized.

Variational inference [36] is the third approach used to approximate intractable integrals used for intricate statistical models consisting of given observed data and unobserved variables (unidentified parameters and latent variables). It mainly serves two objectives: 1) to provide an approximate solution to the posterior probability of the unobserved variables for inference purpose, and 2) to come up with a lower bound for the marginal likelihood of the given data. It serves as an alternative to MCMC algorithms and, like them, it leverages a fully Bayesian approach.

2.7 Overview of Deep Generative Models

Generative models are a very useful approach to understand the latent structure of observations such as images, text and audio. A generative model is a statistical model consisting of joint probability distribution $p(x, y)$ given observation x and target variable y. The name "generative" comes from the fact that one can obtain samples of the observations by sampling this model, and is contrasted to "discriminative" models such as $p(y|x)$ that cannot be used to sample x (because they treat x only as a conditioning input and do not model its probability). With the popularity of deep learning, a new wave of generative models has been proposed under the collective name of *deep generative models* (DGMs), which are basically combinations of deep neural networks and generative models. The most popular DGMs are the Variational Autoencoders (VAEs) [15] and the Generative Adversarial Networks (GANs) [37]. These DGMs have proved effective at discovering and learning the hidden structure and patterns of the data by generating samples from the learned observation distributions.

In particular, VAEs supersede some of the drawbacks of other deep models in learning to approximate and maximize the log likelihood of the observations. A VAE is a generalization of the basic autoencoder that removes some of its limitations by modelling the probability distribution of the hidden variables; say, z. In turn, a basic autoencoder consists of two networks: first, the encoder, which takes in input an observation and produces in output the latent variable z through some constraint such as dimensionality reduction; second, the decoder which tries to reconstruct the input as exactly as possible from z. A VAE extends these two networks in a probabilistic sense. In the following, we describe it in greater detail.

2.8 The Variational Autoencoder

2.8.1 Motivation for generative models

The past decade has seen some remarkable discoveries in the area of machine learning, and generative and discriminative models have been the subject of much re-

search in these recent years. While a discriminative model is based on learning the conditional distribution of the target variables given the observed data, a generative model is based on learning the joint distribution of the observed and target variables. Generative models try to simulate data close to the real data and have been attractive in a variety of ways. First, it is not too difficult to model the observed variables and all other unobserved variables in a probabilistic manner. Secondly, these models may be used to explore causal relationships allowing some generalization to new situations. Thirdly, in semi-supervised settings, generative models can be used to improve the accuracy of classification by generating a number of "virtual" samples that can help with the training process (Kingma et al., 2014 [38]). As a natural evolution, the advent of deep learning has made generative models "deep", taking advantage of the increasing model complexity, training data size, and computational capabilities. In research and also the commercial space, such deep generative models had initially been proposed and used for processing images, including generating new synthetic images, compressing images, looking for new image representations and more so. Later, they have been extended and extensively applied also to text data, including topic modelling. For this reason, we review two key deep generative models (VAE and GAN) hereafter.

2.8.2 Autoencoder

An autoencoder (AE) consists of a neural network capable of learning representations for data compression in an unsupervised manner. Basically, the encoder compresses the input data by some dimensionality reduction technique, while the decoder later tries to decompress the compressed data to be as close as possible to the input. In other words, an autoencoder is basically the combination of two networks concatenated with each other, with a bottleneck where the input, x, goes through the encoder to become a latent representation, z, and then an $\hat{x}$ vector is reconstructed from z by the decoder. The loss function to minimize is the norm $|x-\hat{x}|$, or a more general objective function $L = |x-\hat{x}| + \text{regularizer}$. The rationale for using an autoencoder in the first place is that the reconstructed vector, $\hat{x}$, or even the latent representation, z, can prove more effective than the original input

in downstream tasks of pattern recognition. As a necessary condition, the input data are expected to possess some "structure" (i.e. internal correlation), otherwise it would be challenging to compress them and later reconstruct them to be similar to the input.

Adding more layers to an autoencoder makes it a *deep* autoencoder. Many other variations to the basic autoencoder exist including, amongst others, the sparse autoencoder, the denoising autoencoder, the contractive autoencoder, and the variational autoencoder. A sparse autoencoder is a type of autoencoder often having more hidden nodes than the input and capable of learning representations that boost sparsity, allowing only a small number of hidden units to be active at a particular instant. The loss function is built to penalize a hidden layer if this activates more than a few units. This leads to a form of regularization where the weights of the network are regularized rather than the activation functions.

A different variation of the basic autoencoder is the denoising autoencoder. Here some noisy data are first fed as corrupted input to the encoder, then pass through the bottleneck, and eventually the decoder tries to obtain back the initial, undistorted data. A denoising autoencoder is practically used for cleaning noisy data fed as input. It does this by deriving the features that represent a "reliable structure" in the distribution, thus trying to achieve an improved representation of the data.

An autoencoder that is robust to small changes of the input data is called a contractive autoencoder. This feature is obtained by adding a regularizer to the objective function, forcing the model to learn a representation that does not change by minor changes in the input. Here the model is trained to learn a contractive representation during training, since the regularizer is only applied on the training data.

Eventually, the variational autoencoder is explained in detail in the next subsection.

2.8.3 Variational autoencoder

The motivation for the variational autoencoder [15] stems from the standard autoencoder, but in this case the encoder and decoder map probability distributions

rather than deterministic values. VAEs are probabilistic graphical models where the parameters of the distributions of the latent variables are obtained using deep neural networks. This feature makes VAEs belong to the class of the *deep generative models*, joining the properties of graphical models and deep learning architectures. These deep generative models significantly differ from other popular deep networks such as CNNs, RNNs and so forth which are all, in a sense, standard discriminative classifiers.

2.8.4 Statistical motivation

A variational autoencoder is a combination of graphical models and neural networks, and learning to approximate the latent variables is done using variational inference. The variational autoencoder needs to sample the latent variable, z, conditional to observation x as its encoder step. This requires modelling probability $p(z|x)$ which can be expressed as:

$$p(z|x) = \frac{p(z,x)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz} \tag{2.6}$$

Here, x denotes the observed variable while z denotes the hidden variable for which we aim to model the distribution. Unfortunately, computing the $p(x)$ term is challenging as it is a marginal distribution ($\int p(x|z)p(z)dz$) and solving its integral is intractable in many cases. However, it can be approximated by Monte Carlo sampling (an unbiased estimate with high variance) or variational inference (a biased estimate with low variance). To this aim, distribution $p(z|x)$ is replaced by another distribution, $q_\phi(z|x)$, which attempts to generate samples that can well justify the observations, while at the same time remaining close to its prior, $p_\theta(z)$, where ϕ and θ denote the parameters of the encoder and the prior, respectively. Figure 2.2 depicts the main blocks of a VAE.

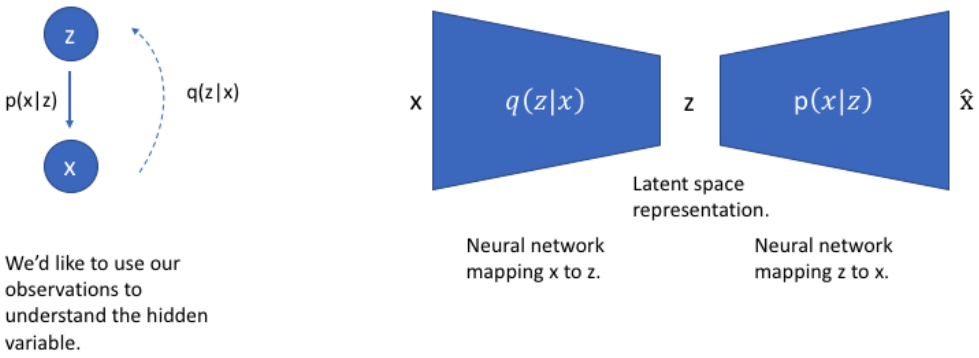

Figure 2.2: Neural network mapping from x to latent space z and back to $\hat{x}$.

2.8.5 Training objective

The training objective of a VAE is to maximize the probability of the observations, $p(x)$. However, as we have said above, this term is normally intractable. Therefore, the approach attempts to maximize a lower bound for $p(x)$. For clarity, a lower bound is a quantity that is guaranteed to always be $\leq$ than the target quantity. This means that if we are able to raise the lower bound up to a certain value, the target quantity will be at least at that same value. In other words, we indirectly raise the value of the target quantity by directly raising the value of its lower bound.

The lower bound is obtained very simply by first applying Bayes' theorem:

$$\begin{aligned} p(x) &= p(x|z)p(z)/p(z|x) \\ \rightarrow \log p(x) &= \log p(x|z) + \log p(z) - \log p(z|x) \end{aligned} \tag{2.7}$$

Then we add and subtract the same quantity, $q(z|x)$ (the encoder distribution), to the right hand side, which leaves the equality unvaried, and regroup the terms:

$$\begin{aligned} \log p(x) &= \log p(x|z) + \log p(z) - \log p(z|x) + \log q(z|x) - \log q(z|x) \\ &= \log p(x|z) + \log \frac{q(z|x)}{p(z|x)} - \log \frac{q(z|x)}{p(z)} \end{aligned} \tag{2.8}$$

Eventually, we compute an expectation of all the terms on the left and right hand sides using $q(z|x)$ as the probability distribution. This, too, leaves the equality unvaried. Before we do, we have to note two things: 1) $\log p(x)$ does not depend on z, and is therefore equal to its own expectation; 2) an expectation of the form $-\mathbb{E}_q[\log \frac{q}{p}]$ is equal to the famous Kullback-Leibler divergence, $D_{\mathrm{KL}}(q\|p)$. Therefore, we obtain the following equality:

$$\log p(x) = \mathbb{E}_{q_\phi(z|x)}[\log p(x|z)] + D_{\mathrm{KL}}(q(z|x)\|p(z|x)) - D_{\mathrm{KL}}(q(z|x)\|p(z)) \tag{2.9}$$

Now, any Kullback-Leibler divergence is provenly always ≥ 0. Thus, if we take away one term from the right hand side, we obtain the following inequality:

$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p(x|z)] - D_{\mathrm{KL}}(q(z|x)\|p(z)) \tag{2.10}$$

The term to the right hand side is the lower bound that we were seeking (the Evidence Lower BOund, or ELBO for short). By making the parameters explicit in the notation, the training objective of the VAE [36] is to maximize the following function:

$$\mathcal{L}(\varphi, \phi, \theta, \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\varphi(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) \tag{2.11}$$

We can, likewise, turn the ELBO into a cost function to be minimized by simply changing the sign to the right hand side. The intuition of equation (2.11) can be given in these terms: the training attempts to 1) find an encoder, $q_\phi(z|x)$, which, at the same time, can provide good samples to the encoder and stay close to its prior, $p_\theta(z)$; and 2) find a decoder, $p_\varphi(x|z)$, that can give high probability to the observations based on the samples received from the encoder. Note that the KL divergence is a measure of the difference between two distributions ($q_\phi(z|x)$ and

$p_\theta(z)$ here) and as such is one of the terms of the minimization. In addition, we remark that the prior *is trainable* together with the encoder and decoder: training it simply means finding the best parameters for a probability distribution over z which is independent of x (i.e. does not have x in input) over the entire training set.

2.8.6 VAE as a topic model

A VAE can be easily adapted to become a topic model. While we provide full details in the following chapters 3-7, here we highlight the main assumptions: variable x will map the observation for a given document (i.e., its BoW) and variable z will map its topic proportions. The overall model expresses the ability of the VAE to both 1) extract a good set of topics for the entire collection and 2) assign a good vector of topic proportions to the given document, such that the document can be closely reconstructed from them. The actual models vary in the distributional assumptions for the encoder, decoder and prior, and, possibly, a number of other assumptions and approximations.

2.9 The Generative Adversarial Network

A generative adversarial network (GAN) is a deep generative model consisting of two modules: a generator and a discriminator, generally implemented by neural networks. The generator neural network attempts to generate synthetic samples which are as similar as possible to the distribution of the true examples, and could be confused with them. The discriminator neural network typically consists of a binary classifier, trying to discriminate the generated samples from the true samples as accurately as possible. Further, a better generator and discriminator can be obtained by training them against each other. The optimization of GANs is a minimax optimization problem that terminates at a saddle point, minimizing when updating the generator and maximizing when updating the discriminator. The ultimate goal is to reach the Nash equilibrium [39]. If this happens, the generator can be regarded as having captured the actual distribution of the true samples.

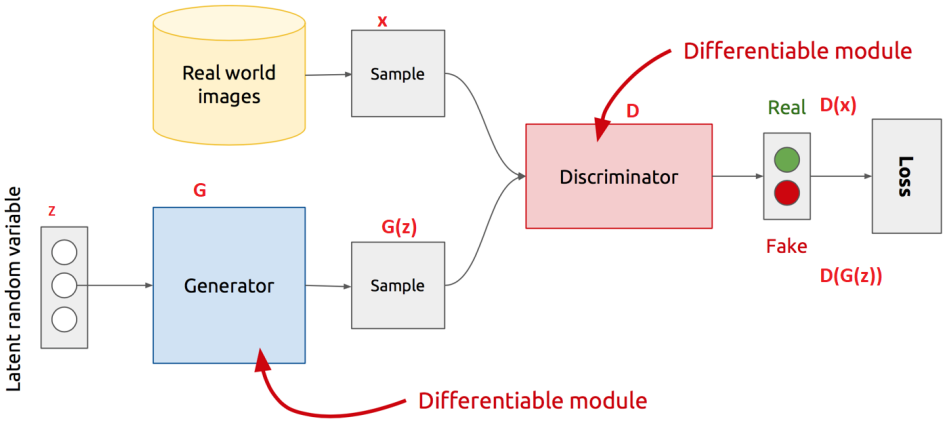

Figure 2.3: A schematic of the GAN with focus on its two neural networks: the generator and the discriminator. The source of this figure is: https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016.

Fig. 2.3 shows a schematic of the GAN. Its main elements are as follows:

- x denotes the real samples.
- z is a latent variable derived from the observations that is used to control the behavior of the generator.
- G is the generator which generates "fake" (i.e., synthetic) samples, noted as $G(z)$, which are as similar as possible in distribution to the real samples.
- D is the discriminator which receives as input both the generated and the real samples and tries to correctly classify both. The probability of the real samples to be true is noted as $D(x)$ (NB: high is desirable), while the probability of the fake samples to be true is noted as $D(G(z))$ (NB: low is desirable).

The general idea of the training of a GAN is that the updates of the generator should make $G(z)$ as close possible to x (in distributional sense), while the updates of the discriminator should increase its ability to tell apart $G(z)$ from x. More details are provided in the following subsection.

2.9.1 Training objective

The training objective of a GAN can be expressed as:

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}[\log(1–D(G(z)))] \tag{2.12}$$

In Equation 2.12, we have introduced the additional notations of $p(x)$ for the distribution of the real samples and $q(z)$ for the distribution of the latent variables that control the generator. The first term in the equation is the log-likelihood of the discriminator's positive class (D) for the real samples (the higher, the better). The second term is the log-likelihood of the discriminator's negative class ($1 - D$) for the fake samples (the higher, the better from the point of view of the discriminator, while vice versa from that of the generator). When training the discriminator's parameters we attempt to maximize $V(D,G)$ (keeping the generator's parameters fixed), while when training the generator's parameters we attempt to minimize $V(D,G)$ (likewise, keeping the discriminator's parameters fixed). These "conflicting objectives" result in a minimax game which is expressed by Equation 2.13:

$$\min_G \max_D V(D,G) \tag{2.13}$$

By now noting with x all the samples (real and fake), the Nash equilibrium of this particular game is achieved when:

$$\begin{aligned} p(x) &= q(x) \quad \forall x \\ D(x) &= 1/2 \quad \forall x \end{aligned} \tag{2.14}$$

The alternation of the minimization and maximization of Equation 2.12 does not always bring the model to a satisfactory equilibrium because $\log(1–D(G(z)))$ may rapidly saturate in the early stage of training, making D easily reject $G(z)$ because of their low quality. The lack of gradient from the discriminator prevents further improvement of G, stranding the model on a poor operating point.

2.9.2 GAN as a topic model

GANs, too, can be easily recast as topic models. An example is the GANTM of [40] where the model behaves as follows: the generator receives in input the topic proportions of an actual document, z, and attempts to generate its BoW representation as output; at its turn, the discriminator tries to assign it a low probability. During training, the generator tries to become better and better at reproducing the BoW vector, while the discriminator tries to remain able to tell apart the generated BoWs from the real ones. The main difference with the VAE is that the latter, too, tries to make the generated samples as similar as possible to the real ones; yet, it does not attempt to discriminate them.

2.10 Interpreting the Topics

It is important to note that the number of topics in a topic model is fundamentally arbitrary, in that there is no right or wrong value for it. Choosing a large number of topics will lead to more granular topics, while choosing a small number will lead to coarser topics. Some heuristics based on the quality of fit of the model such as the "elbow method" are in common use. In addition, principled approaches have been developed to estimate the optimal number of the topics directly from the data (for instance, the famous Hierarchical Dirichlet Process topic model [41]), but they are regarded as computationally-heavy and rarely used in practice.

Once the topics have been extracted, it is also difficult to characterise them concisely. A "topic" is often nothing more than a categorical distribution over the words in the vocabulary and as such is of large size (2K+). To describe a topic, one typically uses its 5 or 10 most-frequent words. Clearly, the number of employed words has an impact on the description, but it is equally arbitrary. However, some techniques have been develop to "label" the topics in a more principled way (e.g., [42]).

As an example, Table 2.1 displays the top words for two of the topics extracted with three different topic modelling techniques from a corpus of COVID-19 news. In the table, each line corresponds to a topic and each topic is denoted by its top

Table 2.1: Examples of topics extracted from a COVID-19 news dataset (out of 50 total topics).

LDA:	itali countri franc europ european spain italian germani measur lockdown new york citi state cuomo san governor mayor francisco andrew
ProdLDA:	rub sampl mer nasal patient symptom cough lung genet molecular diamond passeng disembark repatri dock princess liner hubei aboard cruis
ProdLDA-GS:	symptom cough respiratori patient ill nose hospit infect doctor sneez democrat biden sander republican trump voter vote senat sen nomin

10 words. By looking at each set of words, one can infer the key thematic of the respective topic. For LDA, the first topic clearly covers the lockdown measures taken by various European countries during COVID; the second names the then New York State Governor Andrew Cuomo and the mayor of San Francisco, but fails to include a clear "reason" for their mention. It could be argued that it covers politicians prominent during COVID. For the model called ProdLDA, the first topic refers to COVID symptoms and testing (word "mer" is the stemmed version of "MERS"); the second refers to the case of the Diamond Princess cruise ship. For ProdLDA-GS, the first example clearly refers to COVID symptoms and the risk of infection for the doctors; the second, to the US presidential primaries which were held during the observation period. These concise descriptions can help the users understand the coverage of the collection, and in some cases even help with the further classification and clustering of the documents.

2.11 Performance Evaluation of Topic Models

Topic modelling is a unsupervised NLP task aiming to produce a useful description of a selected document collection. As such, its evaluation differs substantially from that of conventional predictive models such as classifiers and regression models. In the first place, it is important to evaluate the model's performance on the training set itself ("Is the description extracted satisfactory?") with a range of measures. It is also possible to perform an evaluation over a held-out test set by keeping the extracted topics fixed and only inferring the topic proportions for the held-out documents. This evaluation can still be useful to help prevent overfitting the training set, given that both the training and test sets are expected to respect the

same distributional assumptions. In addition, a held-out evaluation can be useful for "open" collections where new documents may be added at later stages, but still being expected to come from the same distributions and fit in the trained model. Otherwise, it would probably be more appropriate to re-run the model's training to adapt it to the changed contents.

A number of evaluation measures are available for topic modelling, but hereafter we limit ourselves to the most widespread: quantitative metrics such as the perplexity and the topic coherence, and qualitative assessment based on human judgement.

2.11.1 Perplexity

The perplexity is an evaluation metric commonly used for evaluating the performance of topic models and also other NLP tasks such as language models. The perplexity of a model over a given dataset S is defined as:

$$\text{perplexity}(S) = \exp(-\mathcal{L}(S)/(\text{no. of tokens in } S)).$$

In the general case, $\mathcal{L}$ denotes the log-likelihood of the data, but for the variational methods, it is given by the ELBO in (2.11). The perplexity is a measure of the "poorness of fit" of the model onto the data; as such, the lower, the better. It can be measured either on the training set itself or on an independent test set; however, given that it is closely related to the training objective, it is generally advisable to measure it over an independent test set.

2.11.2 Topic coherence

Topic coherence aims to quantify the "coherence" of the extracted topics, as a way to assess the effectiveness of the topic modelling exercise. The coherence can be described in these terms: the top N words of a topic, which somehow "represent" it, should co-occur often within the individual documents. If they instead rarely appear together in the same document, it is questionable that they form a cohesive topic. As an example, imagine that we have extracted a topic post-labelled as "pets"

whose top two words are "dogs" and "cats". If half of the documents only contain the word "dogs" and the other half only "cats", we would better have two separate topics called "dogs" and "cats", respectively. The topic coherence is typically measured over the training set itself since this guarantees the presence of all the top words of all topics. In addition, as a measure it is not directly related to the training objective and is unlikely to overfit. It is still possible to measure the topic coherence over a held-out set, but some workarounds need to be introduced.

The topic coherence has had several definitions in the literature, and we have used the *normalized pointwise mutual information* (`coher-NMPI`) [43] and the C_V *coherence* (`coher-Cv`) [44] in their Gensim implementation for evaluation throughout this thesis. For all the experiments, N has been set to 10. For all the variational methods, the top words per topic have been selected as those with the highest probabilities in the term-topic matrix. For LSI, they have been selected as those with the highest weights in the term-topic matrix (which is not normalized to probability values). For the GANTM approach, they have been selected as those with the highest weights in the discriminator's decoder network (equivalent to the term-topic matrix of LSI).

2.11.3 Qualitative evaluation using human judgment

Topic modelling is an unsupervised task and as such there is no gold standard list of topics to compare against for any given corpus. Topic evaluation using human judgement can be performed as either an observation-based approach, inspecting the top N words in a topic, or somehow interpretation-based. Some possible elements to consider in the evaluation are:

- Word intrusion: the topic is presented in terms of its N top words and appears reasonably coherent overall, but the user has to find words that seem out of place, i.e., intruders.
- Topic intrusion: The user has to find topics that do not seem coherent/consistent/cohesive.

Table 2.2 shows how the topics are presented to the user. The topics that seem incoherent to a human evaluator have been highlighted in red.

Table 2.2: Topics discovered from the 20 Newsgroups dataset (50 topics). Seemingly incoherent topics are highlighted in red.

LDA: monitor keyboard event appl mac usa ibm adapt use multi date paper star robert confer divis surface mean june present know say dont week white go your think year that
AVITM: car bike ride honda bmw gear motorcycle rear dod ford game team baseball player pitcher braves hitter score pitch fan sea newspaper mountain april ii times angeles york francisco cambridge
AVITM-REINF: windows microsoft memory setup mode modem nt port video vga clinton congress economic government bush country administration economy american billion laboratory nasa shuttle lab space engineering flight institute solar spacecraft

2.12 Datasets Used in this Thesis

Hereafter, I provide a description of the datasets that I have used in my thesis:

- **20 Newsgroups**: The 20 Newsgroups dataset [45] is a collection of approximately 18,747 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The reason for using this dataset is that it has been used by virtually every research paper on topic modelling to date and it can thus be regarded as a benchmark. The training set consists of 11,259 documents and the test set consists of 7,488 documents (40% of the whole dataset). All our experiments have used these splits. In the dataset, column "filenames" contains the actual text and has been used for our experiments. The average number of tokens per document in this corpus, after preprocessing and tokenization, is approximately 86. The following example gives an idea of the nature of the documents:

  "Hi, I have a problem, I hope some of the 'gurus' can help me solve. Background of the problem: I have a rectangular mesh in the uv domain, i.e the mesh is a mapping of a 3d Bezier patch into 2d. The area in this domain

which is inside a trimming loop had to be rendered. The trimming loop is a set of 2d Bezier curve segments. For the sake of notation: the mesh is made up of cells. My problem is this: The trimming area has to be split up into individual smaller cells bounded by the trimming curve segments. If a cell is wholly inside the area...then it is output as a whole, else it is trivially rejected. Does anybody know how this can be done, or is there any algo, somewhere for doing this. Any help would be appreciated. Thanks, Ani."

- **Amazon Fine Food Reviews:** The Amazon Fine Food Reviews dataset [18] consists of food reviews of fine foods from Amazon, with $568,454$ reviews from a period of 10 years (up to October 2012). The reviews include product and user information, ratings, and a plain-text review. The main reason for my using this dataset is that it is relevant to the domain of the Food Agility CRC that has sponsored my scholarship. In addition, it consists of predominantly short documents, which has allowed us to experiment with this scenario. The training set consists of $454,763$ reviews whereas the test set consists of $113,691$ reviews (20% of the total). In our experiments, we have made use of the plain-text field, but the title field could also be used. The average number of tokens per document for the plain-text field is approximately 36. The following example gives an idea of the nature of these documents:
  "I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most."

- **Wiki20K**: The Wiki20k dataset [46] consists of approximately $20,000$ English Wikipedia abstracts. The average number of tokens per document in the corpus, after preprocessing and tokenization, is approximately 49. The following example gives an idea of the nature of the documents:
  "The Mid-Peninsula Highway is a proposed freeway across the Niagara Peninsula in the Canadian province of Ontario. Although plans for a highway connecting Hamilton to Fort Erie south of the Niagara Escarpment have surfaced

for decades, it was not until The Niagara Frontier International Gateway Study was published by the Ministry [...]"

- **COVID-19 News**: The COVID-19 News dataset (also known as CORD-19) [17] is a large-scale news dataset consisting of over $1,500,000$ news articles related to the pandemic published since the outbreak took place in late 2019. This dataset is interesting to use as it is a large-scale, recently-collected dataset on a contemporary topic. However, for computational reasons we have carried out our experiments using a corpus subset of $528,838$ documents, with the training set of size $423,078$ and the test set of size $105,770$ (20% of the entire subset). In the dataset, the field that we have employed is the "text" field. The average number of tokens per document in this dataset is approximately 240. The following example gives an idea of the nature of these documents:
  "On Sunday, British Prime Minister Boris Johnson was hospitalized "for tests" because of "persistent" COVID-19 symptoms 10 days after he tested positive, CNN reports. Johnson reportedly went to the unspecified London hospital after his doctor advised him to do so. A press release from his office called the move "precautionary." On March 26, Johnson revealed he had tested positive and that he had been dealing with symptoms since that date. Britain had gone into lockdown two days earlier. Since the 26th, Johnson has been quarantined at his Downing Street residence. He is the first known world leader to have contracted the virus. Roughly a month ago, right around the time the U.K. started dealing with an outbreak, Johnson garnered media coverage for saying he'd shook hands with coronavirus patients during a hospital visit. "I shook hands with everybody, you will be pleased to know, and I continue to shake hands," Johnson said during a press conference that took place on March 3. His positive test was registered 23 days later. On Saturday, Johnson's fiancée, Carrie Symonds, tweeted out that she'd spent a week in bed with coronavirus symptoms. She had not officially been tested for the disease, but said she felt "stronger" and "on the mend" following the week of rest."

- **UNGDC**: the United Nation General Debate Corpus (UNGDC) [47] consists

of the texts of the UN General Debate statements from 1970 to 2015 annotated by country, session and year. We have used this dataset only for experiments on *dynamic* topic modelling, splitting it into yearly slices.

- **ACL dataset**: The ACL dataset [48] includes $10,874$ title and abstract pairs from the ACL Anthology Network which is a repository of computational linguistics and natural language processing articles. With this dataset, too, we have only carried out experiments on dynamic topic modelling, splitting it into yearly slices from 1973 to 2006 (NB: three years are missing).

Chapter 3

A REINFORCEd Variational Autoencoder Topic Model

Topic modeling is an unsupervised natural language processing approach for automatically extracting the main topics from a large collection of documents, and simultaneously assigning the individual documents to the extracted topics. While many algorithms for topic modelling have been proposed in the literature, to date there has been little use of the popular reinforcement learning framework for this task. For this reason, in this chapter we leverage two pillars of reinforcement learning – the policy gradient theorem and the REINFORCE algorithm – to define a novel loss function for training topic models. In the chapter, the loss function is applied to a state-of-the-art topic model based on a variational autoencoder. Experimental results on two social media datasets have shown that the proposed approach has been able to outperform the original variational autoencoder and other baselines in terms of evaluation measures such as model perplexity and topic coherence.

3.1 Introduction and Related Work

The continued growth of digital data sources, and especially social media, has led to an unprecedented rise in the volume of available text documents. This presents a major challenge for the systematic analysis of their contents, together with their

management and organisation. While until the recent past these tasks could be undertaken based on human annotation, nowadays there is a compelling need for computational tools that can automatically extract topics and patterns from document collections and organise them accordingly.

In recent years, topic models have emerged as a powerful, unsupervised tool for identifying useful structure in such vast amounts of unstructured text data.

In technical terms, a topic model is an algorithm that can efficiently discover the main topics of a potentially large corpus of documents, and assign the individual documents to the topics. A "topic" is commonly intended as a characteristic probability distribution over the words of a vocabulary. For example, a topic like "computers" can be described by a probability distribution where words such as "motherboard," "CPU", "monitor," "mouse" and the like have the highest probabilities. In turn, individual documents can be assigned to multiple topics in specific proportions. Topic models have proved useful for the analysis of a variety of data, from scientific publications to user posts on social media [1].

Many topic models have been proposed over the years, primarily based on techniques such as non-negative matrix factorization and variational inference. Latent semantic indexing (LSI) is generally regarded as the first "proper" topic model [10]. However, the most widespread topic model is likely the latent Dirichlet allocation (LDA) [12]. LDA's basic components are: 1) the word distributions of each topic, and 2) the topic proportions of each document. Since both are modeled as multinomial distributions, LDA conveniently uses an eponymous Dirichlet distribution as their prior. The conjugacy between the multinomial and the Dirichlet makes it easy to derive the posteriors and support inference (more details are provided in the following section). In addition, many LDA derivatives have been proposed over time, including, among others, sparse [27], sequential [32], and hierarchical [34] versions.

Recently, neural topic models have started to appear in the literature, joining the benefits of traditional models such as LDA with those of *deep generative models* [14], [16], [40], [49], [50]. Some neural topic models have made use of generative adversarial networks (GANs) [40], [50] and convolutional neural networks (CNNs)

[49]. However, the most effective neural topic models seem to be those based on *variational autoencoders* (VAEs) [14], [16]. Miao et al. in [16]. have proposed a VAE based neural topic model using the logistic normal distribution and the stick-breaking construction to infer the topic proportions. More recently, Srivastava and Sutton in [14] have proposed a neural topic model integrating LDA with a variational autoencoder, establishing state-of-the-art performance on all the tested datasets.

Despite the many available models, to date topic modeling has made limited use of the popular *reinforcement learning* framework [51]. Reinforcement learning offers the potential to leverage both differentiable and non-differentiable "rewards" to guide the extraction of the topics. An example of topic modeling with reinforcement learning has been presented in [52], leveraging word-reweighting rewards to encourage within-topic coherence and between-topic separation. However, we are not aware of any model that has used reinforcement learning to learn an effective *policy* over the topics. For this reason, in this chapter we propose a topic model that uses the policy gradient theorem and the REINFORCE algorithm [19] to improve learning of an effective topic model.

Experiments performed over two challenging datasets (20 Newsgroups and Amazon Fine Food Reviews, both collected from social media) have shown that the proposed approach has achieved a better performance than all the compared approaches in terms of topic coherence and model perplexity in a large majority of cases.

3.2 Methodology

Here, we present an overview of variational autoencoders for topic modeling followed by the proposed approach.

3.2.1 Topic modeling with variational autoencoders

In recent years, deep generative models have gained widespread adoption in the deep learning community, thanks to their effective integration of features of generative models, Bayesian inference and deep neural networks. In particular, variational

autoencoders (VAEs) have proven specially effective at learning representations for latent variables [15], making them appealing for topic modeling.

A VAE is basically a generalized version of an autoencoder, which is a neural network subdivided into an encoder and a decoder. The encoder takes in input a multidimensional measurement, and produces a latent representation in output. In turn, the decoder takes in input the latent representation and produces a "reconstruction" of the original measurement. In the case of a VAE, the reconstruction is simply meant as the probability of the measurement in the parametrized decoder. When VAEs are used for topic modeling, the measurement in input is a document representation, w (typically, a bag-of-words or a TF-IDF vector), while the latent variable is its topic vector, θ. In turn, the likelihood of the document representation, w, can be obtained by marginalizing the topic vector, θ, as in:

$$p(w|\alpha, \beta) = \int_{\theta} p(w, \theta|\alpha, \beta) d\theta \tag{3.1}$$

where α is the parameter of the prior probability over the topics, β is the matrix of the word distributions for all the topics, and $p(w, \theta|\alpha, \beta)$ is the joint probability of the document representation and the topic vector.

The training of a VAE aims to maximize (3.1) over the given document collection. However, this is typically impossible to perform directly. Therefore, the VAE sets to maximize a tractable lower bound (the evidence lower bound, or ELBO) [15]:

$$\mathcal{L}(w|\alpha, \beta) = \mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big] - D_{\mathrm{KL}}(q(\theta|w)\|p(\theta|\alpha)) \tag{3.2}$$

Hereafter, we briefly describe the meaning of the terms in (3.2); further details can be found in [15]. Term $q(\theta|w)$ (the "encoder") estimates the probability of the topic vector for the given document. Term $\log p(w|\theta, \beta)$ (the "decoder") is the log-probability of the document given its topic vector and the word distributions; its expectation over $q(\theta|w)$, $\mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big]$, is the "reconstruction term". Fi-

nally, term $p(\theta|\alpha)$ is a trainable prior over the topic vectors. During training, (3.2) trades off increasing the reconstruction term against reducing the Kullback-Leibler divergence (D_{KL}) between the encoder and the prior.

To facilitate the reparametrization of the encoder and the prior, Srivastava and Sutton in [14] have proposed replacing the usual Dirichlet distribution with a logistic normal distribution. Samples of a logistic normal distribution, $\mathcal{LN}(\mu, \Sigma)$, can be conveniently obtained by applying the softmax operator to samples of a Gaussian distribution of equal parameters, $\mathcal{N}(\mu, \Sigma)$. In turn, the Gaussian distribution can be reparametrized with the common inverse transform approach. Srivastava and Sutton's model [14], called *AVITM* (from autoencoding variational inference for topic models), models the prior as $p(\theta|\alpha) = \mathcal{LN}(\theta|\mu(\alpha), \Sigma(\alpha))$, where $\mu(\alpha)$ and $\Sigma(\alpha)$ are closed-form expressions for the mean and the variance obtained with a Laplace approximation [53]. In turn, the encoder is modeled as $q(\theta|w) = \mathcal{LN}(\theta|\mu(w, \phi_1), \Sigma(w, \phi_2))$, where ϕ_1 and ϕ_2 are the parameters of two feed-forward neural networks that infer, respectively, the mean and covariance of the encoder. Finally, the decoder is given by:

$$p(w|\theta, \beta) = \mathrm{Mult}(w \,|\, \mathrm{softmax}(\beta)\theta) \tag{3.3}$$

where Mult() denotes the multinomial distribution, and the word distributions are parametrized as logits rather than probabilities to bypass the simplex constraint during gradient descent. A second version of the decoder, inspired by products-of-experts and nicknamed *ProdLDA*, first computes the product, and then the softmax:

$$p(w|\theta, \beta) = \mathrm{Mult}(w \,|\, \mathrm{softmax}(\beta\theta)). \tag{3.4}$$

3.2.2 The proposed approach: a VAE topic model with REINFORCE

Reinforcement learning has become increasingly popular in recent years thanks to its ability to train models beyond conventional maximum-likelihood approaches. The main advantages of reinforcement learning are its ability to minimize non-differentiable training objectives and its use of sampling, which permits a certain degree of *exploration* in the parameter space. In the case of our model, the loss function in (3.2) is an expectation over θ, the topic vector for the document, and should therefore not depend on it. However, since the expectation is empirical and based on typically only one sample per document, some dependence on θ persists, and we emphasize it by noting the loss as $\mathcal{L}(\theta)$ in the following. To improve the estimate of the encoder distribution, $q(\theta|w)$, we choose to minimize the *predictive risk*:

$$\mathcal{R} = \mathbb{E}_{q(\theta|w)}\big[\mathcal{L}(\theta)\big] = \int_{\theta} \mathcal{L}(\theta) q(\theta|w) d\theta \tag{3.5}$$

which is the expectation of the loss function, $\mathcal{L}(\theta)$, over the probability of variable θ, the document's topic vector. In order to minimize (3.5), training will attempt to assign high probability to values of θ that cause low values of the loss, and the vice versa, thus promoting an effective encoder. The minimization of (3.5) can be performed using the policy gradient theorem [19], which ignores the indirect dependence of the loss on the model's parameters and only differentiates the probability distribution in its own parameters, ϕ:

$$\begin{aligned} \frac{\partial}{\partial \phi}\mathcal{R} &= \int_{\theta} \mathcal{L}(\theta) \frac{\partial}{\partial \phi} q(\theta|w) d\theta \\ &= \int_{\theta} \mathcal{L}(\theta) \frac{\partial}{\partial \phi} \log q(\theta|w) q(\theta|w) d\theta \\ &= \mathbb{E}_{q(\theta|w)}\big[\mathcal{L}(\theta) \frac{\partial}{\partial \phi} \log q(\theta|w)\big] \end{aligned} \tag{3.6}$$

As common in practice, we compute the resulting expectation empirically from a single sample:

$$\frac{\partial}{\partial \phi}\mathcal{R} \approx \mathcal{L}(\theta)\frac{\partial}{\partial \phi}\log q(\theta|w), \quad \theta \sim q(\theta|w) \tag{3.7}$$

The above estimator of the gradient of the predictive risk is the popular REINFORCE, a fundamental approach of reinforcement learning which has been applied successfully in many fields [19]. However, the REINFORCE estimator typically suffers from high variance, often affecting the stability of training. This issue can be mollified by subtracting a baseline, b, from the loss (an approach known as REINFORCE *with baseline*):

$$\frac{\partial}{\partial \phi}\mathcal{R} \approx \big(\mathcal{L}(\theta) - b\big)\frac{\partial}{\partial \phi}\log q(\theta|w), \theta \sim q(\theta|w) \tag{3.8}$$

With this modification, a training iteration will decrease $q(\theta|w)$ only if the loss, $\mathcal{L}(\theta)$, is greater than b (i.e., a remarkably bad value). Otherwise, it will increase it or leave it unchanged. In addition, from the gradient estimator we can derive an expression for a loss that can be automatically differentiated by common autodiff tools[1]:

$$\mathcal{L}_{REINF} = \big(\mathcal{L}(\theta) - b\big)_{nograd}\log q(\theta|w) \tag{3.9}$$

where subscript *nograd* prevents differentiating the subscripted term.

The VAE loss (3.2) and the REINFORCE loss (3.9) can also be conveniently mixed, to explore trade-offs between the two. We therefore define the overall loss as:

$$\mathcal{L}_{overall} = \mathcal{L}(w|\alpha, \beta) + \epsilon\,\mathcal{L}_{REINF} \tag{3.10}$$

[1] http://www.autodiff.org/, https://www.tensorflow.org/guide/autodiff.

3.3 Experiments and Results

3.3.1 Datasets

The experiments have been carried out over two probing datasets, *20 Newsgroups* (a benchmark for the field) and *Amazon Fine Food Reviews*. The 20 Newsgroups dataset comprises 18,846 documents from news shared on social media, while Amazon Fine Food Reviews consists of 568,454 user-posted food reviews. For 20 Newsgroups, we have used the 1,995 most-frequent words publicly shared by [14] as vocabulary and the same pre-processing for direct comparability of the results. For Amazon Fine Food Reviews, the raw documents have been preprocessed with a combination of tokenization, stopword removal, stemming and lemmatization; special characters and punctuation have also been removed, and the pre-processed documents have been converted to NumPy arrays for input into the various topic models. These datasets are very challenging because of their great variety of topics and their utmost diversity of authors.

3.3.2 Experiments

As models, we have compared the proposed approach against two strong baselines (LDA and LSI) and the state-of-the-art topic model of Srivastava and Sutton, in its two versions AVITM and ProdLDA. For this reason, we present the results for the corresponding versions of our model, AVITM-REINF and ProdLDA-REINF. As hyperparameters, for those shared with the model of Srivastava and Sutton we have used the same values. For the loss balance parameter, ϵ, we have carried out a preliminary evaluation and chosen $\epsilon = 10^{-15}$ since the scale of $\mathcal{L}_{REINF}$ is much larger. To set the baseline, b, we have first trained the models without the REINFORCE loss and recorded the value of their loss at convergence, noted as l; then, we have set b in the range $[l, l \pm 25, l \pm 50]$, using only the training set for the selection. As a number of topics to explore, we have used the oft-used values of 20 and 50. For performance evaluation, we have adopted two popular measures, the *perplexity* and the *topic coherence*. The perplexity measures how poorly the

Table 3.1: Results on the 20 Newsgroups dataset with 20 topics.

Metrics	LDA	LSI	AVITM	ProdLDA	AVITM-REINF	ProdLDA-REINF
Perplexity	*1480.3*	—	1140.2	1173.3	**1137.8**	1167.8
Coher-NPMI	-0.033	-0.053	0.094	0.141	0.131	**0.153**
Coher-Cv	0.309	0.371	0.671	0.779	0.734	**0.786**

Table 3.2: Results on the 20 Newsgroups dataset with 50 topics.

Metrics	LDA	LSI	AVITM	ProdLDA	AVITM-REINF	ProdLDA-REINF
Perplexity	*2389.6*	—	1133.1	1159.9	**1132.1**	1162.8
Coher-NPMI	-2.346	-0.062	0.117	0.111	0.115	**0.141**
Coher-Cv	-0.053	0.294	0.704	0.751	0.699	**0.763**

Table 3.3: Results on the Amazon Fine Food Reviews dataset with 20 topics.

Metrics	LDA	LSI	AVITM	ProdLDA	AVITM-REINF	ProdLDA-REINF
Perplexity	*1480.3*	—	**1000.9**	1099.7	1137.8	1091.4
Coher-NPMI	0.047	0.004	**0.144**	0.066	0.131	0.105
Coher-Cv	0.493	0.395	0.707	0.651	**0.734**	0.676

Table 3.4: Results on the Amazon Fine Food Reviews dataset with 50 topics.

Metrics	LDA	LSI	AVITM	ProdLDA	AVITM-REINF	ProdLDA-REINF
Perplexity	*2697.3*	—	1008.6	1012.5	**1008.3**	1009.0
Coher-NPMI	0.033	-0.008	0.144	-0.048	**0.155**	0.036
Coher-Cv	0.470	0.359	0.682	0.430	**0.699**	0.588

model fits a given set of data (NB: lower values are better); to assess the models' ability to generalize, we have measured it over the test sets. The topic coherence measures the internal "coherence" of the extracted topics (NB: higher values are better). Since coherence can be quantified in different ways, we report both the *normalized pointwise mutual information* (`coher-NMPI`) [43] and the C_V *coherence* (`coher-Cv`) [44]. Unlike the perplexity, the coherence is computed over the training set itself to ensure that all of the topics' M most-frequent words are present in the set. In all the experiments, M has been set to 10. Given the significantly different nature of the perplexity and the topic coherence, some disagreement in their ranking of the models is to be expected.

3.3.3 Results

Tables 3.1 and 3.2 show the experimental results for the 20 Newsgroups dataset for 20 and 50 topics, respectively. Due to the different architecture and amount of degrees of freedom, the perplexity values for LDA cannot be directly compared to those of

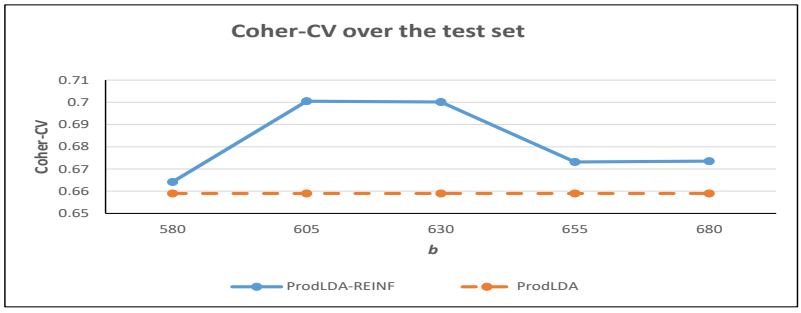

Figure 3.1: Comparison of `coher-CV` on the test data for ProdLDA and ProdLDA-REINF (20 Newsgroups, 50 topics) by varying the baseline, b.

the autoencoder models; for this reason, we display them in italics. At its turn, LSI is not a probabilistic model and the perplexity values are not defined. When compared to the variational autoencoder approaches in terms of coherence, both LDA and LSI have reported significantly worse results and cannot be considered competitive. AVITM has achieved better perplexity values than ProdLDA, but ProdLDA has achieved higher coherence values in most cases, so there is no clear winner between them. However, both our proposed variants have been able to gain improvements over AVITM and ProdLDA, respectively: compared to AVITM, AVITM-REINF has achieved better perplexity and coherence in the case of 20 topics, and coherence in the case of 50 topics; compared to ProdLDA, ProdLDA-REINF has achieved better perplexity as well as coherence in the case of 20 topics, and coherence in the case of 50 topics. Overall, AVITM-REINF has achieved the best perplexity of all compared models, and ProdLDA-REINF the best coherence.

Tables 3.3 and 3.4 show the results for the Amazon Fine Food Reviews dataset with 20 and 50 topics, respectively. Again, LDA and LSI have reported significantly lower coherence values than all the autoencoder models and cannot be regarded as competitive. For this dataset, AVITM has neatly outperformed ProdLDA in both perplexity and coherence. At its turn, our proposed AVITM-REINF has outperformed AVITM in 4 out of 6 measures across 20 and 50 topics, and should be deemed as the best performing model for this dataset. In addition, ProdLDA-REINF has improved in all measures compared to the original ProdLDA. Overall, we can conclude that our REINFORCE-based models have led to marked improvements over both datasets.

As further analysis, we have explored the sensitivity of the topic coherence to the value of the baseline, b, using the test set to simultaneously probe generalization. To this aim, Figure 3.1 plots the values of the C_V coherence for ProdLDA-REINF (20 Newsgroups, 50 topics) over the range of the baseline values. The coherence value for ProdLDA is also displayed for comparison. In this experiment, the loss at convergence without REINFORCE has been $l = 630$, and the best coherence value over the training set has been obtained for $b = l - 25 = 605$. Figure 3.1 shows that this has also been the best value for the test set, showing excellent generalization. In addition, ProdLDA-REINF has achieved better coherence values than ProdLDA for all values of the baseline.

Finally, for a qualitative analysis of the results, Table 3.5 displays a few examples of topics extracted from the 20 Newsgroups dataset. The first topic extracted by LDA is clearly meaningful, but the other two (highlighted in red) seem incoherent. The third topic extracted by AVITM also seems, at least, uninformative. Conversely, all the examples of topics extracted by AVITM-REINF seem consistent and properly descriptive.

Table 3.5: Topics discovered from the 20 Newsgroups dataset (50 topics). Seemingly incoherent topics are highlighted in red.

LDA: monitor keyboard event appl mac usa ibm adapt use multi date paper star robert confer divis surface mean june present know say dont week white go your think year that
AVITM: car bike ride honda bmw gear motorcycle rear dod ford game team baseball player pitcher braves hitter score pitch fan sea newspaper mountain april ii times angeles york francisco cambridge
AVITM-REINF: windows microsoft memory setup mode modem nt port video vga clinton congress economic government bush country administration economy american billion laboratory nasa shuttle lab space engineering flight institute solar spacecraft

3.4 Conclusion

This chapter has presented a novel training loss function for VAE topic models based on the reinforcement learning framework. In the proposed approach, we leverage the

predictive risk and the REINFORCE algorithm to learn an effective policy over the topic vectors. The experimental results over two social media datasets have shown that the proposed approach has been able to attain a strong performance as measured by perplexity and topic coherence, with improvements of up to 2.4 percentage points in NPMI coherence and 2.7 percentage points in C_V coherence compared to the runner-up. In addition, the model has given evidence of good generalization over new documents. In the near future, we plan to explore other architectures for the implementation of the model's neural networks, possibly including transformers and document embeddings.

Chapter 4

Topic-Document Inference with the Gumbel-Softmax Distribution

Topic modeling is an important application of natural language processing (NLP) that can automatically identify the set of main topics of a given, typically large, collection of documents. In addition to identifying the main topics in the given collection, topic modeling infers which combination of topics is addressed by each individual document (the so-called topic-document inference), which can be useful for their classification and organization. However, the distributional assumptions for this inference are typically restricted to the Dirichlet family which can limit the performance of the model. For this reason, in this chapter we propose modeling the topic-document inference with the Gumbel-Softmax distribution, a distribution recently introduced to expand differentiability in deep networks. To set up a performing system, the proposed approach integrates Gumbel-Softmax topic-document inference in a state-of-the-art topic model based on a deep variational autoencoder. Experimental results over two probing datasets show that the proposed approach has been able to outperform the original deep variational autoencoder and other popular topic models in terms of test-set perplexity and two topic coherence measures.

4.1 Introduction

Unstructured textual data are growing by the day in the form of news, press releases, blogs, social media posts and others. The possibility for humans to annotate such documents is limited since manual annotation is labor-intensive and time-consuming. Therefore, there is an urgent and widespread need for automated, unsupervised analysis tools that can provide an understanding of such data and work at scale [54]. Topic modeling is an unsupervised, probabilistic approach of natural language processing (NLP) that is capable of discovering the main topics of large amounts of unstructured text,and presenting them to a user in succinct and comprehensible forms. It has established a strong reputation as a useful text analytics technique and has found application in fields ranging from business and finance to healthcare and scientific corpora analysis [2]–[6], [9], [55]. In topic modeling, a topic is typically represented by the set of its most-frequent words. For instance, a topic such as "cricket" may be represented by words such as "innings", "stump", "wicket" and all the other typical terminology of cricket commentaries. As a more sobering example, a topic such as "pandemic" may be represented by words such as "infection", "intensive care", "death", "recovery" and so forth. In more general terms, a topic can be seen as a probability distribution over the words of an available vocabulary, where the words that are distinctive for that topic are characterized by the highest probabilities.

Topic modeling is able to parse a whole corpus of documents and identify the most common topics "shared" by these documents. Simultaneously, it is able to determine what proportion of topics is addressed by each individual document. The existing approaches for topic modeling are predominantly based on non-negative matrix factorization and probabilistic inference, and the most famous is undoubtedly the latent Dirichlet allocation (LDA) of Blei, Ng and Jordan [12]. In this approach and many of its derivatives, the topic proportions of the individual documents are modeled using the Dirichlet distribution which is a convenient conjugate prior for the topic frequencies. However, limiting the models to this assumption may be restrictive, since other distributions over the topic proportions may be able to achieve better

performance figures for the derived topic models.

For this reason, in this chapter we propose modeling the topic proportions of the individual documents using the Gumbel-Softmax distribution [20], [21]. This distribution has been recently introduced to expand the applicability of backpropagation in deep learning models with latent categorical variables, where it is used to replace non-differentiable, categorical samples with "soft" samples from a differentiable transformation. The main expected advantage of using this distribution for topic modeling is that it can effectively control the sparsity of its samples by a pseudo-temperature hyperparameter, and can thus be able to control the expected number of topics of each individual document during the so-called topic-document inference. To set up a performing system, we have integrated this distribution into the sampling step of a state-of-the-art topic model, the autoencoding variational inference for topic models (AVITM) of Srivastava and Sutton [14].

Experiments have been carried out on two challenging text datasets: the popular 20 Newsgroups dataset [45], consisting of 18,846 user-posted documents from newsgroups, and the recent, large-scale COVID-19 news dataset[1], aggregated by AYLIEN using their news API on more than 400 different sources. The experimental results show that the proposed topic-document inference approach has been able to achieve higher topic coherence and lower perplexity than all the other compared approaches.

The rest of this chapter is organized as follows: Section 4.2 presents the related work. Section 4.3 presents the proposed model, preceding it with a concise review of LDA and a state-of-the-art variational topic model. Section 4.4 describes the experiments, and presents and discusses the results. Section 4.5 concludes the chapter.

4.2 Related Work

Topic modeling is unarguably one of the most researched areas of natural language processing. Its aim is to find concise descriptors for a typically-large ($> 10,000$ documents) given corpus and for its individual documents. This is generally achieved

[1] https://aylien.com/resources/datasets/coronavirus-dataset

by introducing a set of latent variables, known as the "topics", which are shared across the corpus and describe it, while simultaneously determining the proportions of the topics in each document. The input to topic modeling is typically a simplified representation of the documents in the corpus known as the term-document matrix. Topic modeling has found application in a large number of areas including news [9], social media [6], [8] finance [5], [6], healthcare [2]–[4] and many others.

Among the many techniques proposed over the years, latent semantic indexing (LSI, also known as latent semantic analysis, or LSA) is credited as the first explicit topic model [10]. It consists of the factorization of the term-document matrix in a low-rank latent space by means of a singular value decomposition.

To more clearly explain this factorization, which will also be useful for the remainder of the chapter, let us introduce the following notations: V is the size of the given vocabulary, D is the number of documents in the given corpus, K is the number of topics chosen to describe the corpus, and W is the term-document matrix, of $V \times D$ size. The LSI factorization can then be expressed as:

$$W \approx \beta\theta \tag{4.1}$$

where β is a $V \times K$ matrix usually referred to as the term-topic matrix, and θ is a $K \times D$ matrix referred to as the topic-document matrix. The values for β and θ can be obtained by applying singular value decomposition to W, and incorporating the resulting eigenvalues into either of the other two factors. This ensures that $\beta\theta$ is the best possible approximation of W in a least-square sense. For this factorization to be of any practical utility, the chosen number of topics, K, must satisfy $K \ll D$. However, since K is typically chosen in a range such as [20, 100] and the corpora are large, this condition is always easily met. Among various uses, the LSI factorization can be used to compare, cluster and classify documents (e.g. [56]); to extract the top words of each topic; and even to compare and cluster words.

Probabilistic latent semantic analysis (pLSA, or, analogously, pLSI) [11] has overlaid a probabilistic interpretation to the LSI factorization: the first factor, the term-topic

matrix, is interpreted as the probability of a word, w, in a given topic, t, while the second factor, the topic-document matrix, is interpreted as the probability of a topic, t, in a given document, d. Both probabilities are modeled as multinomial distributions. The computation of the factorization is similar to that of LSI, but the elements of the factor matrices must all belong to interval $[0, 1]$, and the relevant columns and rows must abide by a sum-to-one constraint (the simplex domain). The multinomial distributions of the term-topic matrix, $p(w|t)$, are concisely called the "topics", as they express how probable it is that any of the words in the given vocabulary will appear in text from a given topic. The multinomial distributions in the topic-document matrix, $p(t|d)$, are called the "topic vectors" and express the mixture of topics covered by a given document. A highly popular generalization of pLSA called latent Dirichlet allocation (LDA) adds prior probabilities to both the topics and the topic vectors in the form of Dirichlet distributions [12]. Since the Dirichlet distribution is conjugate to the multinomial, the posterior probabilities can be computed analytically, allowing for efficient inference algorithms. We review LDA in detail in Section 4.3.1. LDA has also spawned a large number of extensions and variants, including hierarchical versions[13], [33], sequential versions [32], class-supervised versions [26], sparse versions [28]–[30], and many others.

In recent years, neural topic models have come into the spotlight by combining the advantages of deep neural networks and LDA. Deep models based on variational autoencoders (VAEs) such as [14], [16], [38], [57] have proved effective at automatic discovery of the latent topics in the corpus, and deep models based on CNNs have been used for topic-based document classification and non-negative matrix factorization [49], [58]. Recently, Srivastava and Sutton [14] have proposed a topic model that joins the properties of LDA with the strong representational power of a deep variational autoencoder. This approach has proved to clearly outperform LDA both quantitatively and qualitatively, and can be regarded as one of the current state-of the-art approaches. In addition, various deep topic models have been proposed based on generative adversarial networks (GANs). Among them, [40] uses a denoising autoencoder to implement the discriminator network, under the expectation that the discriminator should achieve a small reconstruction error on the documents in the

corpus, while a large reconstruction error on the synthetic documents generated by the generator network. The main aim of this GAN-based topic model is to provide effective topic vectors for document classification [40]. However, it can also be used for extracting the top words of the topics, and vector representations for the words.

4.3 Methodology

In this section, we present the proposed methodology, preceded by an overview of latent Dirichlet allocation and variational autoencoders for topic modeling.

4.3.1 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA), proposed by Blei, Ng and Jordan in 2003 [12], is probably the reference model for the field of topic modeling. To briefly describe it hereafter, let us introduce the following notations:

- $w_{d,n}$ is the n-th word in the d-th document in the corpus. By "word" we mean a categorical value in the corpus' vocabulary (essentially, an index). The size of the vocabulary is noted as V. Wherever unambiguous, we omit the document index for brevity.

- w_d is the set of all the words in document d (again, where possible, we omit the document index).

- Each word, $w_{d,n}$, is assigned to a corresponding topic, $z_{d,n}$. A topic, too, is a categorical variable taking values in a set of $1 \dots K$ possible values (NB: the topics are "nameless", but can be later assigned meaningful names with a post-analysis). This correspondence means that, for example, a word such as "bat" can be assigned to topic "mammals" in one instance and "cricket" in another.

The model makes the following distributional assumptions:

- The topic variables for a given document are independently and identically distributed according to a multinomial distribution, $\text{Mult}(z_{d,n}|\theta_d)$, parametrized by a K-dimensional probability vector, θ_d.

- At its turn, vector θ_d is distributed according to a Dirichlet distribution, $\text{Dir}(\theta_d|\alpha)$, parametrized by a K-dimensional integer vector, α, shared by the whole corpus. (The conjugacy between the multinomial and Dirichlet eases the computation of the required posteriors.)

- The words in the corpus are distributed according to a set of K multinomial distributions, parametrized by K corresponding V-dimensional probability vectors, $\beta = \beta_1, \ldots \beta_K$. Each word in a given document is independently distributed according to one of these distributions, indexed by its topic variable, as in $\text{Mult}(w_{d,n}|\beta_{z_{d,n}})$.

All these assumptions can be concisely noted in a "generative" model, that is a model that allows sampling an entire synthetic corpus from these distributions:

$$\begin{aligned} &\forall d = 1 \ldots D : \\ &\quad \theta_d \sim \text{Dir}(\theta|\alpha) \\ &\quad \forall n = 1 \ldots N : \\ &\qquad z_n \sim \text{Mult}(z_n|\theta_d) \\ &\qquad w_n \sim \text{Mult}(w_n|\beta_{z_n}) \end{aligned} \tag{4.2}$$

which also corresponds to the following factorization:

$$\begin{aligned} &p(w_n, z_n, \theta_d|\alpha, \beta) \\ &= \text{Mult}(w_n|\beta_{z_n})\text{Mult}(z_n|\theta_d)\text{Dir}(\theta_d|\alpha) \end{aligned} \tag{4.3}$$

Since both w_n and z_n are multinomially distributed, it is also possible to dispose

of z_n altogether by marginalizing it analytically. In this case, the generative model simplifies to:

$$\begin{aligned} &\forall d = 1 \dots D : \\ &\quad \theta_d \sim \text{Dir}(\theta|\alpha) \\ &\quad \forall n = 1 \dots N : \\ &\qquad w_n \sim \text{Mult}(w_n|\beta\theta_d) \end{aligned} \tag{4.4}$$

where with $\beta\theta_d$ we have noted the product between $V \times K$ matrix β and $K \times 1$ vector θ_d. The corresponding factorized probability is:

$$p(w_n, \theta_d|\alpha, \beta) = \text{Mult}(w_n|\beta\theta_d)\text{Dir}(\theta_d|\alpha) \tag{4.5}$$

and the probability for all the words in a document can be simply expressed as:

$$p(w, \theta_d|\alpha, \beta) = \prod_{n=1}^{N} p(w_n, \theta_d|\alpha, \beta) \tag{4.6}$$

The inference problem for this model consists of maximizing (4.6) by estimating θ_d, β and α over a given training corpus of documents. In essence, answering these questions: what is the distribution of words in each of these topics? $(\beta = \beta_1, \dots \beta_K)$; what are the proportions of the topics in each of these documents? $(\theta = \theta_1, \dots \theta_D)$; and what are the proportions of the topics across the whole corpus? (α). For new/test documents given after training is complete, β and α are kept unchanged and only their topic vectors are inferred.

4.3.2 Variational autoencoders for topic modeling

Since the ascendance of deep learning, a fresh wave of models best known as deep generative models (DGM) have come into existence, fundamentally a blend of deep neural nets, generative models and Bayesian inference. Among them, variational

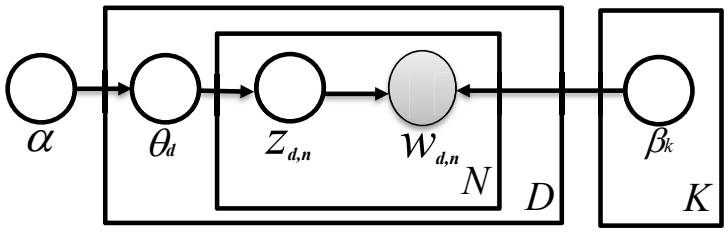

Figure 4.1: The graphical model of LDA. The meaning of the notations is as follows: α denotes the parameter vector for the Dirichlet prior over the topic vectors (i.e. the topic proportions per document), unique for the corpus. θ_d is the topic vector of the d-th document, sampled from $\mathrm{Dir}(\theta_d|\alpha)$. For each document, N topics, $z_{d,n}$, are then sampled from $\mathrm{Mult}(z_{d,n}|\theta_d)$. Finally, the corresponding N words, $w_{d,n}$ are sampled from a multinomial distribution over the vocabulary, $\mathrm{Mult}(w_{d,n}|\beta_{z_{d,n}})$; its parameter vector, $\beta_{z_{d,n}}$, is chosen from a set of K parameter vectors, $\beta = \{\beta_1, \ldots \beta_k \ldots \beta_K\}$, based on the value of topic $z_{d,n}$.

autoencoders (VAEs) have proved very effective for models that contain latent variables (in our case, the topics) [15]. VAEs are able to efficiently maximize the log-likelihood of the observed data even when this function is not directly optimizable, making them widely applicable in all fields of big data including, among others, signal processing, computer vision, natural language processing and transactional data analytics.

A VAE is essentially a generalization of a traditional autoencoder, which is a neural network consisting of two sub-networks: an encoder and a decoder. The encoder receives a multidimensional measurement in input, and outputs a latent representation for it; the decoder receives the latent representation in input, and outputs a "reconstruction" of the original measurement. Through this process, the model is able to generate latent representations and reconstructed measurements which are often more useful than the original measurements in downstream tasks of pattern recognition (e.g. [59]).

A variational autoencoder is a probabilistic extension of an autoencoder where both the measurement and the latent representation are treated as random variables, and therefore the encoder and the decoder are treated as probability distributions. The "reconstruction" of the original measurement is meant in a probabilistic manner in terms of log-likelihood maximization. In the case of our topic model, the aim of the VAE is to maximize the log-likelihood of the words of each document:

$$p(w|\alpha, \beta) = \int_{\theta} p(w, \theta|\alpha, \beta) d\theta \tag{4.7}$$

However, the above objective is too complex to be maximized directly, and therefore the VAE establishes an approachable lower bound for the log-likelihood known as the Evidence Lower Bound, or ELBO, and sets to maximize it [15]. In the case of the topic model, the ELBO has the following form:

$$\begin{aligned} \mathcal{L}(w|\alpha, \beta) =& \mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big] \\ &- D_{\mathrm{KL}}(q(\theta|w) \| p(\theta|\alpha)) \end{aligned} \tag{4.8}$$

The terms in (4.8) have the following meaning: 1) $q(\theta|w)$ is an estimator for the probability of the topic proportions for a given document (represented by its words, w) and is known as the "encoder"; 2) $\log p(w|\theta, \beta)$ is the log-probability of the given document given its topic proportions and is known as the "decoder"; 3) $\mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big]$ is the expectation of this quantity over $q(\theta|w)$ and is known as the "reconstruction term"; 4) $p(\theta|\alpha)$ is a learnable prior probability for the topic proportions that is shared by the entire corpus. The rationale for (4.8) is twofold: first, it is a proven lower bound for (4.7), that is the target of the maximization; second, it consists of a trade-off between two terms that can be interpreted intuitively: the model is rewarded for either improving the reconstruction term, or for keeping the encoder close to the prior.

Srivastava and Sutton in [14] have proposed a VAE for topic modeling (AVITM) that

leverages a Laplace approximation of the usual Dirichlet prior to permit its integration into the autoencoder. In AVITM, both the prior and the encoder are modeled as logistic normal distributions: the prior is modeled as $p(\theta|\alpha) = \mathcal{LN}(\theta|\mu(\alpha), \Sigma(\alpha))$, and the encoder is modeled as $q(\theta|w) = \mathcal{LN}(\theta|f_\mu(\phi, w), f_\Sigma(\phi, w))$, where ϕ are the internal parameters of two neural networks that predict the mean and covariance of the encoder, respectively. The expectation in (4.8) is computed by sampling $q(\theta|w)$, which in turn is performed through reparametrization. The decoder takes the following form:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta)\theta) \tag{4.9}$$

where $\sigma()$ is the softmax operator and the word distributions are parametrized in the softmax basis rather than the simplex to remove unnecessary constraints during backpropagation. The authors have also proposed a second, heuristic version of the decoder, called ProdLDA, that performs the product before the softmax:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta\theta)) \tag{4.10}$$

As shown in [14], both AVITM and ProdLDA have outperformed a number of compared topic model approaches by large margins, and can be regarded as state-of-the-art approaches for this task.

4.3.3 The proposed approach: VAE topic models with the Gumbel-Softmax

The Gumbel-Softmax distribution, co-credited to [20] and [21], has channeled much attention from the deep learning community in recent years. This distribution models "soft" categorical variables (categorical variables that are not restricted to have one-hot values) and has been introduced to circumvent issues related to backpropagation in models with latent categorical variables. Many deep learning models (prominently, variational autoencoders and generative adversarial networks,

or GANs) need to sample from distributions, and sampling is a non-differentiable operation that breaks the backpropagation chain. The Gumbel-Softmax distribution is an alternative to the multinomial distribution that allows sampling of quasi-categorical variables and is differentiable via reparametrization. Given a multinomial distribution, $\text{Mult}(z|\theta)$, with K possible values, samples from the corresponding Gumbel-Softmax distribution, $\mathcal{GS}(\tilde{z}|\theta, \tau)$, can be obtained as:

$$\begin{aligned} \tilde{z} &= \sigma\big([\log\theta - \log(-\log u)]/\tau\big) \\ u &\sim \mathcal{U}(0,1)^K \end{aligned} \tag{4.11}$$

where u is a vector of K random variables each sampled from the uniform distribution over $(0, 1)$, and τ is a hyperparameter (referred to as "temperature") that controls the sparsity of $\tilde{z}$ (the lower τ, the more the samples resembles one-hot values; the larger, the more the samples become uniform). Note that the sampled distribution is fixed and does not need gradient updates, and the functions in (4.11) are all differentiable.

To take advantage of its properties, we propose sampling the topic vector from a Gumbel-Softmax distribution. The modified decoder (nicknamed *AVITM-GS*) becomes:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta)\,\tilde{z}),\ \ \tilde{z} \sim \mathcal{GS}(\theta, \tau) \tag{4.12}$$

and in the case of ProdLDA (*ProdLDA-GS*) it becomes:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta\tilde{z}),\ \ \tilde{z} \sim \mathcal{GS}(\theta, \tau) \tag{4.13}$$

Please note that the number of trainable parameters is the same as in the original decoders, with the exception of the scalar hyperparameter τ that we can use to control the sparsity of the inferred topic vectors.

4.4 Experiments and Results

4.4.1 Datasets

As datasets for the experiments, we have used the popular 20 Newsgroups dataset (a de-facto benchmark for the field) and a 500K-document subset of AYLIEN's recently released COVID-19 news dataset. 20 Newsgroups consists of 18,846 news documents posted by users, split over 11,314 as training set and 7,532 as test set. The average length of these documents is 311 words. To be consistent with the experiments carried out in [14], we have used the preprocessed version publicly released by the authors[2] which uses a vocabulary of 1,995 words. The COVID-19 news dataset is a dataset aggregated by AYLIEN using their News Intelligence Platform from November 2019 to July 2020 from approximately 440 different sources. For our experiments, we have used the first 500K documents (over 7 GB of uncompressed text) split over 400K as training set and 100K as test set since this size could still be managed by a PC with 16 GB of RAM. The documents were preprocessed with tokenization, stop-word elimination, stemming and lemmatization, and encoded with a vocabulary formed by the most-frequent 5,000 unique words.

4.4.2 Experimental set-up

To probe the comparative performance of the proposed approach, we have integrated it in both AVITM and ProdLDA, and compared these versions with the original versions. In the following, we refer to them as AVITM-GS and ProdLDA-GS, respectively. We have also included LDA and LSI from Gensim [60] in the comparison as baselines, and the GAN-based topic model from [40] that we refer to as GANTM in the following. As learning rate for the variational autoencoders, we have used the rather standard value of 0.001. Any other hyperparameters were left to their default values. For the temperature of the Gumbel-Softmax distribution, τ, we have carried out a preliminary sensitivity analysis and chosen to run experiments with $\tau \in [1.5 - 2.5]$ in steps of 0.25. This range corresponds to moderately-sparse to dense topic vectors. As number of topics, we have used both 50 and 100 top-

[2] Available at: https://github.com/akashgit/autoencoding_vi_for_topic_models.

ics for both datasets. We have also initially carried out multiple runs per model, and realised that the performance did not vary significantly ($< 0.5\%$ in all cases). Therefore, in Section 4.4.3 we report results from single runs of each model.

As an unsupervised technique, the performance evaluation of a topic model is non-trivial. For our work, we have used two common measures:

- *perplexity over the test set*: the perplexity of a model over a set S is defined as: $\text{perplexity}(S) = \exp(-\mathcal{L}(S)/(\text{number of tokens in } S))$. In the general case, $\mathcal{L}$ denotes the log-likelihood of the data, but for the variational methods (all except LSI and GANTM in our case), it is given by the ELBO in (4.8). The perplexity is a measure of the "poorness of fit" of the model on the data (the lower, the better) and, as such, it is important that it is measured over an independent test set for realistic generalization.

- *topic coherence*: topic coherence quantifies the coherence of a topic by measuring how often its top K words co-occur within a text window that slides across the documents (the higher the co-occurrence, the better). Since this measure is not uniquely defined, we report both the normalized pointwise mutual information (`coher-NPMI`) [43] and the C_V coherence (`coher-Cv`) [44] from their Gensim implementation. The coherence is typically measured on the training set itself since this guarantees the presence of all the top words. For the experiments, K has been set to 10. For the variational methods, the top words per topic have been selected as those with highest probability in the term-topic matrix. For LSI, they have been selected as those with highest weight in the term-topic matrix (which is not normalized to probability values). For GANTM, they have been selected as those with highest weight in the discriminator's decoder network (equivalent to the term-topic matrix of LSI).

Given their significantly different nature, some disagreement in model ranking between perplexity and topic coherence is to be expected. Perplexity is, essentially, a measure of fit of the model, while topic coherence is a measure of quality of the ex-

Table 4.1: Results with 50 topics on 20 Newsgroups.

Measure/Model	LDA	LSI	GANTM	ProdLDA	AVITM	ProdLDA-GS	AVITM-GS
Perplexity	*2389.6*	—	—	1159.9	1133.0	1136.6	**1110.6**
Coher-NPMI	-2.346	-0.062	-0.234	0.111	0.117	**0.148**	0.104
Coher-Cv	-0.053	0.294	0.247	0.751	0.704	**0.806**	0.638

Table 4.2: Results with 100 topics on 20 Newsgroups.

Measure/Model	LDA	LSI	GANTM	ProdLDA	AVITM	ProdLDA-GS	AVITM-GS
Perplexity	*4857.1*	—	—	1147.1	1128.0	1136.1	**1111.4**
Coher-NPMI	-0.063	-0.071	-0.223	0.114	0.085	**0.117**	0.079
Coher-Cv	0.296	0.267	0.259	0.742	0.650	**0.763**	0.616

Table 4.3: Results with 50 topics on COVID-19.

Measure/Model	LDA	LSI	ProdLDA	AVITM	ProdLDA-GS	AVITM-GS
Perplexity	*1130.0*	—	2178.7	1909.0	1957.7	**1850.5**
Coher-NPMI	0.086	-0.008	0.076	**0.180**	0.170	0.175
Coher-Cv	0.589	0.310	0.682	0.760	**0.787**	0.744

Table 4.4: Results with 100 topics on COVID-19.

Measure/Model	LDA	LSI	ProdLDA	AVITM	ProdLDA-GS	AVITM-GS
Perplexity	*1119.2*	—	2251.7	1904.3	**1855.2**	1855.7
Coher-NPMI	0.090	-0.017	0.049	**0.177**	0.174	0.158
Coher-Cv	0.581	0.271	0.652	0.736	**0.765**	0.700

tracted topics and may better reflect the user's perception of performance. For this reason, for comparing the models we resort to a majority criterion, with emphasis on the topic coherence.

4.4.3 Results

Tables 4.1 and 4.2 report the results over the 20 Newsgroups dataset for 50 and 100 topics, respectively. In terms of test-set perplexity, it is evident that the proposed approach has been able to improve over the original variational autoencoder, both for ProdLDA and AVITM. In these and the following tables, we report the perplexity also for LDA, but the scale of its ELBO is not directly comparable with that of the autoencoder techniques; for this reason, its values are marked in italics and not commented further. For LSI and GANTM, the perplexity is simply not available

Table 4.5: Results for ProdLDA-GS (50 topics, 20 Newsgroups) with varying temperature hyperparameter, τ.

Measure/τ	10^{-5}	1.5	1.75	2.0	2.25	2.5	10
Perplexity	1131.7	1180.0	1161.1	1145.4	1136.6	1124.7	**1099.7**
Coher-NPMI	-0.224	0.126	0.131	0.125	**0.148**	**0.148**	0.010
Coher-Cv	NaN	0.788	0.780	0.785	**0.806**	0.799	0.638

Table 4.6: Examples of topics extracted from the COVID-19 dataset (50 topics).

LDA:	itali countri franc europ european spain italian germani measur lockdown new york citi state cuomo san governor mayor francisco andrew south korea japan africa countri north tokyo korean japanes brazil
ProdLDA:	rub sampl mer nasal patient symptom cough lung genet molecular diamond passeng disembark repatri dock princess liner hubei aboard cruis trophi leagu europa juventus hudson champion coach footbal munich albert
ProdLDA-GS:	symptom cough respiratori patient ill nose hospit infect doctor sneez democrat biden sander republican trump voter vote senat sen nomin crude barrel oil opec investor output price brent bpd yield

since they are not probabilistic models. In terms of coherence, ProdLDA-GS has been able to achieve significantly higher values than all the other techniques in both coherence metrics. In addition, the two topic model baselines, LDA and LSI, and the GANTM model have scored significantly lower values of topic coherence than all the variational autoencoder approaches. Overall, ProdLDA-GS has achieved the best performance in 4 cases out of 6 (combined number of topics/metrics) and can be regarded as the best-performing technique overall.

In turn, Tables 4.3 and 4.4 report the results over the COVID-19 dataset for 50 and 100 topics, respectively. In terms of test-set perplexity, the proposed approach has again been able to improve over the original variational autoencoders. In terms of coherence, the original AVITM has achieved the highest values for `coher-NPMI`, while ProdLDA-GS has achieved the highest values for `coher-Cv`. Again, all the variational autoencoder approaches have scored significantly higher coherence values than both the LDA and LSI baselines. GANTM generated an out-of-memory error while training over larger training sets, and is therefore not reported. Overall, ProdLDA-GS has achieved the best performance in 3 cases out of 6 and may still be regarded as the best-performing overall.

As expected, the choice of the temperature hyperparameter, τ, in the Gumbel-

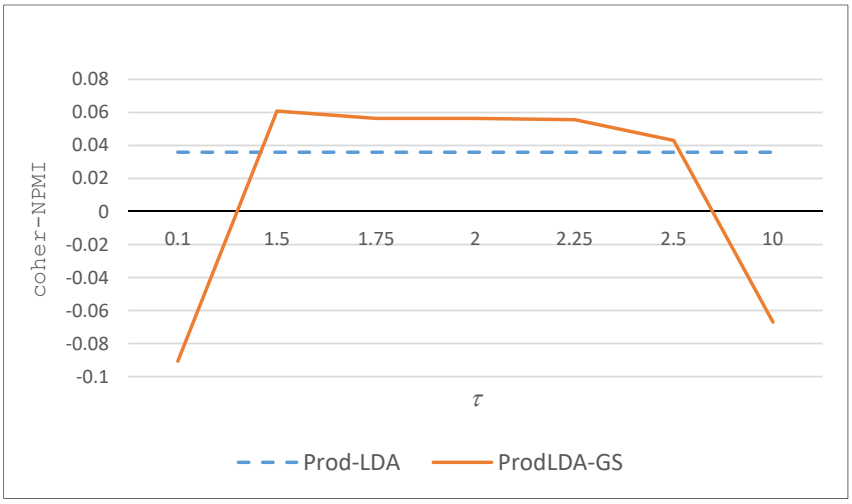

Figure 4.2: Comparison of `coher-NPMI` on the test set for ProdLDA and ProdLDA-GS (50 topics, 20 Newsgroups) with varying temperature hyperparameter, τ.

Softmax distribution has a major impact on the performance as it substantially changes the shape of the samples (from almost one-hot to almost uniform). Since the coherence measures are to be computed on the training set, it is legitimate to choose the value of τ that empirically maximizes them. Conversely, the perplexity is a test-set measure and the optimal τ should be chosen on the training set or a separate validation set. In all cases, the different measures may be maximized by different values of τ, and a trade-off between them is required. To illustrate this dependence, Table 4.5 shows the results with varying τ for ProdLDA-GS with 50 topics on 20 Newsgroups. With $\tau = 10^{-5}$ (almost one-hot samples), the model has achieved a very low coherence. At the other end of the spectrum, with $\tau = 10$ (almost uniform samples), the coherence has been again very low. The equal-best `coher-NPMI` coherence values have been achieved with $\tau = 2.25$ and 2.5, and the best value for `coher-Cv` has been achieved with $\tau = 2.25$, so we have used these results for the comparison in Table 4.1. To further evaluate the model's quality with varying τ, we have also measured the topic coherence (`coher-NPMI`) of ProdLDA-GS over the test set, using ProdLDA as the reference. Figure 4.2 shows that τ has played a key role also for this measure: for $\tau \in [1.5 - 2.5]$, the topic coherence of ProdLDA-GS has been invariably higher than that of Prod-LDA, while it has noticeably deteriorated for more "extreme" values $(0.1, 10)$.

In terms of qualitative analysis of the extracted topics, all approaches seem to have performed well overall. The extracted topics are presented to the user as the lists of their $K = 10$ top words, and such lists must appear informative and coherent. Examples for LDA, ProdLDA and ProdLDA-GS from the COVID-19 topic models are displayed in Table 4.6. For LDA, the first example clearly addresses the lockdown measures taken by various European countries; the second names New York State Governor Andrew Cuomo and the mayor of San Francisco, but fails to include the "reason" for their mention; and the last is simply a list of countries, again with no explicit mention of the COVID outbreak. For ProdLDA, the first example refers to COVID symptoms and testing (word "mer" is the stemmed version of "MERS"); the second refers to the case of the Diamond Princess cruise ship; and the last addresses football news from the observation period. For ProdLDA-GS, the first example clearly refers to COVID symptoms and the risk of infection for the doctors; the second to the recent US presidential primaries, which were held during the observation period; and the last to economic news. Their lists of top words seem very consistent and descriptive. A possible limitation of both ProdLDA and ProdLDA-GS, and possibly of all autoencoding methods which are based on sampling, is the presence of a number of repeated topics. However, it should be easy to prune them post-hoc.

4.5 Conclusion

This chapter has presented an approach for topic modeling based on the Gumbel-Softmax distribution and variational autoencoders. During the step of topic-document inference, the topic proportions of the current document are sampled in the autoencoder from a Gumbel-Softmax distribution with appropriate temperature. The samples are then used to mix either the topic distributions (AVITM-GS) or their logits (ProdLDA-GS). To validate the proposed approach, experiments have been carried out on two challenging datasets, the well-known 20 Newsgroups and a recently-released, large-scale COVID-19 news dataset. The experimental results have shown that the proposed approach has been able to outperform the original variational

autoencoders and two significant baselines in terms of topic coherence, and achieve the best trade-off across two coherence metrics and the test-set perplexity. In addition, a qualitative analysis of the extracted topics has shown that they appear informative and consistent. In the near future, we plan to extend our research to other distributional models and reparametrization approaches.

Chapter 5

Neural Topic Model Training with the REBAR Gradient Estimator

Topic modelling is an important approach of unsupervised machine learning that allows automatically extracting the main "topics" from large collections of documents. In addition, topic modelling is able to identify the topic proportions of each individual document, which can be helpful for organizing the collections. Many topic modelling algorithms have been proposed to date, including several that leverage advanced techniques such as variational inference and deep autoencoders. However, to date topic modelling has made limited use of reinforcement learning, a framework that has obtained vast success in many other unsupervised learning tasks. For this reason, in this paper we propose training a neural topic model using a reinforcement learning objective, and minimizing the objective with the recently-proposed REBAR gradient estimator. Experiments performed over two probing datasets have shown that the proposed model has achieved improvements over all the compared models in terms of both model perplexity and topic coherence, and produced topics that appear qualitatively informative and consistent.

5.1 Introduction

The recent years have witnessed an astonishing growth of unstructured text data in the form of blogs, social media posts, web pages, speech-to-text transcriptions, automated translations, and so forth. Manually analyzing such vast amounts of text data is simply prohibitive, and it is therefore necessary to turn to machine learning and text analytics to set up some form of automated analysis. However, standard machine learning approaches such as classification and regression expect that a significant amount of training data be manually annotated, thus reintroducing a "human bottleneck". Given the typical size and diversity of the relevant datasets, the most suitable candidates for this type of tasks are unsupervised or few-shot machine learning approaches [54].

Within the unsupervised machine learning domain, topic modelling is a popular approach for the identification of the thematic content of a collection of documents. The goal of topic modelling is to automatically discover the main topics of typically-large document collections, and simultaneously identify the topic proportions in each of their documents. Its fundamental assumptions are that each document is thought of as deriving from a combination of topics, and each topic is represented by characteristic frequencies of words in a vocabulary. For instance, a topic such as "neurology" may be characterized by the frequent occurrence of words such as "aphasia", "cortex" and so forth, and a topic such as "education" by words such as "tutorial", "exam" etc. Therefore, a document representing an assignment in a neurology course could be associated with these two topics in specific proportions. Topic modelling is a very well established approach for the analysis of document corpora and has found copious application in domains as diverse as healthcare [2]–[4], finance [5], [6], agriculture [7], social media [6], [8], news [9] and many others.

Topic modelling approaches revolve around the concepts of topics, documents and words. The topics are typically treated as a set of latent categorical values; the documents are treated as either sets or sequences of words; and the words are treated as either categorical values out of a given vocabulary, or as word embeddings. For instance, the popular latent Dirichlet allocation (LDA) treats the words as categor-

ical values, and assumes Dirichlet priors over their multinomial distributions [12]. Conversely, Gaussian-LDA treats words as embedding vectors, and models them with multivariate Gaussian distributions [61], [62]. In all cases, most topic modelling approaches are probabilistic, since probabilistic framings allow for a flexible and effective treatment of latent variables. The number of approaches proposed to date is remarkable, and we concisely review the main in the next section.

However, despite the many approaches proposed in the literature, topic modelling has to date made limited use of *reinforcement learning*. Reinforcement learning is a widely adopted framework for unsupervised tasks, since it can exploit a variety of reward functions to drive the model toward effective parametrizations [51]. An example of application of reinforcement learning to topic modelling is [52], that has proposed word weighting rewards to encourage within-topic coherence and discourage topic overlapping. However, no approaches we are aware of have used reinforcement learning to learn an effective "policy" over the topics themselves. For this reason, in this chapter we propose an approach to topic modelling that leverages the policy gradient theorem [19], [51] to learn the topic distributions. The gradient of the objective function is estimated using the recently-proposed *REBAR* gradient estimator [22] that has a number of attractive properties, including being unbiased with respect to the exact gradient and enjoying a low-variance design. To compose a performing solution, we have integrated the reinforcement learning objective and the REBAR gradient estimator in the state-of-the-art deep variational-autoencoder topic model of Srivastava and Sutton [14]. Experiments performed on two probing document datasets – 20 Newsgroups [45], consisting of $18,747$ documents from newsgroups, and Amazon Fine Food Reviews [18], consisting of $568,454$ food reviews from $256,059$ users of diverse background – have shown improvements over all the compared models in terms of both model perplexity and topic coherence. The main contributions of our chapter can be summarized as follows:

- the use of a reinforcement learning objective (i.e., the predictive risk) for training a deep variational autoencoder for topic modelling (Section 5.3.4);

- the adoption of a recently-proposed gradient estimator (REBAR [22]) that is

both unbiased and low-variance to minimize the objective based on the policy gradient theorem (Section 5.3.5);

- positive experimental results showing that the proposed approach has been able to outperform standard baselines, a Bayesian nonparametric topic model [41], and a state-of-the-art neural topic model [14] (Section 5.4).

The rest of this chapter is organized as follows: the related work is presented in Section 5.2. A brief summary of LDA and a state-of-the-art variational-autoencoder topic model are presented in Section 5.3. The proposed model is presented in Section 5.3.3, while the experiment and experimental results are described in Section 5.4. Finally, concluding remarks are addressed in Section 5.5.

5.2 Related Work

Topic modelling is a well-established unsupervised technique for identifying the main topics in a collection of unstructured text documents. Concurrently to identifying the topics, topic modelling extracts the topic proportions of each individual document, which can be useful for their categorization and organization. The input data for a topic modelling algorithm are typically arranged as a *term-document matrix*, where the rows are the words in the vocabulary, the columns are the documents in the collection, and the individual elements are the number of occurrences of each vocabulary word in each document. Before being converted into the term-document matrix, the documents are usually pre-processed by steps such as punctuation and stopword removal, stemming, lemmatization, and various others [63]. In addition, in order to curb complexity and increase robustness, the vocabulary is typically limited to the words with highest frequency in the collection. As a versatile natural language processing technique, topic modelling has found successful application in a number of areas including marketing, finance, social media, healthcare, news and several others [2], [3], [5], [6], [8], [9].

Many topic modelling approaches have been proposed over the years, but we limit this brief review to the techniques which are needed to position the proposed ap-

proach. Latent semantic indexing (LSI) (also known as latent semantic analysis (LSA)) [10] is considered as the first, proper topic model. LSI learns the hidden topics by carrying out a matrix decomposition of the term-document matrix using singular value decomposition (SVD). SVD is a low-dimensional factorization of the original matrix that drastically reduces the total number of degrees of freedom, while ensuring minimum mean square error (MSE) with respect to the original matrix. To describe LSI more precisely, let us note the size of the vocabulary (i.e. the number of distinct words) as V; the number of documents in the collection as D; the chosen number of hidden topics as K, with $K \ll D$; and the term-document matrix (a $V \times D$ matrix) as W. With these notations, the factorization of LSI can be expressed as:

$$W \approx \beta\theta \tag{5.1}$$

where β is a $V \times K$ matrix known as term-topic matrix and θ is a $K \times D$ matrix known as topic-document matrix. The product of β and θ approximates W in an MSE sense. In addition, the columns of β can be interpreted as the "weights" of the various words in each of the K topics, and the columns of θ can be interpreted as the weights of the various topics in each of the D documents.

Probabilistic latent semantic analysis (pLSA, or pLSI) [11] improves the interpretation of LSI by adding a proper probabilistic modelling for the terms. First, the columns of W are normalized to add up to one, so that they can be interpreted as the probability of the words in the given document, $p(w|d)$. Then, simplex constraints are imposed on the columns of β and θ, so that they can be interpreted as the probability of the words in a given topic, $p(w|t)$, and the probability of the topics in a given document, $p(t|d)$, respectively. For brevity, we refer to the columns of β as the "word distributions", and to the columns of θ as the "topic vectors". Both these distributions are conventional multinomial distributions. With these positions, pLSA can be expressed as:

$$p(w|d) = \sum_{t=1}^{K} p(w|t)p(t|d) \quad w = 1 \dots V, d = 1 \dots D \tag{5.2}$$

Latent Dirichlet allocation (LDA), a generalization of pLSA proposed by Blei, Ng and Jordan in 2003 [12], is a highly popular probabilistic generative model that incorporates prior probabilities from the Dirichlet family for both the word distributions and the topic vectors. Since the Dirichlet distribution is conjugate to the multinomial, it is analytically possible to derive the posterior probabilities. More details of the model will be provided in Section 5.3.1. Over the years, LDA has proved an extremely successful approach and has spawned many variants and extensions, including sparse [27]–[31], hierarchical [13], [33], [34], [41], sequential [32] and class-supervised [26] LDA.

More recently, several topic models have been proposed that integrate features of LDA with those of deep generative models [64]. Among them, [14], [16], [38], [57] have all used variational autoencoders (VAEs) to build topic models for large document collections. Other deep topic models have employed generative adversarial networks (GANs) [40], [50] and CNNs [49], [58]. In particular, Srivastava and Sutton in [14] have proposed a neural topic model that combines LDA with a deep variational autoencoder and has achieved state-of the-art performance in both qualitative and quantitative evaluations. For this reason, we have adopted it as the base model for our approach and for performance comparison.

The typical training objectives of topic models are differentiable functions that can be minimized by gradient descent. However, some training objectives may contain non-differentiable terms. In this case, reinforcement learning and the policy gradient theorem can be used for their minimization [51]. The most well-known approach based on the policy gradient theorem is REINFORCE [19]. However, REINFORCE is a sampling-based algorithm that suffers from very high variance, and for this reason several reduced-variance estimators such as actor-critic algorithms [65] and the Gumbel-Softmax [21] have been proposed. An example of use of a low-variance estimator is [66], where Gumbel-Softmax sampling has been added to a deep variational topic model. However, the Gumbel-Softmax and many REINFORCE variants suffer, in turn, from *bias*, i.e. an average difference from the exact gradient of the target objective, which can lead to sub-optimal parametrizations. For this reason,

Tucker *et al.* have recently introduced REBAR [22], a gradient estimator for the training objective which is simultaneously low-variance and unbiased. To reduce the variance, REBAR uses a control variate that is aptly sampled from a conditional, truncated Gumbel distribution and exhibits high correlation with the training objective. We will describe this estimator in detail in Section 5.3.5. To the best of our knowledge, ours is the first attempt to leverage the policy gradient theorem and REBAR in topic modelling.

5.3 Methodology

In this section, we cover the background needed to understand the proposed model, namely LDA (subsection 5.3.1) and variational-autoencoder topic models (subsection 5.3.2).

5.3.1 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) [12] is to date the reference model for the field of topic modeling. To briefly describe it hereafter, let us introduce the following notations:

- $w_{d,n}$ is the n-th word in the d-th document in the corpus. By "word" we mean a categorical value in the corpus' vocabulary (essentially, an index). The size of the vocabulary is noted as V.
- w_d is the set of all the words in document d.
- Each word, $w_{d,n}$, is assigned to a corresponding *topic*, $z_{d,n}$. The topic is another categorical variable, simply taking values in an index set, $1 \dots K$ (once all the words have been assigned to their topics, the "scope" of each topic can be determined by analyzing its word distribution).

The model makes the following distributional assumptions:

- The topic variables for a given document are independently and identically dis-

tributed according to a multinomial distribution, $\text{Mult}(z_{d,n}|\theta_d)$, parametrized by a K-dimensional probability vector, θ_d.

- At its turn, vector θ_d is distributed according to a Dirichlet distribution, $\text{Dir}(\theta_d|\alpha)$, parametrized by a K-dimensional integer vector, α, shared at corpus level. Since the Dirichlet distribution is a conjugate prior for the multinomial, posteriors can be computed in closed form.

- The words in the corpus are distributed according to a set of K multinomial distributions, one per topic. Each such distribution is parametrized by a V-dimensional probability vector, noted as $\beta_k, k \in [1 \dots K]$. Each word in a given document is distributed according to one of these distributions, indexed by its topic variable, as in $w_{d,n} \sim \text{Mult}(w_{d,n}|\beta_{z_{d,n}})$.

Fig. 5.1 shows the overall model as a graphical model. Since both $w_{d,n}$ and $z_{d,n}$ are multinomially distributed, it is possible to marginalize $z_{d,n}$ analytically. This allows us to rewrite the probability of $w_{d,n}$ as:

$$w_{d,n} \sim \text{Mult}(w_{d,n}|\beta\theta_d) \tag{5.3}$$

where $\beta\theta_d$ stands for the product between $V \times K$ matrix $\beta = [\beta_1 \dots \beta_K]$ and $K \times 1$ vector θ_d. Eventually, this allows us to derive the posterior probability of the word and the topic vector as:

$$p(w_{d,n}, \theta_d|\alpha, \beta) = \text{Mult}(w_{d,n}|\beta\theta_d)\text{Dir}(\theta_d|\alpha) \tag{5.4}$$

and the probability of all the words in a document and their topic vector can be simply expressed as:

$$p(w_d, \theta_d|\alpha, \beta) = \prod_{n=1}^{N} p(w_{d,n}, \theta_d|\alpha, \beta) \tag{5.5}$$

The training objective for this model consists of maximizing Equation (5.5) by esti-

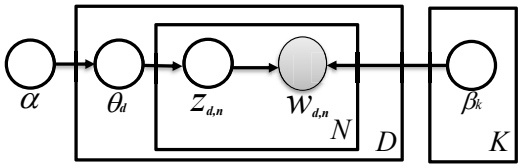

Figure 5.1: The graphical model of LDA. Notations are as follows: α denotes the parameter vector for the Dirichlet prior over the topic vectors (i.e. the topic proportions per document), unique for the corpus. θ_d is the topic vector of the d-th document, sampled from $\text{Dir}(\theta_d|\alpha)$. For each document, N topics, $z_{d,n}$, are sampled from $\text{Mult}(z_{d,n}|\theta_d)$. Finally, the corresponding N words, $w_{d,n}$ are sampled from a multinomial distribution over the vocabulary, $\text{Mult}(w_{d,n}|\beta_{z_{d,n}})$; its parameter vector, $\beta_{z_{d,n}}$, is chosen from a set of K parameter vectors, $\beta = \{\beta_1, \ldots \beta_k \ldots \beta_K\}$, based on the value of topic $z_{d,n}$.

mating α, β, and the θ_d vector of every document over a given training corpus. In essence, estimating: the word distributions in each of the topics, $\beta_1, \ldots \beta_K$; the topic proportions in each of the documents, $\theta_1, \ldots \theta_D$; and the topic proportions across the whole corpus, α. For new/test documents given after training is complete, α and β are kept unchanged, and only their topic vectors are inferred.

5.3.2 Variational-autoencoder topic models

In recent years, variational autoencoders (VAEs) have established themselves as very effective and flexible models for problems that include latent variables [15]. These features make them attractive for topic modelling, where both the topics and the topic proportions are to be treated as latent variables. Another appealing feature of VAEs is that they are able to maximize the log-likelihood of a given training set even when this function is not directly optimizable, by maximizing an analytical lower bound. In addition, VAEs integrate features of deep learning by modelling the parameters of their distributions via neural networks of arbitrary complexity that can be tailored to the needs of the application at hand. For these reasons, they have found wide adoption over a range of tasks in computer vision, signal processing,

natural language processing and many other fields.

A VAE is essentially a generalization of a conventional autoencoder, which is a neural network split over two sub-networks: an encoder and a decoder. The encoder receives a multidimensional measurement in input, and outputs a latent representation for it, typically much smaller in dimension; the decoder receives the latent representation in input, and outputs a "reconstruction" of the original measurement. Through this process, the model is able to generate latent representations and reconstructed measurements which are often more useful than the original measurements in downstream tasks of pattern recognition.

A variational autoencoder is a probabilistic extension of an autoencoder where both the measurement and the latent representation are treated as random variables, and therefore the encoder and the decoder are treated as probability distributions. The "reconstruction" of the original measurement is meant in a probabilistic manner in terms of log-likelihood of the measurement. In the case of topic models, the aim of the VAE is to maximize the log-likelihood of the words of each document[1]:

$$\log(w|\alpha,\beta) = \log \int_{\theta} p(w,\theta|\alpha,\beta)d\theta \tag{5.6}$$

However, Equation (5.6) is too complex to be maximized directly, and therefore the VAE establishes an approachable lower bound for the log-likelihood known as the *evidence lower bound*, or ELBO, and sets to maximize it [15]. In the case of the topic model, the ELBO has the following form:

$$\begin{aligned}\mathcal{L}(w,\alpha,\beta) = \mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta,\beta)\big] \\ - D_{\mathrm{KL}}(q(\theta|w)\|p(\theta|\alpha))\end{aligned} \tag{5.7}$$

The terms in Equation (5.7) have the following meaning:

- $q(\theta|w)$ is an estimator for the probability of the topic proportions for a given

[1] In this section, we omit the document index to avoid unnecessarily cluttering the notations.

document (represented by its words, w), and is known as the "encoder";

- $\log p(w|\theta, \beta)$ is the log-probability of the document given its topic proportions and the word distributions, and is known as the "decoder";

- $\mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big]$ is the expectation of this quantity over $q(\theta|w)$ and is known as the "reconstruction term";

- $p(\theta|\alpha)$ is a learnable prior probability for the topic proportions that is shared by the entire corpus.

The rationale for Equation (5.7) is twofold: first, it is a proven lower bound for Equation (5.6), that is the target of the maximization; second, it consists of a trade-off between two terms that can be interpreted intuitively: the model is rewarded for either improving the reconstruction term, or for keeping the encoder close to the prior.

Srivastava and Sutton in [14] have proposed a VAE for topic modeling (AVITM) that leverages a Laplace approximation of the usual Dirichlet prior to permit its integration into the autoencoder. In AVITM, both the prior and the encoder are modeled as logistic normal distributions: the prior is modeled as $p(\theta|\alpha) = \mathcal{LN}(\theta|\mu(\alpha), \Sigma(\alpha))$, and the encoder is modeled as $q(\theta|w) = \mathcal{LN}(\theta|f_\mu(\phi, w), f_\Sigma(\phi, w))$, where ϕ are the internal parameters of two neural networks that predict the mean and covariance of the encoder, respectively. The expectation in Equation (5.7) is computed by sampling $q(\theta|w)$, which in turn is performed through reparametrization. The decoder takes the following form:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta)\theta) \tag{5.8}$$

where $\sigma()$ is the softmax operator, and the word distributions are parametrized as logits rather than in the simplex to remove unnecessary constraints during backpropagation. The authors have also proposed a second, heuristic version of the decoder, called ProdLDA, that performs the product before the softmax:

$$p(w|\theta, \beta) = \text{Mult}(w|\sigma(\beta\theta)) \tag{5.9}$$

As shown in [14], both AVITM and ProdLDA have outperformed a number of compared topic model approaches by large margins, and can be regarded as state-of-the-art approaches for this task.

5.3.3 The proposed approach: model training with the RE-BAR gradient estimator

Reinforcement learning has the potential to improve the performance of models beyond what can be achieved by the optimization of conventional loss functions. The main advantages of reinforcement learning are its ability to deal with non-differentiable objectives and its use of sampling, which allows exploring regions of the parameter space that may not otherwise be traversed by the optimization process. In the case of topic modelling, one way to leverage reinforcement learning is to introduce an additional variable in the model and attempt to learn an effective "policy" (a conditional probability) for it. To this aim, we introduce a new random variable, y, which is meant to represent the "main" topic of a given document (we will relax this hard assumption later in the section). With variable y, the ELBO in Equation (5.7) is changed to:

$$\begin{aligned} \mathcal{L}(w, y, \alpha, \beta) = \mathbb{E}_{q(\theta|w)}\big[\log p(w|y, \beta)\big] \\ - D_{\text{KL}}(q(\theta|w)\|p(\theta|\alpha)), \\ y \sim p(y|\theta) = \text{Mult}(y|\theta) \end{aligned} \tag{5.10}$$

where, with a slight abuse of notations, we have noted the sample and the random variable with the same symbol, y. In essence, in Equation (5.10) we sample a categorical variable, y, from a multinomial distribution of parameters θ, and we use it in lieu of θ to mix (i.e. select) the word distributions in Equations (5.8-5.9).

5.3.4 REINFORCE

The training objective of reinforcement learning is the *expected risk*:

$$R(w, \alpha, \beta) = \mathbb{E}_{p(y|\theta)}\big[\mathcal{L}(w, y, \alpha, \beta)\big] = \sum_{y} \mathcal{L}(w, y, \alpha, \beta) p(y|\theta) \tag{5.11}$$

which is the expectation of the loss function, $\mathcal{L}(w, y, \alpha, \beta)$, over the probability of variable y, the document's main topic. In order to minimize Equation (5.11), training will attempt to assign high probability to values of y that cause low values of the loss, $\mathcal{L}(w, y, \alpha, \beta)$, and vice versa, thus enforcing an effective policy. The minimization of Equation (5.11) can be performed using the policy gradient theorem [19], which ignores the dependency of the loss on the parameters and only differentiates the policy:

$$\begin{aligned} \frac{\partial}{\partial \theta} R(w, \alpha, \beta) &= \sum_{y} \mathcal{L}(w, y, \alpha, \beta) \frac{\partial}{\partial \theta} p(y|\theta) \\ &= \sum_{y} \mathcal{L}(w, y, \alpha, \beta) \frac{\partial}{\partial \theta} \log p(y|\theta) p(y|\theta) \\ &= \mathbb{E}_{p(y|\theta)}\big[\mathcal{L}(w, y, \alpha, \beta) \frac{\partial}{\partial \theta} \log p(y|\theta)\big] \end{aligned} \tag{5.12}$$

The resulting expectation is computed empirically, often from a single sample. With these approximations, we have:

$$\frac{\partial}{\partial \theta} R(w, \alpha, \beta) \approx \mathcal{L}(w, y, \alpha, \beta) \frac{\partial}{\partial \theta} \log p(y|\theta), \quad y \sim p(y|\theta) \tag{5.13}$$

The above gradient estimator is the popular REINFORCE, a key approach of reinforcement learning that does not require differentiation of the loss function in the parameters, making it applicable to a wide variety of scenarios [19]. However, REINFORCE is known for its high variance, which often compromises the stability of training. For this reason, many revised estimators have been proposed in the literature such as REINFORCE with baseline [19], actor-critic algorithms [65] and

the Gumbel-Softmax [21]. However, many of these algorithms introduce a *bias*, i.e. an average difference with respect to the exact gradient. Recently, Tucker *et al.* in [22] have proposed an alternative gradient estimator – REBAR – that is both low-variance and unbiased, and has empirically outperformed other, state-of-the-art biased and unbiased estimators. For this reason, we have adopted it in this work to boost the performance of topic modelling.

5.3.5 The REBAR gradient estimator

To introduce the REBAR gradient estimator, we first streamline the notations of the loss function and probability distribution as $\mathcal{L}(y)$ and $p(y)$, respectively, keeping the dependency on the parameters implicit and leaving only the required dependencies explicit. Next, we make use of the "Gumbel-Max trick" to obtain the samples for y [67], [68]. The Gumbel-Max trick allows sampling categorical variables such as y by manipulating samples of the Gumbel distribution, which in turn can be obtained from an inverse transform of uniform samples. Concisely, the following properties hold:

$$\begin{aligned} y &= \text{argmax}(s) \\ s &= \sigma\big([\log\theta - \log(-\log u)]/\tau\big) \\ u &\sim \mathcal{U}(0,1)^K \end{aligned} \tag{5.14}$$

Operationally, in Equation (5.14) we first sample a vector of K uniformly-distributed numbers in the $(0,1)$ interval. Then, via an inverse transform and a softmax with temperature (noted as $\sigma(\cdot/\tau)$), a "soft" version of variable y is obtained, s, encoded as a vector of K values in the $K-1$-simplex. Eventually, $y = \text{argmax}(s)$ returns the index of the largest value of s, which is provenly equivalent to directly sampling y from $\text{Mult}(y|\theta)$. The reason for this seemingly complex manipulation is that variable s, obtained as a by-product, will be utilized in the REBAR gradient estimator.

The standard REINFORCE estimator can be improved by introducing a suitable

"baseline" to condition the sign and magnitude of the gradient updates. By referring to the baseline as b, REINFORCE with baseline can be expressed as:

$$\begin{aligned}\frac{\partial}{\partial\theta}R(w,\alpha,\beta) &= \frac{\partial}{\partial\theta}\mathbb{E}[\mathcal{L}(y)-b+b] \\ &\approx [\mathcal{L}(y)-b]\frac{\partial}{\partial\theta}\log p(y) + \frac{\partial}{\partial\theta}\mathbb{E}[b]\end{aligned} \tag{5.15}$$

The first term in (5.15) determines the sign of the gradient for the policy update: only if the loss caused by y, $\mathcal{L}(y)$, is greater than b (i.e. a particularly bad value), the training iteration will decrease $p(y)$. Otherwise, it will increase it or leave it unchanged. The second term ensures that the estimator is unbiased, i.e. has the same expected value as the exact gradient.

The intuition behind REBAR is that an effective baseline can be obtained by "relaxing" the mixing variable, y, conditionally to its observed value. The relaxed variable, noted as $\tilde{s}$, is sampled from a truncated Gumbel distribution:

$$\begin{aligned}\tilde{s} \sim p(\tilde{s}|y) &= \sigma(\text{TruncatedGumbel}(\theta,T)/\tau) \\ v &\sim \mathcal{U}(0,1)^K\end{aligned} \tag{5.16}$$

In Equation (5.16) we, again, first sample a vector of K uniformly-distributed numbers in the $(0,1)$ interval. Notation $\text{TruncatedGumbel}(\theta,T)$ denotes a sample from a truncated Gumbel distribution of mean θ and threshold T. The threshold at which the Gumbel distribution is truncated is $T = -\log(-\log v_y))$, where v_y is the element of v at index y. This truncation ensures that $\text{argmax}(\tilde{s}) = y$ by construction, as required by the conditional probability $p(\tilde{s}|y)$. For the full details of the derivation of $\tilde{s}$, we refer the reader to [22], [69].

Once y, s and $\tilde{s}$ have all been derived, the REBAR gradient estimator can be finally computed as:

$$[\mathcal{L}(y) - \eta\mathcal{L}(\tilde{s})]\frac{\partial}{\partial\theta}\log p(y) + \eta\frac{\partial}{\partial\theta}[\mathcal{L}(s) - \mathcal{L}(\tilde{s})] \tag{5.17}$$

In (5.17), term $\eta\mathcal{L}(\tilde{s})$ is the baseline, with η a positive hyperparameter tunable by cross-validation. The second term in the equation, $\eta\frac{\partial}{\partial\theta}[\mathcal{L}(s) - \mathcal{L}(\tilde{s})]$, ensures the unbiasedness of the overall estimator [22], [69]. From the gradient estimator, we can also backderive an expression for a loss that can be used with common automatic differentiation and backpropagation tools:

$$\mathcal{L}_{REBAR} = [\mathcal{L}(y) - \eta\mathcal{L}(\tilde{s})]_{nogr}\log p(y) + \eta[\mathcal{L}(s) - \mathcal{L}(\tilde{s})] \tag{5.18}$$

where subscript *nogr* states that the subscripted term should not be differentiated.

The original VAE loss (5.7) and the REBAR loss (5.18) can also be conveniently mixed, to explore trade-offs between the two. We therefore define the overall loss as:

$$\mathcal{L}_{overall} = \mathcal{L}(w, \alpha, \beta) + \epsilon\,\mathcal{L}_{REBAR} \tag{5.19}$$

with ϵ a positive hyperparameter tunable with validation techniques.

5.3.6 Summary of the operational steps

For clarity, the following boxed list recapitulates all the main steps of the proposed approach:

Operational steps

1. Sample u and v, two vectors of random numbers from the uniform distribution in (0,1), each with a number of elements equal to the number of topics, K.

2. Obtain the "soft" prediction for the topic of the current document, s, by computing:
$$s = \sigma\big([\log\theta - \log(-\log u)]/\tau\big)$$
where:
   - θ are the probabilities of the topics for the current document, sampled from the encoder network, $q(\theta|w)$;
   - τ is the chosen temperature;
   - σ is the softmax operator;
   - s is a vector of K elements in the probability simplex, Δ_{K-1}.

3. Obtain the actual prediction for the topic of the current document, y, by computing:
$$y = \text{argmax}(s)$$
where y is the index of the largest value of s.

4. Obtain the "soft" prediction conditioned on y for the topic of the current document, $\tilde{s}$, by computing:
$$\tilde{s} = \sigma(\text{TruncatedGumbel}(\theta, T)/\tau)$$
where distribution TruncatedGumbel(θ, T) uses vector v as input and parameter T as threshold. By construction, argmax$(\tilde{s}) = y$. All the details of this manipulation can be retrieved from [69], Appendix B.

5. Compute objective $\mathcal{L}(\cdot)$ in Equation (5.10) with arguments y, s, and $\tilde{s}$, respectively.

6. Compose the REBAR loss function:
$$\mathcal{L}_{REBAR} = [\mathcal{L}(y) - \eta\mathcal{L}(\tilde{s})]_{nogr}\log p(y) + \eta[\mathcal{L}(s) - \mathcal{L}(\tilde{s})]$$
where η is a positive coefficient, $p(y)$ is the value of θ indexed by y, and subscript *nogr* states that the subscripted term should not be differentiated.

7. Combine the REBAR loss function with the original VAE loss (5.7):
$$\mathcal{L}_{overall} = \mathcal{L}(w, \alpha, \beta) + \epsilon\,\mathcal{L}_{REBAR}$$
with ϵ a positive trade-off coefficient.

8. Lastly, automatically differentiate $\mathcal{L}_{overall}$ with any common autodiff libraries such as TensorFlow, PyTorch or JAX [70]–[72].

5.4 Experiments and Results

5.4.1 Datasets

For the experiments, we have employed two popular document datasets, namely 20 Newsgroups (regarded as a standard benchmark for the field) and Amazon Fine Foods Reviews. The 20 Newsgroups dataset consists of 18,747 documents collected from twenty different newsgroups and split over 11,259 documents as training set and 7,488 as test set. The average number of tokens per document for this dataset is approximately 86.5. As the vocabulary, we have used the 1,995 most-frequent words publicly shared by [14] and the same pre-processing for direct comparability of the results. Amazon Fine Foods Reviews is a much larger dataset consisting of 568,454 food reviews posted by Amazon users and collected over a period of 10 years (up to October 2012). For our experiments, we have used the plain-text review field, with 454,763 reviews as training set and 113,691 reviews as test set. The average number of tokens per document for this dataset is approximately 36.3. As vocabulary, we have retained the most-frequent 5,000 words. For this dataset, the raw documents have been preprocessed with a combination of tokenization, stopword removal, stemming and lemmatization; special characters and punctuation have also been removed, and the pre-processed documents have been converted to NumPy arrays for input into the various topic models. All models have been coded in Python 3, and the variational-autoencoder topic models have been implemented in the TensorFlow 1.X framework. For processing, we have used an Intel Xeon node with 8 cores and 64 GB of RAM.

5.4.2 Experimental set-up

To evaluate the comparative performance of the proposed approach, we have included: 1) LDA and LSI from Gensim [60] as baselines; 2) the hierarchical Dirichlet process (HDP), a much-cited, hierarchical, Bayesian nonparametric topic model [41]; and 3) the state-of-the-art AVITM and ProdLDA models [14]. In this section, we compare these methods with corresponding versions of our REBAR approach,

namely AVITM-REBAR and ProdLDA-REBAR.

For training our model, we have used the following values for the hyperparameters: the learning and dropout rates have been set to the same values (0.001 and 0.25, respectively) used in [14]. The temperature hyperparameter, τ, has been set to the same value (2.25) used in [66]. For the other two hyperparameters, η and ϵ, we have chosen ranges and values based on an initial sensitivity analysis. For η, we have explored values in the $[0.5, 2.5]$ range in 0.5 steps, and reported the results for $\eta = 2.0$. For ϵ, we have used values $\{1e-10, 1e-12, 1e-14\}$ since the reinforcement learning objective, $\mathcal{L}_{REBAR}$, is much larger in scale than the VAE objective, and reported the results for $\epsilon = 1e-12$. As number of training epochs, for the 20 Newsgroups dataset we have used the same number (200) used in [66], while for Amazon Fine Food Reviews we have limited it to 40, as the training set is much larger and each epoch takes about 100x a 20 Newsgroups epoch. To explore the impact of initialization and sampling, we have also initially carried out multiple training runs for a few of the models, and noted that the variations on all performance figures were within 0.5 pp in all cases, thus not altering the ranking of the compared approaches.

For performance evaluation of the trained models, we note that it is common practice for the topic modelling field to report performance also over the training set themselves. This is because topic modelling is quintessentially a *descriptive* (rather than predictive) task that aims to best describe an assigned collection of documents. However, in addition to the results over the training sets, in this section we also include performance measures and results over the given test sets. As number of topics, we have used the rather common values of 20 and 50 for both datasets. For performance evaluation, we have used two widely-used measures:

- *perplexity*: the perplexity of a model over a set S is defined as: perplexity$(S) = \exp(-\mathcal{L}(S)/(\text{number of tokens in } S))$. In the general case, $\mathcal{L}$ denotes the log-likelihood of the data, but for the variational methods (all except LSI in our case), it is given by the ELBO in Equation (5.7). The perplexity is a measure of the "poorness of fit" of the model on the data

(the lower, the better). To assess the models' generalization, we report the perplexity over the test sets.

- *topic coherence*: topic coherence quantifies the coherence of a topic by measuring how often its top K words co-occur within a text window that slides across the documents (the higher the co-occurrence, the better). Since this measure is not uniquely defined, we report both the normalized pointwise mutual information (`Coher-NPMI`) [43] and the C_V coherence (`coher-Cv`) [44] using their Gensim implementation. The coherence is typically measured on the training set itself since this guarantees the presence of all the top words. For the experiments, K has been set to 10. For all the variational methods, the top words per topic have been selected as those with highest probability in the term-topic matrix. For LSI, they have been selected as those with highest weight in the term-topic matrix.

Perplexity is, essentially, a measure of fit of the model, while topic coherence is a measure of the quality of the extracted topics and may better reflect the user's perception of performance. Given their significantly different nature, some disagreement in model ranking between perplexity and topic coherence is to be expected. For this reason, for comparing the models we resort to a majority criterion, with emphasis on topic coherence.

5.4.3 Main results

Tables 5.1 and 5.2 show the results for the 20 Newsgroups dataset for 20 and 50 topics, respectively. In the tables, the perplexity values for LDA cannot be directly compared with those of the autoencoder models because of the differing architecture and number of degrees of freedom; therefore, they are marked in italics. LSI is not a probabilistic model, so its perplexity values are unavailable, and also the log-likelihood returned by the HDP is not easily convertible to a perplexity. In terms of performance, both LDA and LSI have reported much lower coherence compared to the other models, and therefore cannot be regarded as competitive. The HDP has

Table 5.1: Results on the 20 Newsgroups dataset with 20 topics (suffix "REBAR" is abbreviated as "RB").

Measure/Model	LDA	LSI	HDP	AVITM	ProdLDA	AVITM-RB	ProdLDA-RB
Perplexity	*1516.2*	—	—	1140.2	1173.3	**1139.7**	1165.5
Coher-NPMI	-0.060	-0.060	0.027	0.094	0.141	0.133	**0.147**
Coher-Cv	0.383	0.356	0.448	0.671	0.779	0.742	**0.805**

Table 5.2: Results on the 20 Newsgroups dataset with 50 topics.

Measure/Model	LDA	LSI	HDP	AVITM	ProdLDA	AVITM-RB	ProdLDA-RB
Perplexity	*2531.4*	—	—	1133.1	1159.9	**1130.2**	1160.9
Coher-NPMI	-0.084	-0.062	0.017	0.117	0.111	0.104	**0.143**
Coher-Cv	0.348	0.351	0.432	0.704	0.751	0.693	**0.778**

Table 5.3: Results on the Amazon Fine Food Reviews dataset with 20 topics.

Measure/Model	LDA	LSI	HDP	AVITM	ProdLDA	AVITM-RB	ProdLDA-RB
Perplexity	*1426.0*	—	—	1000.9	1099.7	**1000.2**	1098.3
Coher-NPMI	0.081	0.004	0.011	0.144	0.066	**0.152**	0.118
Coher-Cv	0.564	0.395	0.419	0.707	0.651	**0.715**	0.710

Table 5.4: Results on the Amazon Fine Food Reviews dataset with 50 topics.

Measure/Model	LDA	LSI	HDP	AVITM	ProdLDA	AVITM-RB	ProdLDA-RB
Perplexity	*2789.9*	—	—	1008.6	1012.5	**1007.3**	1009.0
Coher-NPMI	0.076	-0.009	0.011	0.144	-0.048	**0.149**	0.047
Coher-Cv	0.551	0.360	0.420	0.682	0.430	**0.691**	0.585

achieved higher coherence than both LDA and LSI, but still substantially lower than all the variational autoencoder models. Between AVITM and ProdLDA, AVITM has achieved better (i.e., lower) perplexity values, while ProdLDA has achieved better (i.e., higher) coherence values in the majority of cases. Notably, our REBAR models have been able to attain a marked improvement over both AVITM and ProdLDA: compared to AVITM, AVITM-REBAR has improved both perplexity and coherence in the case of 20 topics, and perplexity in the case of 50 topics; compared to ProdLDA, ProdLDA-REBAR has improved both perplexity and coherence in the case of 20 topics, and coherence in the case of 50 topics. In addition, AVITM-REBAR and ProdLDA-REBAR have achieved the overall best perplexity and coherence, respectively.

In turn, Tables 5.3 and 5.4 show the results on the Amazon Fine Food Reviews dataset for 20 and 50 topics, respectively. Again, LDA, LSI and the HDP have achieved markedly lower coherence values than the autoencoder models. However, on this dataset LDA has performed better than both LSI and the HDP, reaching a `Coher-NPMI` value even higher than that of ProdLDA. Between AVITM and ProdLDA, AVITM has achieved the best results in terms of both perplexity and coherence. However, AVITM-REBAR has outperformed AVITM in all measures for both 20 and 50 topics, and achieved the best performance of all models. Based on the results on both 20 Newsgroups and Amazon Fine Food Reviews, we can conclude that our REBAR-based models have outperformed all the compared models.

5.4.4 Ablation, sensitivity and qualitative analysis

As an ablation analysis, we compare the performance of the proposed model with that of two ablated versions:

1. a standard implementation of the REINFORCE gradient estimator, without any baseline: $\mathcal{L}(y)_{nogr} \log p(y)$;

2. an implementation of REINFORCE with the same baseline used by REBAR to limit the variance, but without the offset term that maintains the gradient

Table 5.5: Ablation analysis for ProdLDA-REBAR (20 Newsgroups dataset, 50 topics).

Measure/model	REINFORCE	REINFORCE (baseline)	REBAR
Perplexity	1163.5	1161.4	**1160.9**
Coher-NPMI	0.106	0.112	**0.143**
Coher-Cv	0.724	0.742	**0.778**

Table 5.6: Ablation analysis for AVITM-REBAR (20 Newsgroups dataset, 50 topics).

Measure/model	REINFORCE	REINFORCE (baseline)	REBAR
Perplexity	1133.6	1131.7	**1130.2**
Coher-NPMI	**0.110**	0.094	0.104
Coher-Cv	0.687	0.665	**0.693**

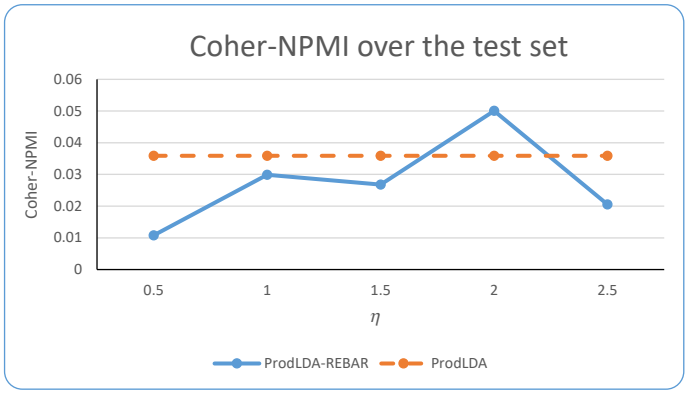

Figure 5.2: Comparison of `coher-NPMI` on the test set for ProdLDA and ProdLDA-REBAR (20 Newsgroups, 50 topics) by varying hyperparameter η.

estimator unbiased:

$[\mathcal{L}(y) - \eta\mathcal{L}(\tilde{s})]_{nogr} \log p(y)$.

Tables 5.5 and 5.6 show the results of the ablation analysis for ProdLDA-REBAR and AVITM-REBAR, respectively, over the 20 Newsgroups dataset with 50 topics. The results show that the complete approach has outperformed the ablated versions in almost all cases. The addition of the baseline to REINFORCE has generally improved the performance of ProdLDA-REBAR, but not of AVITM-REBAR. This is evidence that the gradient estimator benefits from being both low-variance and unbiased, as ensured by REBAR [22].

As sensitivity analysis, we have repeated the experiments with ProdLDA-REBAR

Table 5.7: Results for ProdLDA-REBAR on the 20 Newsgroups dataset with 50 topics, with variable η hyperparameter.

Measure/η	0.5	1.0	1.5	2.0	2.5
Perplexity	1161.9	1163.3	1161.7	**1159.1**	1159.2
Coher-NPMI	0.112	0.129	0.117	**0.137**	0.118
Coher-Cv	0.732	0.777	0.750	**0.780**	0.750

Table 5.8: Results for ProdLDA-REBAR on the 20 Newsgroups dataset with 50 topics, with variable temperature hyperparameter, τ.

Measure/τ	10^{-5}	1.5	1.75	2.0	2.25	2.5	10
Perplexity	1162.9	**1160.2**	1162.9	1163.3	1163.7	1162.6	1161.6
Coher-NPMI	0.111	0.100	0.123	0.129	**0.137**	0.105	0.121
Coher-Cv	0.734	0.720	0.764	0.760	**0.772**	0.724	0.768

on the 20 Newsgroups dataset for 50 topics by varying the values of the η and τ hyperparameters. Table 5.7 shows the results with η varying in the $[0.5, 2.5]$ range in 0.5 steps. Interestingly, the model has achieved both the best perplexity and the best coherence for $\eta = 2.0$. However, the results have proved very sensitive to the η value, and there is a risk that the coherence would drop if measured over an independent test set. For this reason, in Fig. 5.2 we report the values of the C_V coherence over the test set for the same values of η. The plot shows that the best C_V coherence has been attained for $\eta = 2.0$, the same value as the best C_V coherence for the training set. We regard this result as encouraging evidence of generalization. In turn, Table 5.8 shows the results with ProdLDA-REBAR for the temperature hyperparameter, τ, varying in the $[1.5, 2.5]$ range in 0.25 steps, and for values 10^{-5} (approximately one-hot samples) and 10 (approximately uniform samples). The best result in terms of coherence have been obtained with $\tau = 2.25$, alongside a very modest worsening of the perplexity, confirming the indications from [66]. To further probe the generalization of the model, Figure 5.3 compares the behavior of the training loss, $\mathcal{L}_{overall}$ in Equation (5.19), at successive training epochs with that of the perplexity over the test set at the same epochs. The plots show that the perplexity over the test set nicely decreases as training progresses, showing no evidence of overfitting. This confirms that the proposed training objective has been able to achieve good generalization over unseen data.

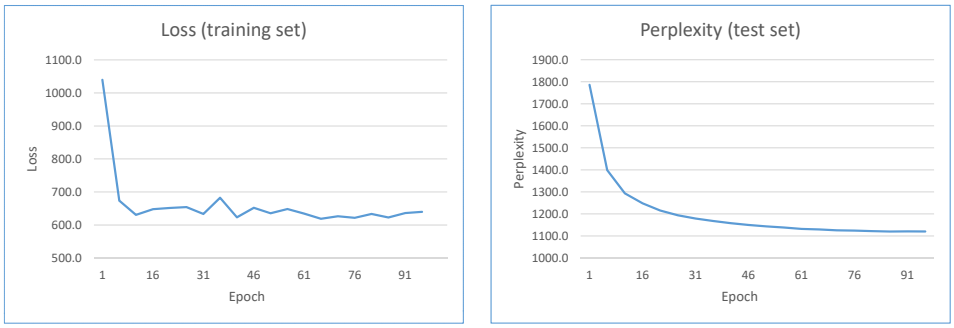

Figure 5.3: Comparison of the behavior of the training loss and the test-set perplexity (20 Newsgroups, 50 topics). Left: The values of the training loss function, $\mathcal{L}_{overall}$, at successive training epochs. Right: The values of the test-set perplexity at the same epochs.

Table 5.9: Examples of topics extracted from the Amazon Fine Food Reviews dataset (50 topics).

LDA: chip bag potato open plant come vinegar small littl kettle gum fun shoot counter respons bewar edg xylitol wide buyer coffe cup tast like flavor good strong brew pod great
AVITM: snack chip salti salt cracker bag jerki potato theyr crunchi salt chip salti jerki potato cracker sea bbq vinegar spici coffe cup brew pod roast keurig bean machin bold maker
AVITM-REBAR: sauc soup cook noodl spici salad dish rice pepper pasta chip salt salti popcorn pop cracker potato vinegar sea cheddar coffe cup bold keurig roast bitter brew smooth strong french

Finally, for a qualitative analysis, Table 5.9 shows the top $K = 10$ words for a few example topics extracted by LDA, AVITM and AVITM-REBAR (the best model in terms of quantitative measures) from the Amazon Fine Food Reviews dataset with 50 topics. For LDA, word "kettle" in the first topic seems to be an outlier. In addition, the second topic seems of very difficult interpretation and might be regarded as an example of unsuccessful extraction. For AVITM, the extracted topics look generally good; however, word "theyr" in the first topic seems to be an uninformative outlier. In the case of AVITM-REBAR, all the example topics look informative and consistent. While this analysis is not exhaustive, it shows a promising alignment between strong quantitative measures and appealing qualitative results.

Table 5.10: Comparison of the various improvements over ProdLDA.

Model	Perplexity	Coher-NPMI	Coher-CV
ProdLDA-REINF (Ch. 3)	1162.8	0.141	0.763
ProdLDA-GS (Ch. 4)	1136.6	0.148	0.806
ProdLDA-RB (Ch. 5)	1160.8	0.143	0.778

5.5 Conclusion

In this chapter, we have proposed an approach for neural topic modelling that leverages a reinforcement learning objective (i.e., the predictive risk) and the recently-proposed REBAR gradient estimator. The predictive risk objective has allowed us to make use of the reinforcement learning framework, while the REBAR gradient estimator has provided us with a solution that is both low-variance and unbiased. The proposed approach has been integrated in a deep variational-autoencoder topic model (AVITM/ProdLDA) that can be regarded as the previous state of the art [14]. Experiments carried out over two topic modelling datasets (20 Newsgroups and Amazon Fine Food Reviews) have given evidence to the strong comparative performance of the proposed approach, with marked improvements in all the reported measures (perplexity, normalized pointwise mutual information, and C_V coherence) for both datasets. As future work, we plan to explore the use of the REBAR gradient estimator for other NLP tasks that are mainly unsupervised such as taxonomy extraction, ontology creation and knowledge graph construction.

5.5.1 Comparison across chapters: REINFORCE vs Gumbel-Softmax vs REBAR

In this and the previous two chapters, we have proposed three independent improvements over the state-of-the-art model ProdLDA. Within the fuller scope of the thesis, it is certainly worth comparing them where possible. To this aim, Table 5.10 shows the results for REINFORCE, Gumbel-Softmax and REBAR over 20 Newsgroups with 50 topics (the only case in common). It is interesting to see that the Gumbel-Softmax has quite neatly outperformed the other two. Assuming that this result

can be extrapolated to other cases, it would show that the simple Gumbel-Softmax, despite its bias as a gradient estimator, has proved more effective than the unbiased and more sophisticated REBAR. A possible way to improve the latter could then be to experiment with various values of its Gumbel-Softmax' temperature parameter. We leave this to future work.

Chapter 6

The Contextualized Regressive Topic Model

Topic modelling is a popular natural language processing task which automatically extracts the main topics from a collection of documents, concurrently identifying the topic proportions of each document. For simplicity and efficiency, most conventional topic models still use the bag-of-words (BoW) representation to represent the documents, but more recent models have started to leverage embedded document representations to capture their context more fully. However, none of the existing models has incorporated the embedded representations directly in the training objective of the topic model. For this reason, in this chapter we propose training a state-of-the-art variational autoencoder topic model by simultaneously reconstructing the BoW and a BERT-based embedded representation of the documents. Experiments performed over three diverse datasets have shown that the proposed model — nicknamed the Contextualized Regressive Topic Model (CRTM) — has been able to outperform its BoW counterpart and well-established baselines on all datasets and performance measures.

6.1 Introduction

The recent years have witnessed a rapid increase in the amounts of textual data, which are no longer generated only by conventional sources such as publications and documents, but also by digital sources such as podcasts, blogs, social media posts, speech-to-text converters and so forth. This growth has led to substantial challenges in their analysis, exploration, manipulation, and eventual organization. While all these tasks could theoretically be performed by manual annotators, they are prohibitively time-consuming, subjective to a significant extent, and prone to distraction and fatigue errors. Therefore, computational approaches, preferably unsupervised or weakly supervised, are more needed than ever to tackle the complexity of large textual data collections.

Amongst the unsupervised approaches, *topic models* have established a strong reputation for their ability to identify meaningful patterns in large amounts of unstructured textual data. In simple words, a topic model extracts the shared "topics" from a given document collection, and simultaneously assigns each document in the collection to its respective topics in proportion. The extracted topics help the users understand the overall focus of the collection, while the topic memberships assigned to the individual documents assist their categorization and organization. Thanks to their flexibility and effectiveness, topic models have found useful application in domains as diverse as healthcare [2]–[4], finance [5], [6], agriculture [7], social media [6], [8], news [9] and many others.

Many topic models have been proposed to date, the most famous of which is probably the Latent Dirichlet Allocation (LDA) of Blei et al. [12], a probabilistic model where both the shared topics and the topic proportions of the individual documents are modelled with Dirichlet distributions. Many topic models have evolved from LDA, including models based on variational autoencoders [14], [16], [73] which currently hold the state of the art in performance. All these models typically convert the individual documents into a so-called bag-of-words (or BoW, concisely) representation, which is simply a histogram of the occurrences in the document of the distinct words of a given vocabulary. While such a representation is able to "cap-

ture" the fundamental information of a document, it neglects both the order and the context in which the document's words appear. For this reason, more recent approaches such as the contextualized topic model (CTM) of Bianchi et al. [46] have enriched the representation of the individual documents with contextualized embeddings obtained from language models such as BERT [74], showing remarkable performance improvements. The goal of our chapter is to explore how contextualized representations can be more fully integrated into topic models. In detail, the main contributions of our chapter are:

- We propose a novel training objective that minimizes the distance between a pre-computed embedding for the document and an embedding predicted by the topic model.
- We explore the effectiveness of different distance functions in the training objective, including the Euclidean distance, the Manhattan distance, the Minkowski distance of order three, and the cosine distance.
- We apply the proposed training objective to a state-of-the-art variational encoder topic model, CTM [46], and carry out extensive experiments over three topic modelling datasets (Wiki20K, 20 Newsgroups, and Amazon Fine Food Reviews). The experimental results show that the proposed approach has invariably led to performance improvements over the compared approaches.

The remainder of this chapter is organized as follows: the related work is presented in Section 6.2, including a brief review of the main topic models. Variational autoencoder topic models are recapped in greater detail in Section 6.3.1, while the proposed approach is presented in Section 6.3.2. The experiments and results are presented in Section 6.4. Eventually, the conclusion is given in Section 6.5.

6.2 Related work

In this section, we first review the topic models that are closely related to the proposed work, and then we review contextual representations and their prior use in topic modelling.

The fundamental quantities in topic modelling are the given collection of D documents and the chosen vocabulary of V distinct words. The documents are tokenized, and the n-th word in the d-th document is noted as $w_{d,n}$ and treated as a categorical variable in index set $[1 \dots V]$. With these assumptions, topic modelling can be framed in either of two equivalent ways: as a matrix factorization problem, or as a statistical model. In the matrix factorization view, a bag-of-words (the histogram of occurrences of the V distinct words) is first formed for each document, and then the bags-of-words of all documents are concatenated together into a matrix of $V \times D$ size, known as the term-document matrix. The goal of topic modelling is to factorize the term-document matrix into two matrices: the term-topic matrix, β, of size $V \times K$, where $K \ll D$ is the chosen number of topics, and the topic-document matrix, Θ, of size $K \times D$. The K columns of β are interpreted as the "topics" (i.e., specific weights over the vocabulary), and the D columns of Θ as the topic proportions for each of the individual documents. In the statistical view, the columns of β and Θ are constrained to be proper categorical probability distributions (i.e., elements bound between 0 and 1, and sum equal to 1; the "simplex" domain). In addition, it is possible to include prior distributions over the columns of both β and Θ. The famous Latent Dirichlet Allocation (LDA) uses Dirichlet priors over Θ and, optionally, over β. The Dirichlet distribution is conjugate to the categorical distribution, and therefore the posterior distributions can be conveniently computed in closed form. LDA has established a strong reputation for performance and has de-facto become a "workhorse" for the field. It has also spawned a huge number of extensions and variants, including class-supervised versions [26], hierarchical versions [13], [33], sequential versions [26], sparse versions [28]–[30], and many others. However, the extensions that are most relevant to our work are those based on variational autoencoders and contextual representations [14], [16], [46]. For this reason, we briefly review them hereafter.

A variational autoencoder (VAE) is a generalization of an autoencoder, an unsupervised neural network containing an encoder and a decoder [15]. The encoder takes a measurement, x, in input and generates a latent representation, z, as output, while the decoder takes z as input and generates a "reconstruction" of the input measure-

ment, $\tilde{x}$. In general, $\tilde{x}$ is not identical to x because the latent representation is not invertible. The rationale of an autoencoder is that the reconstructed measurement, $\tilde{x}$, or even the latent representation itself, z, can lead to higher accuracies than the original measurement, x, in downstream tasks. VAEs have been successfully employed in a range of domains, including for topic modelling, where both the topics and the topic proportions are mapped to latent variables. Miao et al. in [16] have introduced a VAE topic model that uses a Gaussian distribution as the prior over the logits of the topic proportions. In turn, Srivastava and Satton in [14] have proposed an improvement over [16] using an approximation of a Dirichlet prior. Their model, called ProdLDA, is still one of the state-of-the-art models for the field.

In general, topic models have proved to be an effective and viable technology. However, a standing limitation is their prevalent use of BoW representations as input, which dismisses both the order and the context of the words in the document. In addition, categorical representations of words fail to capture the similarity between words of similar meaning. To leverage contextuality in topic modelling, it is possible to represent the words with popular embeddings such as word2vec, GloVe, fastText and others [75]. These embeddings ensure that words of similar meaning have similar representations, and they are able to capture an average context. For instance, [76] has proposed factorizing the term-topic matrix, β, into a word and topic embeddings matrices. In the more recent years, *contextualized embeddings* such as ELMo, FLAIR, BERT and others have been able to also capture the context at word level and have become dominant choices for the representation of text [77]. In particular, the Bidirectional Encoder Representations from Transformers (BERT) pre-trained language models [74] have achieved state-of-the-art results in a range of NLP application for their ability to embed both the sequential and the contextual information of the words. Another key advantage of these models is that they can be pre-trained in a completely unsupervised way on large amounts of unannotated text, while at the same time being easily fine-tunable for specific downstream tasks. However, there has been little research to date integrating contextualized embeddings and topic models. As an exception, the contextualized topic model (CTM) of Bianchi et al. [46] has integrated Sentence-BERT [78], a sentence-embedding ver-

sion of BERT, with ProdLDA, reporting significant performance improvements over ProdLDA itself. For this reason, we use it in this work as our baseline.

6.3 Methodology

6.3.1 Variational autoencoder topic models: ProdLDA

ProdLDA is a state-of-the-art topic model based on variational autoencoders [14]. Like all other probabilistic topic models, ProdLDA can be best described in the form of a *generative model*, i.e. a model that, in principle, can be used to generate "synthetic" documents by sampling from the model's distributions. In practice, the generative model is only used to describe the model, and a separate training procedure is responsible for fitting the model's parameters onto the given document collection. The generative model of ProdLDA can be expressed as:

- For the d-th document, draw a K-dimensional vector, θ_d, from a K-dimensional Gaussian distribution:

  $$\theta_d \sim \mathcal{N}(\theta|\mu(\alpha), \Sigma(\alpha))$$

  where vector θ_d represents the topic proportions of the d-th document in logit scale, and α is a reparametrization for the mean and covariance of the Gaussian distribution described later in the section.

- For each word in the d-th document, draw the word from a multinomial distribution over the vocabulary obtained by mixing the term-topic matrix β with vector θ_d and then applying the softmax operator, $\sigma()$:

  $$w_{d,n} \sim \text{Mult}(w|\sigma(\beta\theta_d))$$

The ideal training objective of a topic model such as ProdLDA would be to maximize the log-likelihood of the given documents in the model's parameters. However, this objective is very challenging to be optimized directly, and it is therefore customary to maximize an approachable lower bound, known as the Evidence Lower Bound,

or ELBO [15]. The ELBO for a single training document can be expressed as:

$$L(w, \alpha, \beta) = \mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big] - D_{\mathrm{KL}}(q(\theta|w)\|p(\theta|\alpha)) \tag{6.1}$$

The terms in equation (6.1) have the following meaning:

1. $q(\theta|w)$ is an estimator for the probability of the topic proportions for the document (represented by its bag-of-words, w) and is known as the "encoder";

2. $\log p(w|\theta, \beta)$ is the log-probability of the document given its topic proportions, θ, and the word distributions, β, and is known as the "decoder";

3. $\mathbb{E}_{q(\theta|w)}\big[\log p(w|\theta, \beta)\big]$ is the expectation of this quantity over $q(\theta|w)$ and is known as the "reconstruction term";

4. $p(\theta|\alpha)$ is a learnable prior probability for the topic proportions that is shared by the entire corpus;

5. D_{KL} is a Kullback-Leibler divergence that acts as a regularizer to keep q close to p.

In ProdLDA, both the prior and the encoder are, de facto, modeled as Gaussian distributions: the prior is modeled as $p(\theta|\alpha) = \mathcal{N}(\theta|\mu(\alpha), \Sigma(\alpha))$, where α is derived from a Laplace approximation to the Dirichlet distribution (see [14] for details); and the encoder is modeled as $q(\theta|w) = \mathcal{N}(\theta|f_\mu(\phi, w), f_\Sigma(\phi, w))$, where ϕ are the internal parameters of two neural networks that predict its mean and covariance, respectively. The expectation in (6.1) is approximated by sampling from $q(\theta|w)$. Eventually, the decoder takes the following form:

$$p(w|\theta, \beta) = \mathrm{Mult}(w|\sigma(\beta\theta)) \tag{6.2}$$

where the word distributions for each topic, stored in the columns of matrix β, are parametrized as logits rather than probabilities to remove domain constraints during backpropagation.

Given a collection of documents, ProdLDA uses a VAE training procedure to extract the shared topics, in the form of matrix β, and to identify the topic proportions of each document, in the form of matrix θ (and, indirectly, the learnable parameters ϕ and α) [14]. The trained model can also be used to infer the topic proportions of new documents by only inferring θ and leaving all other parameters unchanged.

6.3.2 The proposed approach: the contextualized regressive topic model

As mentioned in the previous sections, conventional topic models use a BoW representation for the input document, which is simply a histogram of the number of occurrences of each distinct word in the document. While this representation captures all the words in the document and their frequencies, it fails to account for the sequentiality and contextuality of the words themselves. For this reason, Bianchi et al. [46] have proposed extending the BoW representation with a document embedding obtained from BERT, and consequently named their approach the Contextualized Topic Model (CTM). The document embedding is a 768-D vector computed by pooling all the hidden states from the final layer of a BERT models that receives the document in input [78].

However, while [46] has usefully extended the input representation, it has kept the decoder and the reconstruction term of ProdLDA unchanged, still measuring the expected probability of the BoW vector alone. For this reason, in our approach we propose incorporating a BERT embedding in the decoder and the reconstruction term. In our model — named the Contextualized Regressive Topic Model (CRTM) — the V-dimensional $\beta\theta$ vector of equation (6.2) is linearly transformed to a 768-D vector. The linear transformation, noted as T, is therefore an additional matrix of $V \times 768$ learnable parameters. The output of this transformation is the embedding "predicted" by the topic model for the input document, $\bar{x}_{BERT}$, that we can compare to the actual BERT embedding for the document, x_{BERT}, in the training objective using a chosen distance measure. With these notations, the new reconstruction term can be expressed as:

$$\bar{x}_{BERT} = T\ \beta\theta$$
$$\mathcal{L}_{BERT} = \text{dist}(\bar{x}_{BERT}, x_{BERT}) \tag{6.3}$$

where as dist() we have used four different distance measures: Euclidean (or L2), Manhattan (or L1), Minkowski of order three, and the cosine distance [79].

Finally, the ProdLDA objective of equation (6.1), changed in sign, and the BERT loss of equation (6.3) can be combined with a positive coefficient, ϵ, into an overall loss to explore trade-offs between the two terms:

$$\mathcal{L}_{CRTM} = -L(w, \alpha, \beta) + \epsilon\, \mathcal{L}_{BERT} \tag{6.4}$$

In addition to training the model with a contextualized representation, the main advantages of the proposed approach are that it is fully differentiable like the original ProdLDA objective, and that its only new parameter is matrix T, with the remaining network parameters being shared and co-trained by both losses.

6.4 Experiments and Results

6.4.1 Experimental set-up

For the experiments, we have used three diverse document datasets, namely *20 Newsgroups* [45] (a benchmark for the field), *Amazon Fine Food Reviews* [18] and *Wiki20K* [46]. The 20 Newsgroups dataset consists of $11,300$ documents from news shared on social media, while Wiki20K is a collection of 20,000 English Wikipedia abstracts. Amazon Fine Food Reviews is a much larger dataset consisting of $568,454$ user-posted food reviews. Their main statistics are reported in Table 6.1, showing that the Amazon dataset is much larger in size, but 20 Newsgroups' documents are longer on average. To obtain the BoW representation for the documents, the

Table 6.1: Main statistics of the datasets used for the experiments (NB: number of tokens computed after preprocessing and with the given vocabulary size).

Datasets	Size	Avg # tokens
Wiki20K	20,000	49
20NG	11,300	134
Amazon	568,454	36

same preprocessing pipeline has been applied to all the datasets, including removing digits, punctuation, stopwords, and infrequent words, and reducing the vocabulary to a manageable size of $V = 2,000$ unique words. The BERT representation[1] has instead been computed directly from the unpreprocessed text. For performance evaluation, we have compared the proposed model against established topic models such as LDA, Latent Semantic Indexing (LSI) and the Hierarchical Dirichlet Process (HDP), as well as the state-of-the-art Contextualized Topic Model (CTM) of Bianchi et al. [46]. LSI [80] is a popular topic model that dispenses with probabilistic assumptions, while the HDP [41] is a sophisticated probabilistic model which can automatically determine the optimal number of the topics within a given bound. As performance metrics, we have used the *topic coherence* which is the de-facto standard for this task. The topic coherence measures the "coherence" of the extracted topics by computing the co-occurrence of the top N words of each topic within single documents, and should be as high as possible. Given that multiple definitions for the topic coherence exist, for the evaluation we have used both the coherence *NPMI* [43] and the coherence *Cv* [44]. As number of topics, we have used $K = 20$ and $K = 50$ since they are common choices in the literature. As number of top words per topic, we have set $N = 10$. The key hyperparameter of the proposed approach, ϵ in equation 6.4, has been explored in the range $[0 - 25]$ in 2.5 steps. All the other hyperparameters have been left to their default values.

6.4.2 Results

Tables 6.2 and 6.3 show the results over the three datasets for 20 and 50 topics, respectively. The results show that the proposed CRTM model has obtained both

[1] https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1.

Table 6.2: Results on the three datasets with 20 topics (L2 distance).

Dataset Coherence	Wiki20K NPMI	Wiki20K Cv	20NG NPMI	20NG Cv	Amazon NPMI	Amazon Cv
LDA	0.004	0.449	-0.006	0.456	0.034	0.462
LSI	0.008	0.392	-0.015	0.435	0.020	0.447
HDP	0.022	0.398	0.016	0.427	0.011	0.418
CTM	0.171	0.711	0.096	0.652	0.147	**0.673**
CRTM	**0.187**	**0.739**	**0.153**	**0.712**	**0.148**	0.672

Table 6.3: Results on the three datasets with 50 topics (L2 distance).

Dataset Coherence	Wiki20K NPMI	Wiki20K Cv	20NG NPMI	20NG Cv	Amazon NPMI	Amazon Cv
LDA	0.031	0.521	0.024	0.496	0.040	0.498
LSI	-0.037	0.328	-0.068	0.354	-0.009	0.362
HDP	0.011	0.378	0.024	0.443	0.012	0.423
CTM	0.181	0.714	0.096	0.652	0.129	0.641
CRTM	**0.182**	**0.731**	**0.133**	**0.711**	**0.134**	**0.649**

the highest coherence NPMI and coherence Cv in all but one settings. The CTM model has obtained the second-best results, while the values for LDA, LSI and the HDP have been comparatively much lower. These results confirm the importance of using contextualized representations for topic modelling, and show that the proposed training objective has noticeably improved the performance of the original CTM which only uses the BoW in the training objective. The performance improvement has been more limited for the largest dataset (Amazon Fine Food Reviews), suggesting that the proposed training objective has acted as a "regularizer" for the standard BoW objective, and that the benefit becomes more pronounced for datasets of smaller size.

As a second experiment, we have performed a comparison of four distances for the objective of equation (6.3), namely the Manhattan distance (L1), the Euclidean distance (L2), the Minkowski distance of order 3 (L3) and the cosine distance. To make the comparison less dependent on the dataset size, in this experiment we have limited the size of the Amazon dataset to 20K documents. Tables 6.4 and 6.5 report the results for 20 and 50 topics, respectively. With 20 topics, L2 has performed the best in the majority of cases, followed by either L3 or L1 depending on the dataset,

Table 6.4: Comparison of different distances on the three datasets with 20 topics (NB: 20K documents for Amazon).

Dataset Coherence	Wiki20K NPMI	Wiki20K Cv	20NG NPMI	20NG Cv	Amazon NPMI	Amazon Cv
L1 Distance	0.170	0.712	0.145	0.709	0.080	0.589
L2 Distance	**0.187**	**0.739**	**0.153**	**0.712**	**0.118**	0.638
L3 Distance	0.185	0.730	0.129	0.693	**0.118**	**0.646**
Cosine Dist.	0.146	0.678	0.100	0.650	0.098	0.634

Table 6.5: Comparison of different distances on the three datasets with 50 topics (NB: 20K documents for Amazon).

Dataset Coherence	Wiki20K NPMI	Wiki20K Cv	20NG NPMI	20NG Cv	Amazon NPMI	Amazon Cv
L1 Distance	**0.197**	**0.746**	**0.146**	**0.715**	0.034	0.489
L2 Distance	0.182	0.731	0.133	0.711	0.093	0.575
L3 Distance	0.184	0.736	0.119	0.676	0.102	0.582
Cosine Dist.	0.186	0.732	0.118	0.679	**0.108**	**0.603**

while for 50 topics, L1 has performed the best for two datasets and the cosine distance for another. These results show that the selection of the best distance for a given dataset can have a significant impact on the coherence of the extracted topics.

To explore the sensitivity of the topic coherence to the ϵ hyperparameter, Fig. 6.1 shows a plot of the topic coherence for the proposed model as a function of hyperparameter ϵ for two Amazon subsets with 20K and 100K documents (20 topics, L2 distance). The plots show that the value of ϵ has a major impact on the coherence,

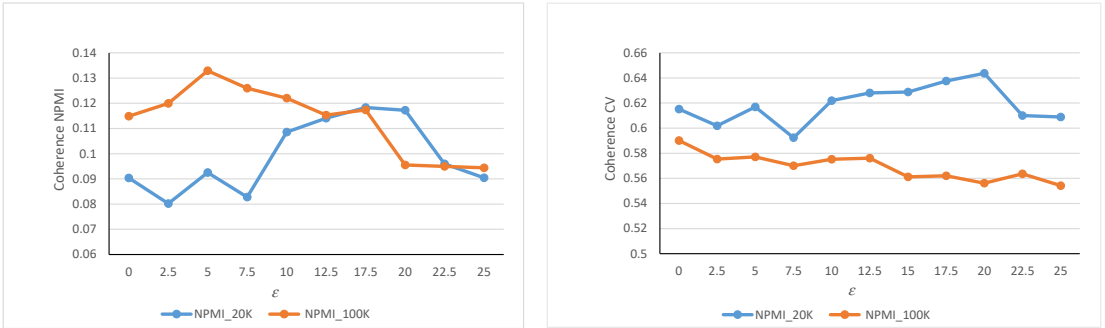

Figure 6.1: Topic coherence for the proposed model as a function of the ϵ hyperparameter (Amazon 20K and 100K, 20 topics, L2 distance). Left: coherence NPMI; right: coherence Cv.

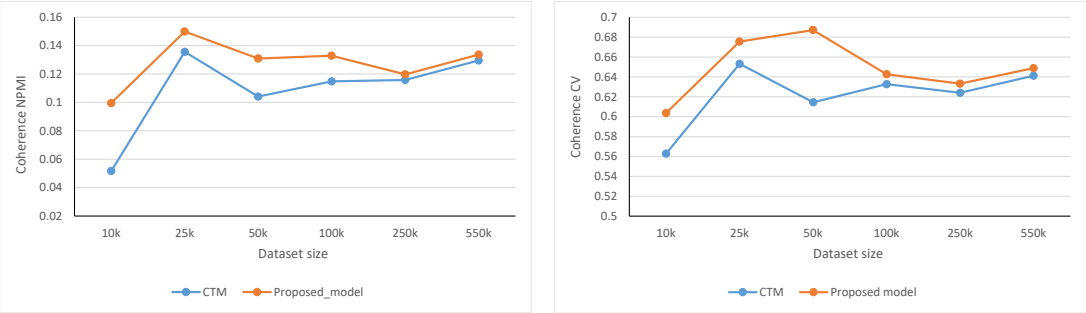

Figure 6.2: Topic coherence for the proposed model and CTM for increasing dataset sizes (Amazon dataset, 20 topics, L2 distance). Left: coherence NPMI; right: coherence Cv.

and that larger values are better for the smaller dataset, confirming the "regularizing" behavior of the proposed loss. In addition, the best values of ϵ for the coherence NPMI and the coherence Cv are similar (17.5-20 for Amazon 20K, 0-5 for Amazon 100K), making it easy to select a value that is near-optimal for both. In turn, Fig. 6.2 explores the sensitivity of the topic coherence to the dataset size by plotting the coherence of the proposed model and CTM for the Amazon dataset at an increasing number of documents, from 10K to full size (20 topics, L2 distance). The plots show that the coherence of the proposed model has been higher than that of CTM for all dataset sizes, and, as expected, that the difference in performance between the two models has been larger for smaller dataset sizes.

6.5 Conclusion

This chapter has presented the Contextualized Regressive Topic Model (CRTM), a novel topic model which extends the conventional BoW training objective of most topic models with an objective leveraging an embedded representation of the document. Experiments performed over three diverse datasets (Wiki20K, 20 Newsgroups and Amazon Fine Food Reviews) have shown that the proposed model has been able to outperform its BoW counterpart and other established topic models such as LDA, LSI and the HDP in all cases. The performance improvement has been more marked for smaller dataset sizes, suggesting that the extra objective has acted as a regularizer on the model's learning. Additional experiments over the distance function

used in the objective have shown that tuning the distance to each specific dataset can further increase the performance. In the future, we plan to investigate the use of embeddings from multilingual language models such as mT5[2] and XLM-R[3] to explore the potential for multilingual extensions.

[2]https://huggingface.co/docs/transformers/model_doc/mt5.
[3]https://github.com/facebookresearch/XLM.

Chapter 7

A Temperature-Modified Dynamic Embedded Topic Model

Topic models are natural language processing models that can parse large collections of documents and automatically discover their main topics. However, conventional topic models fail to capture how such topics change as the collections evolve. To amend this, various researchers have proposed dynamic versions which are able to extract sequences of topics from timestamped document collections. Moreover, a recently-proposed model, the dynamic embedded topic model (DETM), joins such a dynamic analysis with the representational power of word and topic embeddings. In this chapter, we propose modifying its word probabilities with a temperature parameter that controls the smoothness/sharpness trade-off of the distributions in an attempt to increase the coherence of the extracted topics. Experimental results over a selection of the COVID-19 Open Research Dataset (CORD-19), the United Nations General Debate Corpus, and the ACL Title and Abstract dataset show that the proposed model – nicknamed DETM-tau after the temperature parameter – has been able to improve the model's perplexity and topic coherence for all datasets.

7.1 Introduction

Topic models are natural language processing (NLP) models which are able to extract the main topics from a given, usually large, collection of documents. In addition, topic models are able to identify the proportions of the topics in each of the individual documents in the given collection, which can be useful for their categorization and organization. As a machine learning approach, topic models are completely unsupervised and, as such, they have proved a very useful tool for the analysis of large amounts of unstructured textual data which would be impossible to tackle otherwise. Thanks to their flexibility and ease of use, topic models have found application in domains as diverse as finance [5], [6], news[9], agriculture [7], social media [6], [8], healthcare [2]–[4] and many others.

Among the topic models proposed to date, latent Dirichlet allocation (LDA)[12] is broadly regarded as the most popular. Its simple, fundamental assumption is that every word in each document of the given collection is associated with a specific "topic". In turn, a topic is represented simply as a dedicated probability distribution over the words in the given vocabulary. Completed by a Dirichlet prior assumption over the topic proportions of each document, LDA has proved at the same time accurate and efficient. However, conventional topic models such as LDA are unable to analyse the sequential evolution of the topics over different time frames. This could be important, instead, for collections that exhibit substantial evolution over time. For instance, a collection of COVID-19-related articles may predominantly display topics such as "outbreak" and "patient zero" in its early stages, "lockdowns" and "vaccine development" in later stages, and "vaccination rates" and "boosters" in the present day.

To analyze the topics over time, one could in principle just partition the document collection into adequate "time slices" (e.g., months or years), and apply a conventional topic model separately over each time slice. However, this would fail to capture the continuity and the smooth transitions of the topics over time. For this reason, Lafferty and Blei in [81] have proposed a *dynamic topic model* (DTM) which is able to extract the topics from each time slice while taking into account the topics'

continuity and temporal dynamics. Motivated by the representational power of word embeddings in NLP, Diang et al. in [23] have recently proposed a *dynamic embedded topic model* (DETM) which integrates DTM with embedded word representations. Since word embeddings can be pre-trained in a completely unsupervised way over large amounts of text, an embedded model such as DETM can take advantage of the information captured by the word embeddings' pre-training.

However, a common limitation for all these topic models is that they cannot be easily tuned to explore improvements of the performance evaluation measures. For this reason, in this chapter we propose adding a tunable parameter (a "temperature") to the word distributions of DETM to attempt increasing the model's performance. We have tested the proposed model, aptly nicknamed *DETM-tau*, over three diverse and probing datasets: a time-sliced subset of the COVID-19 Open Research Dataset (CORD-19) [17], the United Nations (UN) General Debate Corpus [47], and the ACL Title and Abstract Dataset [48], comparing it with the best dynamic topic models from the literature such as DTM and DETM. The experimental results show that the proposed model has been able to achieve higher topic coherence and also lower test-set perplexity than both DTM and DETM in all cases.

The rest of this chapter is organized as follows: the related work is presented in Section 7.2, including a concise review of the key topic models. DETM is recapped in Section 7.3.1, while the proposed approach is presented in Section 7.3.2. The experiments and their results are presented in Section 7.4. Eventually, the conclusion is given in Section 7.5.

7.2 Related Work

In this section, we review the topic models that are closely related to the proposed work, such as latent Dirichlet allocation (LDA), dynamic topic models, and topic models based on word and topic embeddings.

Let us consider a document collection, D, with an overall vocabulary containing V distinct words. In LDA, the generic n-th word in the d-th document can be

noted as $w_{d,n}$, and simply treated as a categorical variable taking values in index set $[1 \dots V]$. One of the key assumptions of LDA is that each such word is uniquely assigned to a corresponding *topic*, $z_{d,n}$, which is another categorical variable taking values in set $[1 \dots K]$, where K is the number of topics that we choose to extract from the collection. In turn, each topic has an associated probability distribution over the words in the vocabulary, $\beta_k, k = 1 \dots K$, which accounts for the word frequencies typical of that specific topic. The full model of LDA can be precisely formulated and understood in terms of the following *generative model*, which is a model able to generate "synthetic" documents by orderly sampling from all the relevant distributions:

- For the d-th document, draw a K-dimensional vector, θ_d, with its topic proportions:

  $$\theta_d \sim \mathrm{Dir}(\theta_d|\alpha)$$

- For each word in the d-th document:

  Draw its topic: $z_{d,n} \sim \mathrm{Cat}(\theta_d)$

  Draw the word from the topic's word distribution:

  $$w_{d,n} \sim \mathrm{Cat}(\beta_{z_{d,n}})$$

In the above model, the first step for each document is to sample its topic proportions, θ_d, from a suitable Dirichlet distribution, $\mathrm{Dir}(\theta_d|\alpha)$. Once the topic proportions are given, the next step is to sample all of the document's words, by first sampling a topic, $z_{d,n}$, from categorical distribution [1] $\mathrm{Cat}(\theta_d)$, and then sampling the corresponding word, $w_{d,n}$, from the word distribution indexed by $z_{d,n}$, $\mathrm{Cat}(\beta_{z_{d,n}})$.

Overall, LDA is a computationally-efficient model that can be used to accurately extract the topics of a given training set of documents, and simultaneously identify the topic proportions of each individual document. LDA can also be applied to a

[1] Otherwise known as the multinomial distribution. The recent literature on variational inference seems to prefer the "categorical distribution" diction.

given *test set*; in this case, the parameters of the Dirichlet distribution, α, and the word distributions, β, are kept unchanged, and only the topic proportions for the given test documents are inferred. LDA has also spawned a large number of extensions and variants, including hierarchical versions[13], [33], sequential versions[26], class-supervised versions [26], sparse versions [28]–[30], and many others. However, the extensions that are closely relevant to our work are the dynamic topic model (DTM) [81], the embedded topic model (ETM) [76], and the dynamic embedded topic model (DETM) [23]. We briefly review DTM and ETM hereafter, while we recap DETM in greater detail in Section 7.3.

DTM is a topic model that captures the evolution of the topics in a corpus of documents that is sequentially organized (typically, along the time dimension). The corpus is first divided up into "time slices" (i.e., all the documents sharing the same time slot), and then the topics are extracted from each slice taking into account a dynamic assumption. For reasons of inference efficiency, DTM uses a logistic normal distribution, $\mathcal{LN}(\theta|\alpha)$, instead of a Dirichlet distribution to model the topic proportions of the individual documents. In addition, the samples of the logistic normal distribution are obtained by explicitly sampling a Gaussian distribution of equivalent parameters, and then applying the softmax operator, $\sigma(\cdot)$, to the Gaussian samples. The sequential dependencies between the time slices are captured by a simple dynamical model:

$$\begin{aligned} \alpha^t &\sim \mathcal{N}(\alpha^{t-1}, \delta^2 I) \\ \beta^t &\sim \mathcal{N}(\beta^{t-1}, \sigma^2 I) \end{aligned} \tag{7.1}$$

where α^t are the parameters of the logistic normal distribution over the topics at time t, and β^t is the matrix of all the word distributions (in logit scale), also at time t. The rest of the generative model for slice t can be expressed as:

- For the d-th document, draw its topic proportions (logit scale):

$$\theta_d \sim \mathcal{N}(\alpha^t, a^2 I).$$

- For each word in the d-th document:

  Draw its topic: $z_{d,n} \sim \text{Cat}(\sigma(\theta_d))$

  Draw the word from the topic's word distribution:

  $w_{d,n} \sim \text{Cat}(\sigma(\beta_{z_{d,n}}))$

DTM has proved capable of good empirical performance, and its inference is provided by efficient variational methods [81]. However, both LDA and DTM might lead to poor modelling in the presence of very large vocabularies, especially if the corpus is not sufficiently large to allow accurate estimation of the word probabilities. A possible mollification consists of substantially pruning the vocabulary, typically by excluding the most common and least common words. However, this carries the risk of excluding important terms a priori. The embedded topic model (ETM) [76] aims to overcome the limitations of categorical word distributions such as those of LDA and DTM by leveraging *word embeddings* [82], [83].

In ETM, each distinct word in the vocabulary is represented as a point in a standard word embedding space (typically, 300-1024D). Each topic, too, is represented as a point (a sort of "average") in the same embedding space. The compatibility between a word and a topic is then simply assessed by their dot product, and the probability of the word given the topic is expressed as in a common logistic regression classifier. The full generative model of ETM can be given as:

- For the d-th document, draw its topic proportions (logit scale):

  $\theta_d \sim \mathcal{N}(0, I)$

- For each word in the d-th document:

  Draw its topic: $z_{d,n} \sim \text{Cat}(\sigma(\theta_d))$

  Draw the word from the topic's word distribution:

$$w_{d,n} \sim \text{Cat}(\sigma(\rho^\top \eta_{z_{d,n}}))$$

In the above, we have noted as ρ the word embedding matrix, which contains the embeddings of all the words in the given vocabulary. Assuming a dimensionality of L for the embedding space, ρ's size is $L \times V$. In turn, with notation η_k we have noted the embedding of the k-th topic. Therefore, the dot product $\rho^\top \eta_k$ evaluates to a V-dimensional vector which, suitably normalised by the softmax, returns the probabilities for the word distribution of topic k.

The ETM is a powerful topic model that joins the advantages of LDA with the well-established word embeddings. The main benefit brought by the word embeddings is that they can be robustly pre-trained using large amounts of unsupervised text from a relevant domain (potentially, even the collection itself). During training of the ETM, a user can choose to either 1) use the pre-trained word embeddings, keeping them fixed, or 2) load them as initial values, but update them during training. In alternative, a user can also choose to update the word embeddings during training, but initialise them from arbitrary or random values (in this case, not taking advantage of pre-training). Dieng *et al.* in [76] have shown that the ETM has been able to achieve higher topic coherence and diversity than LDA and other contemporary models. While the ETM, like LDA, is limited to the analysis of static corpora, it can also be extended to incorporate dynamic assumptions. This is the aim of the dynamic embedded topic model (DETM) that we describe in the following section.

7.3 Methodology

In this section, we first describe our baseline, the dynamic embedded topic model (7.3.1), and then we present the proposed approach (7.3.2).

7.3.1 The dynamic embedded topic model

The dynamic embedded topic model (DETM) joins the benefits of DTM and ETM, allowing the model to capture the topics' evolution over time while leveraging the representational power of word embeddings. The dynamic assumption over the topic

proportions is the same as for the DTM:

$$\alpha^t \sim \mathcal{N}(\alpha^{t-1}, \delta^2 I) \tag{7.2}$$

but a dynamic prior is now assumed over the topic embeddings:

$$\eta^t \sim \mathcal{N}(\eta^{t-1}, \gamma^2 I) \tag{7.3}$$

The rest of the generative model for slice t is:

- For the d-th document, draw its topic proportions (logit scale):

  $\theta_d \sim \mathcal{N}(\alpha^t, a^2 I)$.

- For each word in the d-th document:

  Draw its topic: $z_{d,n} \sim \text{Cat}(\sigma(\theta_d))$

  Draw the word from the topic's word distribution:

  $w_{d,n} \sim \text{Cat}(\sigma(\rho^\top \eta^t_{z_{d,n}}))$

The training of DETM involves maximizing the posterior distribution over the model's latent variables, $p(\theta, \eta, \alpha|D)$. However, maximizing the exact posterior is intractable. Therefore, the common approach is to approximate it with variational inference [84] using a factorized distribution, $q_v(\theta, \eta, \alpha|D)$. Its parameters, noted collectively as v, are optimized by minimizing the Kullback-Leibler (KL) divergence between the approximation and the posterior, which is equivalent to maximizing the following expectation lower bound (ELBO):

$$\mathcal{L}(v) = \mathbb{E}[\log p(\theta, \eta, \alpha, D) - \log q_v(\theta, \eta, \alpha|D)] \tag{7.4}$$

The implementation of q_v relies on feed-forward neural networks to predict the variational parameters, and on LSTMs to capture the temporal dependencies; we

Table 7.1: Key sizes of the datasets used for the experiments.

Dataset	Training set	Validation set	Test set	Timestamps	Vocabulary
CORD-19TM	15,300	900	1,800	18	70,601
UNGDC	1,96,290	11,563	23,097	46	12,466
ACL	8,936	527	1,051	31	35,108

refer the reader to [23] for details.

7.3.2 The proposed approach: DETM-tau

The fundamental evaluation measure for a topic model is the *topic coherence* [43]. This measure looks at the "top" words in the word distribution of each topic, and counts how often they co-occur within each individual document. The assumption is that the higher the co-occurrence, the more "coherent" is the extracted topic model.

However, topic models cannot be trained to optimize the topic coherence. In the first place, the coherence is a counting measure that depends on the outcome of a ranking operation (a top-K argmax), and it is therefore not differentiable in the model's parameters. In the second place, it is evaluated globally over the entire document set. As a consequence, alternative approaches based on reinforcement learning [51] would prove excruciatingly slow, and would not be able to single out and reward the contribution of the individual documents (the so-called "credit assignment" problem [85]).

For this reason, in this work we attempt to improve the topic coherence by utilizing a softmax *with temperature* [86] in the word distributions. The inclusion of a temperature parameter can make the word distributions "sharper" (i.e. the probability mass more concentrated in the top words, for temperatures < 1) or smoother/more uniform (for temperatures > 1). We expect this to have an impact on the final word ranking, as high temperatures will make mixing more pronounced during training, while low temperatures may "freeze" the ranking to an extent. With the addition of the temperature parameter, τ, the word distributions take the form:

$$w \sim \mathrm{Cat}(\sigma(\rho^\top \eta_z / \tau)) \tag{7.5}$$

While parameter τ can be optimized with the training objective like all the other parameters, we prefer using a simple validation approach over a small, plausible range of values to select its optimal value.

7.4 Experiments and Results

7.4.1 Experimental set-up

For the experiments, we have used three popular document datasets: the COVID-19 Open Research Dataset (CORD-19) [17], the United Nation General Debate Corpus (UNGDC) [47] and the ACL Title and Abstract Dataset (ACL) [48]. CORD-19 is a resource about COVID-19 and related coronaviruses such as SARS and MERS, containing over 500,000 scholarly articles, of which 200,000 with full text. For our experiments, we have created a subset organized in monthly time slices between March 2020 and August 2021, limiting each slice to the first 1,000 documents in appearance order to limit the computational complexity. We refer to our subset as CORD-19TM, and we release it publicly for reproducibility of our experiments. UNGDC covers the corpus of texts of the UN General Debate statements from 1970 to 2015 annotated by country, session and year. For this dataset, we have considered yearly slices. The ACL dataset [48] includes 10,874 title and abstract pairs from the ACL Anthology Network which is a repository of computational linguistics and natural language processing articles. For this dataset, too, we have considered yearly slices, with the years spanning from 1973 to 2006 (NB: three years are missing). As in [23], the training, validation and test sets have been created by splitting the datasets into 85%, 5% and 10% splits, respectively. All the documents were preprocessed with tokenization, stemming and lemmatization, eliminating stop words and words with document frequency greater than 70% and less than 10%, as in [23].

As models, we have compared the proposed DETM-tau with: the original DETM, DTM, and LDA applied separately to each individual time slice. As performance metrics, we have used the *perplexity* and the *topic coherence* which are the de-facto

standards for this task. The perplexity is a measure derived from the probability assigned by the model to a document set, and should be as low as possible. It is typically measured over the test set to assess the model's generalization. The topic coherence is a measure of the co-occurrence of the "top" K words of each topic within single documents, and should be as high as possible. It is typically measured over the training set to assess the explanatory quality of the extracted topics. Several measures for the topic coherence have been proposed, and we use the NPMI coherence [43] with $K = 10$, as in [23]. As number of topics, we have chosen 20 and 40 which are commonly-used values in the literature [2]. For the selection of the temperature parameter, τ, we have used range $[0.25 - 2.25]$ in 0.5 steps. All other hyperparameters have been left as in the corresponding original models.

7.4.2 Results

Tables 7.2 and 7.3 show the results over the CORD-19TM dataset with 20 and 40 topics, respectively. In terms or perplexity, the proposed DETM-tau has neatly outperformed the original DETM for both 20 and 40 topics (NB: the perplexity is not available for the LDA and DTM models). In terms of topic coherence, DETM-tau has, again, achieved the highest values. The second-best results have been achieved in both cases by DTM, while DETM and LDA have reported much lower scores. In particular, the very poor performance of LDA shows that applying a standard topic model separately on each time slice is an unsatisfactory approach, and musters further support for the use of dynamic topic models for timestamped document analysis.

Tables 7.4 and 7.5 show the results over the UNGDC and ACL datasets, respectively. For these datasets, we have not carried out experiments with DTM as it proved impractically time-consuming, and we omitted LDA outright because of its non-competitive performance. On both these datasets, too, DETM-tau has been able to achieve both lower perplexity and higher coherence than the original DETM. We

[2]We also experienced computational issues with larger number of topics with the DTM models on some datasets, and we therefore capped the number to 40.

Table 7.2: Results on the CORD-19TM dataset with 20 topics

Model	LDA	DTM	DETM	DETM-tau
Perplexity	—	—	15548.8	**14379.2**
Coher. NPMI	-0.049	0.114	0.059	**0.129**

Table 7.3: Results on the CORD-19TM dataset with 40 topics

Model	LDA	DTM	DETM	DETM-tau
Perplexity	—	—	14966.3	**13129.7**
Coher. NPMI	-0.047	0.081	-0.043	**0.093**

Table 7.4: Results on the UNGDC dataset with 20 and 40 topics

Model	DETM	DETM-tau	DETM	DETM-tau
# topics	20		40	
Perplexity	3032.8	**3023.5**	2798.9	**2782.0**
Coher. NPMI	0.121	**0.129**	0.048	**0.124**

Table 7.5: Results on the ACL dataset with 20 and 40 topics

Model	DETM	DETM-tau	DETM	DETM-tau
# topics	20		40	
Perplexity	5536.4	**5421.1**	4360.0	**4169.6**
Coher. NPMI	0.150	**0.179**	0.153	**0.174**

believe that these results provide clear evidence of the importance of controlling the sharpness-smoothness trade-off of the word distributions.

To explore the sensitivity of the results to the temperature parameter, τ, Fig. 7.1 plots the values of the perplexity and the topic coherence of DETM-tau (CORD-19TM, 20 topics) for various values of τ, using DETM as the reference. It is clear that setting an appropriate value is important for the model's performance. However, the plots show that the proposed model has been able to outperform DETM for an ample range of values. In addition, Fig. 7.2 plots the values of the perplexity and the topic coherence at successive training epochs. The plots show that both metrics improve for both models as the training progresses. Given that the topic coherence is not an explicit training objective, its increase along the epochs is remarkable and gives evidence to the effective design of both models.

Eventually, we present a concise qualitative analysis of the extracted topics through

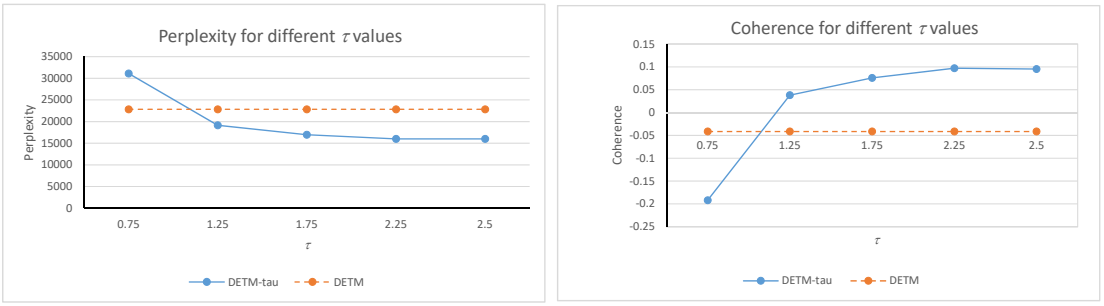

Figure 7.1: Perplexity and topic coherence for DETM-tau for various values of the temperature parameter, τ (CORD-19TM, 20 topics). The value for DETM is used for comparison.

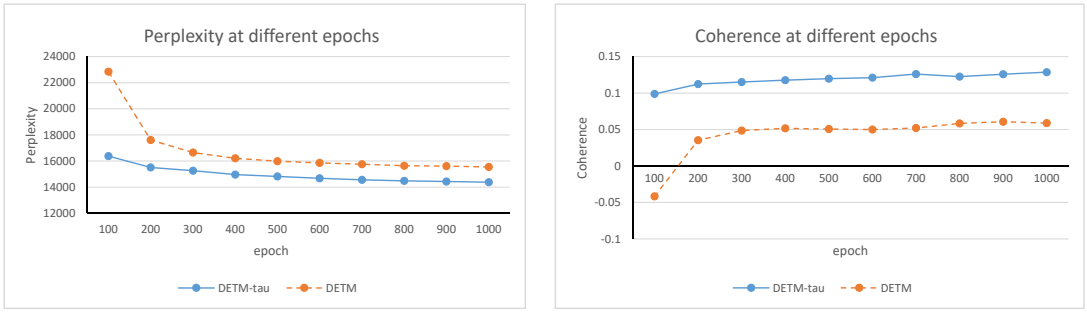

Figure 7.2: Perplexity and topic coherence for DETM and DETM-tau at successive training epochs (CORD-19TM, 20 topics).

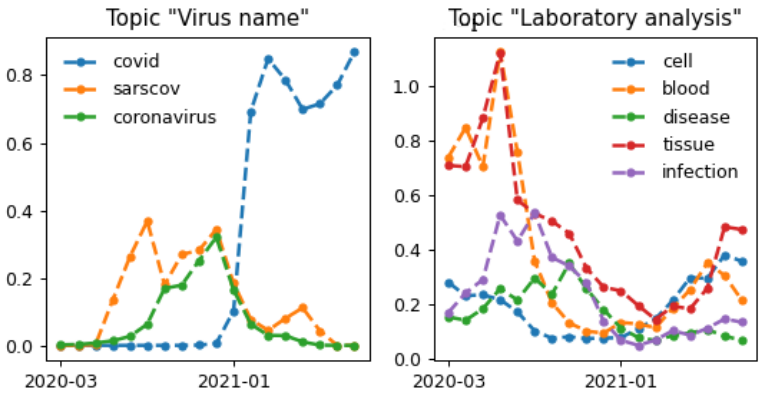

Figure 7.3: Evolution of the probability of a few, selected words within their topics for the DETM-tau model with the CORD-19TM dataset, 20 topics.

Table 7.6: Examples of topics extracted by DETM-tau from the CORD-19TM dataset (20 topics) at different time slices.

Time slice	Examples of topics
0	zikv cytokine proinflammatory resuscitation ferritin antitumor exosomes thoracic evidencebased patienten cells infection cell virus blood disease protein tissue infected receptor patients patient health clinical care hospital months disease years therapy
10	exosomes copd frailty mgml tavi absorbance biofilm sigmaaldrich evidencebased virulence social education research health people services industry culture educational providers macrophages antibacterial antioxidant kshv mmp lmics propolis sdgs inactivation hydrogel patients studies health care patient clinical treatment disease population risk
17	nanoparticles proinflammatory bioactive antifungal inhospital coagulation angiogenesis inflammasome cells cell blood disease tissue cancer infection protein proteins metabolism patients health patient social education hospital clinical people care population

Table 7.6 and Fig. 7.3. Table 7.6 shows a few examples of the topics extracted by DETM-tau from the CORD-19TM dataset (20 topics) at time slices 0, 10 and 17. Each topic is represented by its ten most frequent words. Overall, all the examples seem to enjoy good coherence and descriptive power. For instance, the first topic at time slice 0 could be titled "immune response analysis" or something akin; the last topic at time slice 17 could be titled "population health"; and so forth. Therefore, the automated categorization of the articles into such topics seem to provide a useful, and completely unsupervised, analysis. In turn, Fig. 7.3 shows the temporal evolution of the frequency of a few, manually selected words within their respective topics. The left-most topic, which we have labelled as "virus name", shows that referring to COVID-19 by the names "coronavirus" and "sarscov" was popular during 2020; conversely, as of January 2021, the name "covid" has become dominant. The right-most topic shows that words such as "blood", "infection" and "tissue" have decreased their in-topic frequency over time, possibly in correspondence with an increased understanding of the disease. These are just examples of the insights that can be obtained by dynamic topic models.

7.5 Conclusion

This chapter has presented a temperature-modified dynamic embedded topic model for topic modelling of timestamped document collections. The proposed model uses a softmax with temperature over the word distributions to control their sharpness/smoothness trade-off and attempt to achieve a more effective parameterization of

the overall topic model. Experiments carried out over three timestamped datasets (a subset of the CORD-19 dataset referred to as CORD-19TM, the United Nation General Debate Corpus (UNGDC) and the ACL Title and Abstract Dataset (ACL)) have showed that the proposed model, suitably nicknamed DETM-tau, has been able to outperform the original DETM model by significant margins in terms of both perplexity and topic coherence. In addition, DETM-tau has performed remarkably above the other compared models. A qualitative analysis of the results has showed that the proposed model has generally led to interpretable topics, and can offer insights into the evolution of the topics over time.

Chapter 8

Conclusions and Future Work

To restate the motivations for our work, let us once more acknowledge a key, standing limitation of topic models: the main metric used for their performance evaluation – the topic coherence – is a countable metric and, as such, is not optimizable during training (similarly to the accuracy in supervised classification, or the BLEU score in summarization, and so forth). The surrogate measure used for training the models – the document likelihood (or, in the case of VAEs, the ELBO) – only rewards the fitting of the individual documents to the model and is very different from the topic coherence. In the absence of correctives, there is an inherent risk that training will overfit the training objective and perform poorly on the evaluation measure.

On the other hand, *reinforcement learning* has proved a very powerful framework to train models to perform better. Its general appeal lies in its ability to leverage both differentiable and non-differentiable "rewards" to guide the training of the models, jointly with its use of sampling to increase the exploration of the parameter space. Despite the many existing topic models, until this present work the field had undeniably made limited use of reinforcement learning. In our case, we have set our focus on the probability of the topic proportions of the individual documents as our *policy*, and designed different rewards for it. Unfortunately, the topic coherence is unsuitable even as a non-differentiable reward simply because it is way too computationally-heavy to be evaluated repeatedly during training. Therefore,

we have designed our rewards around the ELBO itself, including baselines, sampling, "temperatures", and ensuring that the gradient estimators be unbiased and low-variance (both highly desirable properties). By adding the reinforcement learning objective to the conventional training objective, we have obtained a form of "regularization" that has led to sizable improvements of the topic coherence in all cases.

The first, specific contribution of this thesis has been the use of the REINFORCE algorithm over the topic proportions generated by a state-of-the-art variational autoencoder topic model. In the conventional configuration, the training of this model only minimizes the training loss, while in the proposed configuration it also directly rewards the generation of suitable topic proportions (the "policy"). This has proved able to achieve remarkable experimental results. In the next two units of work, we have introduced the assumption that each individual document would be generated by a single, "main" topic. While this assumption can seem restrictive, we have immediately relaxed it in two different ways. The first has been the use of the Gumbel-Softmax distribution in place of the categorical distribution to 1) diversify the topic vectors by sampling and 2) control their sparsity by the Gumbel-Softmax' temperature parameter. This has worked well, with valuable experimental results. However, the Gumbel-Softmax introduces a bias in the gradient estimator compared to the exact gradient. For this reason, as a second approach we have experimented with the REBAR gradient estimator, which has a number of attractive properties, including being unbiased with respect to the exact gradient and enjoying a low-variance design. This modification has also led to marked performance improvements. However, as shown in Section 5.5.1, the Gumbel-Softmax approach has performed better, suggesting that suitably adjusting the temperature parameter within REBAR may be a way to achieve the best of both approaches.

Topic models normally convert the individual documents into a bag-of-words (BoW) representation, which is simply a histogram of the frequencies of the vocabulary words in the document. While informative, this representation neglects both the order and the context in which the document's words appear and for this reason

some recent work has proposed augmenting it with Transformer-based embeddings [46]. However, no work to date had used these representations as training objective, and for this reason we have proposed a novel training objective that minimizes the distance between a pre-computed embedding for the document and an embedding predicted by the topic model. To this aim, we have also compared the effectiveness of different distance functions such as the Euclidean distance, the Manhattan distance, the Minkowski distance of order three and the cosine distance.

Conventional topic models fail to capture the continuity and the smooth transitions of the topics over time. For this reason, Lafferty and Blei have proposed a dynamic topic model (DTM) which is able to extract the topics from each time slice while taking into account the topics' continuity and temporal dynamics, and Diang et al. have recently proposed a dynamic embedded topic model (DETM) which integrates DTM with embedded word representations. However, a limitation of these topic models is that they cannot be easily tuned to explore improvements of the performance evaluation measures. For this reason, we have proposed adding a tunable parameter (a "temperature") to the word distributions of DETM to attempt increasing the model's performance. The experimental results have again been very encouraging.

While we are satisfied with these contributions, we believe that there is still ample room for future work. Within it, we believe that it would be worth exploring other, more flexible gradient estimators. An example is RELAX [69] which enjoys some principled advantages over REBAR: while REBAR uses the loss function itself as the baseline, RELAX can use any arbitrary, trained neural network as the baseline. This adds flexibility to the baseline and the training objective overall, and could potentially lead to performance improvements.

Another interesting direction to explore could be the integration of the various reinforcement learning objectives used in this thesis (Gumbel-Softmax, REBAR, REINFORCE), and possibly others, with both the contextualized representations proposed in Chapter 6 (both as input and as training objective) and the dynamic topic model proposed in Chapter 7. While the computational complexity would have to

be carefully monitored, it should be feasible now or in the near future thanks to the constant increase of GPUs' memory and speed.

A last, promising area to explore could be that of *flow-based* deep generative models which overcome some of the standing limitations of variational autoencoders by flow normalization, a powerful statistics tool for density estimation. Normalizing flow transforms a simple distribution into a complex distribution by applying a sequence of invertible transformation functions, where the variable is substituted repeatedly with a new one according to the change of variables theorem to eventually obtain the probability distribution of the target variable. In this way, normalizing flow somehow combines the best of both worlds, allowing both deep feature learning and tractable marginal likelihood estimation.

Bibliography

[1] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[2] C. Arnold, S. El-Saden, A. Bui, and R. Taira, "Clinical case-based retrieval using latent topic analysis," *AMIA Annual Symposium Proceedings*, vol. 2010, pp. 26–30, 2010.

[3] E. Sarioglu, H.-A. Choi, and K. Yadav, "Clinical report classification using natural language processing and topic modeling," in *The 11th International Conference on Machine Learning and Applications*, vol. 2, 2012, pp. 204–209.

[4] R. Zhang, S. Pakhomov, S. Gladding, M. Aylward, E. Borman-Shoap, and G. Melton, "Automated assessment of medical training evaluation text," *AMIA Annual Symposium Proceedings*, vol. 2012, pp. 1459–68, 2012.

[5] M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak, "Making words work: Using financial text as a predictor of financial events," *Decis. Support Syst.*, vol. 50, no. 1, pp. 164–175, 2010.

[6] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction," in *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNL 2015)*, 2015, pp. 1354–1364.

[7] D. Devyatkin, E. Nechaeva, R. Suvorov, and I. Tikhomirov, "Mapping the research landscape of agricultural sciences," *Foresight and STI Governance*, vol. 12, no. 1, pp. 57–76, 2018.

[8] D. Alvarez-Melis and M. Saveski, "Topic modeling in Twitter: Aggregating tweets by conversations," in *The 10th International Conference on Web and Social Media*, 2016, pp. 519–522.

[9] G. Xu, Y. Meng, Z. Chen, X. Qiu, C. Wang, and H. Yao, "Research on topic detection and tracking for online news texts," *IEEE Access*, vol. 7, pp. 58 407–58 418, 2019.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[11] T. Hofmann, "Probabilistic latent semantic analysis," in *Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, 1999, pp. 289–296.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[13] T. L. and Nevin Lianwen Zhang and P. Chen, "Hierarchical latent tree analysis for topic detection," *CoRR*, Lecture Notes in Computer Science, vol. 8725, pp. 256–272, 2014.

[14] A. Srivastava and C. A. Sutton, "Autoencoding variational inference for topic models," in *The 5th International Conference on Learning Representations, (ICLR 2017)*, 2017, pp. 1–12.

[15] D. P. K. and Max Welling, "Auto-encoding variational Bayes," in *The 2nd International Conference on Learning Representations (ICLR 2014)*, 2014, pp. 1–14.

[16] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *The 34th International Conference on Machine Learning (ICML 2017)*, 2017, pp. 2410–2419.

[17] L. L. Wang, K. Lo, Y. Chandrasekhar, *et al.*, "CORD-19: The COVID-19 Open Research Dataset," in *1st Workshop on NLP for COVID-19 at ACL 2020*, vol. 1, 2020, pp. 1–12.

[18] J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *22nd International Conference on World Wide Web*, 2013, pp. 897–908.

[19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, 1992.

[20] C. Maddison, A. Mnih, and Y. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *CoRR*, vol. abs/1611.00712, 2016.

[21] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *The 5th International Conference on Learning Representations (ICLR 2017)*, 2017, pp. 1–12.

[22] G. Tucker, A. Mnih, C. J. Maddison, D. Lawson, and J. Sohl-Dickstein, "RE-BAR: low-variance, unbiased gradient estimates for discrete latent variable models," pp. 2627–2636, 2017.

[23] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "The dynamic embedded topic model," 2019. arXiv: `1907.05545 [cs.CL]`.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[25] "Tf–idf," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Springer US, 2010, pp. 986–987.

[26] F. Rodrigues, M. Lourenco, B. Ribeiro, and F. C. Pereira, "Learning supervised topic models for classification and regression from crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2409–2422, 2017, ISSN: 2160-9292.

[27] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: topic modeling over short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, 2014.

[28] J. Zhu and E. P. Xing, "Sparse topical coding," in *The 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, 2011, pp. 831–838.

[29] A. Zhang, J. Zhu, and B. Zhang, "Sparse online topic models," in *The 22nd International World Wide Web Conference (WWW 2013)*, 2013, pp. 1489–1500.

[30] M. Peng, Q. Xie, Y. Zhang, *et al.*, "Neural sparse topical coding," in *The 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, 2018, pp. 2332–2340.

[31] Y. Zuo, C. Li, H. Lin, and J. Wu, "Topic modeling of short texts: A pseudo-document view with word embedding enhancement," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021.

[32] L. Du, W. Buntine, H. Jin, and C. Chen, "Sequential latent Dirichlet allocation," *Knowledge and Information Systems*, vol. 31, pp. 475–503, 2012.

[33] H. Kim, B. Drake, A. Endert, and H. Park, "Architext: Interactive hierarchical topic modeling," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 9, pp. 3644–3655, 2021.

[34] W. Li, J. Yin, and H. Chen, "Supervised topic modeling using hierarchical Dirichlet process-based inverse regression: Experiments on e-commerce applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1192–1205, 2018.

[35] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Mach. Learn.*, vol. 50, no. 1-2, pp. 5–43, 2003.

[36] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *CoRR*, vol. abs/1601.00670, 2016.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[38] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[39] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. The MIT Press, 1994, ISBN: 0262150417.

[40] J. Glover, "Modeling documents with Generative Adversarial Networks," in *NIPS 2016 Workshop on Adversarial Training*, 2016, pp. 1–7.

[41] C. Wang, J. W. Paisley, and D. M. Blei, "Online variational inference for the hierarchical dirichlet process," in *The Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 15, JMLR.org, 2011, pp. 752–760.

[42] W. Kou, F. Li, and T. Baldwin, "Automatic labelling of topic models using word vectors and letter trigram vectors," in *Information Retrieval Technology*

- *11th Asia Information Retrieval Societies Conference, AIRS 2015*, vol. 9460, 2015, pp. 253–264.

[43] J. H. Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, 2014, pp. 530–539.

[44] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *The Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*, 2015, pp. 399–408.

[45] K. Lang, "Newsweeder: Learning to filter netnews," in *The 12th International Conference on Machine Learning (ICML 1995)*, 1995, pp. 331–339.

[46] F. Bianchi, S. Terragni, and D. Hovy, "Pre-training is a hot topic: Contextualized document embeddings improve topic coherence," in *The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)"*, 2021. DOI: 10.18653/v1/2021.acl-short.96". [Online]. Available: https://aclanthology.org/2021.acl-short.96%22.

[47] M. S. Baturo A Dasandi N, "Understanding state preferences with text as data:" in *Introducing the UN General Debate corpus. Research Politics.*, 2017. DOI: 10.1177/2053168017712821.

[48] S. Bird, R. Dale, B. Dorr, *et al.*, "The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics," in *International Conference on Language Resources and Evaluation*, 2008, pp. 1755–1759.

[49] Y. Zhang, B. Xu, and T. Zhao, "Convolutional multi-head self-attention on memory for aspect sentiment classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, pp. 1038–1044, 2020.

[50] A. Grover, M. Dhar, and S. Ermon, "Flow-gan: Combining maximum likelihood and adversarial learning in generative models," in *The Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, AAAI Press, 2018, pp. 3069–3076.

[51] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. 2018.

[52] L. Gui, J. Leng, G. Pergola, Y. Zhou, R. Xu, and Y. He, "Neural topic model with reinforcement learning," in *The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference onNatural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Association for Computational Linguistics, 2019, pp. 3476–3481.

[53] P. Hennig, D. H. Stern, R. Herbrich, and T. Graepel, "Kernel topic models," in *The Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012*, ser. JMLR Proceedings, vol. 22, 2012, pp. 511–519.

[54] D. J. Hand, "Text mining: Classification, clustering, and applications," *International Statistical Review*, vol. 78, pp. 134–135, 2010.

[55] A. Murakami, P. Thompson, S. Hunston, and D. Vajn, "What is this corpus about?: Using topic modelling to explore a specialised corpus," *Corpora*, vol. 12, no. 2, pp. 243–277, 2017.

[56] X. Kang, F. Ren, and Y. Wu, "Exploring latent semantic information for textual emotion recognition in blog articles," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, pp. 204–216, 2018.

[57] H. Zhang, B. Chen, Y. Cong, D. Guo, H. Liu, and M. Zhou, "Deep autoencoding topic model with scalable hybrid Bayesian inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–22, 2020.

[58] Q. Lian, W. Yan, X. Zhang, and S. Chen, "Single image rain removal using image decomposition and a dense network," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, pp. 1428–1437, 2019.

[59] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, pp. 441–451, 2019.

[60] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.

[61] R. Das, M. Zaheer, and C. Dyer, "Gaussian LDA for topic models with word embeddings," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing)*, 2015, pp. 795–804.

[62] S. Seifollahi, M. Piccardi, and A. Jolfaei, "An embedding-based topic model for document classification," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, no. 3, pp. 1–13, 2021.

[63] A. Schofield, M. Magnusson, L. Thompson, and D. Mimno, "Understanding text pre-processing for latent Dirichlet allocation," in *First Women and Underrepresented Minorities in NLP Workshop*, 2017, pp. 1–4.

[64] L. Ruthotto and E. Haber, *An introduction to deep generative modeling, arXiv:2103.05180*, 2021.

[65] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems 12 (NIPS)*, 1999, pp. 1008–1014.

[66] A. Kumar, N. Esmaili, and M. Piccardi, "Topic-document inference with the Gumbel-Softmax distribution," *IEEE Access*, vol. 9, pp. 1313–1320, 2021. DOI: 10.1109/ACCESS.2020.3046607.

[67] E. J. Gumbel, "Statistical theory of extreme values and some practical applications: A series of lectures," *Number 33. US Govt. Print. Office*, pp. 1–60, 1954.

[68] C. J. Maddison, D. Tarlow, and T. Minka, "A* sampling," in *Advances in Neural Information Processing Systems*, 2014, pp. 3086–3094.

[69] W. G. and Dami Choi, Y. Wu, G. Roeder, and D. Duvenaud, "Backpropagation through the void: Optimizing control variates for black-box gradient estimation," in *The 6th International Conference on Learning Representations, (ICLR 2018)*, 2018.

[70] M. Abadi, A. Agarwal, P. Barham, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf.

[71] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in pytorch," in *NIPS 2017 Workshop on Autodiff*, 2017.

[72] J. Bradbury, R. Frostig, P. Hawkins, *et al.*, *JAX: Composable transformations of Python+NumPy programs*, version 0.1.46, 2018. [Online]. Available: `http://github.com/google/jax`.

[73] A. Kumar, N. Esmaili, and M. Piccardi, "Neural topic model training with the REBAR gradient estimator," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 2022, ISSN: 2375-4699.

[74] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)"*, 2019. DOI: `"10.18653/v1/N19-1423"`. [Online]. Available: `https://aclanthology.org/N19-1423`.

[75] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, 2018.

[76] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.

[77] F. Torregrossa, R. Allesiardo, V. Claveau, N. Kooli, and G. Gravier, "A survey on training and evaluation of word embeddings," *Int. J. Data Sci. Anal.*, vol. 11, no. 2, pp. 85–103, 2021.

[78] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 3982–3992. DOI: `10.48550/ARXIV.1908.10084`. [Online]. Available: `https://arxiv.org/abs/1908.10084`.

[79] N. A. Heckert and J. J. Filliben, *NIST Handbook 148: Dataplot Reference Manual, Volume 2: Let Subcommands and Library Functions*. National Institute of Standards and Technology Handbook Series, 2003.

[80] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[81] J. D. Lafferty and D. M. Blei, "The dynamic topic model," in *The 23rd International Conference on Machine learning*, 2006, pp. 113–120.

[82] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[83] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *The 31th International Conference on Machine Learning*, vol. 32, 2014, pp. 1188–1196.

[84] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," 2017, pp. 859–877.

[85] M. Minsky, "Steps toward artificial intelligence," *Proc. IRE*, vol. 49, no. 1, pp. 8–30, 1961.

[86] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, MIT Press, 1999, pp. 61–74.