SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of
Computer Science, Systems and Communication

PhD program in Computer Science — Cycle XXXIV

Modeling Relational and Contextual Information into Topic Models and their Evaluation

Terragni Silvia

764513

Tutor: Prof. Giuseppe Vizzari

Supervisor: Prof. Elisabetta Fersini

Co-supervisor: Prof. Enza Messina

Coordinator: Prof. Leonardo Mariani

ACADEMIC YEAR 2020/2021

Al piccolo Giovanni

An expert is a person who has made all the mistakes
that can be made in a very narrow field.

— Niels Bohr

ABSTRACT

Textual knowledge is one of the main pillars of our society. Indeed, human knowledge is often passed along using words. Since the invention of writing, humans have narrated and described their existence with words over pieces of papers. This amount of knowledge builds up to what the entire civilization has collected over more than 5'000 years. Historians and social and political scientists look for ways to understand better this vast amount of collective knowledge that cannot be manually explored.

To this end, researchers from machine learning, statistics and computational linguistic have developed topic models, a suite of algorithms that aim to annotate large archives of documents with thematic information. The popularity of these models is due to the fact that they are unsupervised and that they are interpretable. Topic models analyze and summarize the main themes, or topics, of large collections of documents, presenting the information in a compact and understandable form.

Most topic models focus only on the words encoded in the documents. However, additional information can be introduced into topic models to improve their performance. In fact, in many real-world cases, we seldom have only the mere texts to analyze. Instead, we have additional information or metadata related to the documents, e.g., the document's author, the date, hyperlinks to other documents, a set of hashtags, mentions or labels. We can use this prior information to help a topic model discover better topics. For example, knowing that a document cites another document increases our confidence that the documents talk about the same topics. Also, topic models often ignore word order and contextual information, making it difficult to infer high-quality topics.

Another problem in the field is related to the hyperparameters used to train the topics models. These hyperparameters control the training process and may have a significant impact on the performance and results of the models. However, researchers usually fix them, thus preventing us from discovering the best topic model on a given dataset.

In this thesis, we aim to tackle the mentioned problems. We introduce novel families of topic models to obtain better performance. We also explore the issues related to hyperparameter optimization by designing and developing a novel tool to supply researchers with better guidelines on how to train a topic model.

PUBLICATIONS

Publications related to this research work:

- **Terragni, S.**, Fersini, E., & Messina, E. (2020). *Constrained relational topic models.* Information Sciences, 512, 581-594.
- **Terragni, S.**, Nozza, D., Fersini, E., & Messina, E. (2020). *Which Matters Most? Comparing the Impact of Concept and Document Relationships in Topic Models.* Insights @ EMNLP 2020 (pp. 32-40).
- Bianchi, F., **Terragni, S.**, & Hovy, D. (2021). *Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence.* ACL 2021 (pp. 759-766).
- Bianchi, F., **Terragni, S.**, Hovy, D., Nozza, D., & Fersini, E. (2021). *Cross-lingual Contextualized Topic Models with Zero-shot Learning*. EACL 2021 (pp. 1676-1683).
- **Terragni, S.**, Fersini, E., Galuzzi, B. G., Tropeano, P., & Candelieri, A. (2021). *OCTIS: Comparing and Optimizing Topic models is Simple!*. EACL 2021: System Demonstrations (pp. 263-270).
- **Terragni, S.**, Fersini, E., & Messina, E. (2021). *Word Embedding-based Topic Similarity Measures.* NLDB 2021 (pp. 33-45), **Best Paper Award**.
- **Terragni, S.**, & Fersini, E. (2021). *An Empirical Analysis of Topic Models: Uncovering the Relationships between Hyperparameters, Document Length and Performance Measures.* RANLP 2021 (pp. 1408–1416).
- **Terragni, S.**, Candelieri, A., & Fersini, E. (2021). *The Role of Hyperparameters in Relational Topic Models: Prediction Capabilities vs Topic Quality.* (under review).
- **Terragni, S.**, Harrando, I., Lisena, P., Troncy, R., & Fersini, E. (2021). *Multi-Objective Bayesian Optimization for Hyperparameter Transfer in Topic Modeling.* (under review).
- **Terragni, S.**, & Fersini, E.(2022). *OCTIS 2.0: Optimizing and Comparing Topic Models in Italian is Even Simpler!* CLIC-IT 2021 (to appear).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

BACKGROUND

1 INTRODUCTION

1.1 TOPIC MODELING

Textual knowledge is one of the main pillars of our society. Indeed, human knowledge is often passed along using words. Since the invention of writing, humans have narrated and described their existence with words over pieces of papers. This amount of knowledge builds up to what the entire civilization has collected over more than 5'000 years. Historians and social and political scientists look for ways to understand better this vast amount of collective knowledge that cannot be manually explored.

To this end, researchers from machine learning, statistics and computational linguistic have developed **topic models, a suite of algorithms that aim to annotate large archives of documents with thematic information**. From the nineties, several topic models have been proposed across the years. They have been applied to various domains and tasks, becoming one of the major models in Natural Language Processing (NLP). The popularity of these models is due to their interpretability capabilities. Topic models analyze and summarize the main themes, or *topics*, of large collections of documents, presenting the information in a compact and understandable form.

But what is a topic? A topic can be defined as a subject that is discussed, written about, or studied. **In topic modeling, a topic is seen as a cluster of words that make sense together**. For example, the list of words *"learning, machine, deep, neural, network"* can be easily interpreted as a topic related to deep learning. The words *"probability, distribution, gaussian, variable, random"* are related to probability theory. Therefore, the objective of a topic model is to discover lists of representative keywords from a document collection.

The problem setting is simple. We have a collection of documents, and we want to figure out which are the main topics of the documents. The topic model takes as input the corpus of documents and the number of topics one wants to discover, and the model returns the topics, as the lists of keywords we have seen above. The set of discovered topics provides a general view of the corpus. However, a topic model can also specify the most significant topic (or topics) of each considered document. For example, the main topic of this thesis is "topic modeling", but there is also a little of probability theory, deep learning, and natural language processing. On the contrary, a topic that will not be treated in this thesis is reinforcement learning.

The strength of topic models is that they are unsupervised. They do not require any a priori annotations. The required elements are the corpus and the number of topics that we want to extract. This (apparent) simplicity has led many researchers and practitioners to use and apply topic models across the years over a wide range of applications, tasks and domains (Albalawi et al., 2020; Jelodar et al., 2018; Vayansky and Kumar, 2020).

1.2 RESEARCH CHALLENGES.

MODELING ADDITIONAL INFORMATION. All that glitters is not gold. Despite what we usually see in topic modeling papers, the discovered topics do not always make sense (Chang et al., 2009; Hu et al., 2014). A topic model can often discover "bad" topics. These bad topics can confuse two or more themes into one topic; two different topics can be duplicated, or some topics are "junk topics" and make no sense at all. Additional information can help reduce this issue, and thus a challenging direction is the **incorporation of information into topic models**.

In many real-world cases, we seldom have only the mere texts to analyze. Instead we may have additional information or metadata related to the documents, e.g., the document's author, the date, hyperlinks to other documents, a set of hashtags, mentions or labels. We can use this prior information to help a topic model discover better topics. For example, knowing that a document cites another document increases our confidence that the documents talk about the same topics. This information can sometimes be expressed in a relational form, originating a graph of documents or words: a document that cites another document, a web page that links to another web page, or a word may be a synonym of another one. Many approaches extend unsupervised topic models by incorporating this kind of relational information. However, **very few approaches investigate the impact of modeling multiple sources of relational information** (Yang et al., 2016a; Zhao et al., 2017).

Nevertheless, domain experts and domain-specific information about the documents are not always available resources, and a valuable solution in these cases consists in using *general* information. A topic modeling subfield already exploits publicly available resources about language, including knowledge graphs, taxonomies, and pre-trained word embeddings (Dieng et al., 2020; Li et al., 2017; Zhao et al., 2017). Meanwhile, pre-trained language models are becoming ubiquitous in Natural Language Processing, precisely for their ability to capture syntactic and semantic information of the words in a sentence. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), the most prominent architecture in this category, allows us to extract pre-trained word and sentence representations. Their use

has advanced state-of-the-art performance across many tasks. **Topic models could also benefit from the advantages that come from the use of contextualized representations**.

EVALUATION OF TOPIC MODELS. In parallel to the effort of incorporating information into topic models, we need to face another fundamental problem: **evaluating a topic model is not trivial**. Topic models can be evaluated in several ways: from downstream tasks, such as document classification and information retrieval (Boyd-Graber et al., 2017), to the analysis of the generated topics (Doogan and Buntine, 2021; Lau et al., 2014a). Most of the evaluations disregard the pre-processing of the texts, the evaluation metrics, and the hyperparameters of the models. These elements, especially the hyperparameters, have a relevant impact on the performance of the models.

Indeed, topic models are usually controlled by hyperparameters, whose values control the learning process. Fixing them prevents the researchers from discovering the best topic model on a given dataset. Yet, the evaluations of topic models are often limited to the comparison of models whose hyperparameters are held fixed (Doan and Hoang, 2021) or explored with ineffective techniques, e.g. grid search. **Finding the best hyperparameter configuration will therefore guarantee a fairer comparison between the models.**

However, discovering the best setting of hyperparameters may require high computational resources, especially without prior knowledge of the hyperparameters. To the best of our knowledge, there is no approach to transferring the knowledge acquired during the hyperparameter selection experiments. Taking inspiration from meta-learning techniques (Feurer et al., 2015; Jomaa et al., 2020), we can **investigate hyperparameters' transferability from a document corpus to an unseen one.**

1.3 CONTRIBUTIONS

The major contributions of this research are the definition of new methods for modeling additional information into topic models and the study of fair evaluation and comparison methods for topic models, which consider the multiple aspects that impact on the topic models' performance. In detail,

1. we define a method for introducing relational information related to words and documents into topic models, which is easy to implement, modular, and applicable to different topic models.

2. We propose a class of topic models that incorporate contextual information from state-of-the-art language models, allowing the

model to improve the topics' quality and perform zero-shot classification tasks.

3. We design and release to the NLP community a comprehensive framework for evaluating and fairly comparing topic models, which also allow us to investigate the different elements that play a role in evaluating the models.

4. We define an approach to transfer the knowledge we have acquired during the different models' evaluations to obtain topic models that guarantee optimal results efficiently.

1.4 ORGANIZATION OF THIS THESIS

In the following, we detail each Chapter of this thesis. The publications related to each Chapter are listed in chronological order.

CHAPTER 1: INTRODUCTION. In this Chapter, we have introduced the content of this thesis, outlining the main concepts and the most relevant content. The rest of the Chapter will also outline the main research questions that this research work aims to answer.

CHAPTER 2: PRELIMINARIES. In this Chapter, we explain the basic notions that are part of this research work. We give an overview of topic modeling, focusing on the well-known probabilistic topic model Latent Dirichlet Allocation. We also discuss the main types of word and document representations, their differences, and the hyperparameters' optimization methods in machine learning models.

CHAPTER 3: RELATED WORK. In this Chapter, we explore related approaches by analyzing two different areas: topic models and their evaluation. We will detail the topic modeling approaches incorporating additional information and describe the different methodologies and metrics for evaluating topic models. We will end this Chapter by discussing the methods for estimating the hyperparameters in topic models.

CHAPTER 4: MODELING RELATIONAL INFORMATION INTO CLASSICAL MODELS. In this Chapter, we outline the ideas behind incorporating additional information into topic models. We focus on one of the contributions of this research work, which is the incorporation of information into the form of relationships into classical probabilistic topic models.

The research questions answered by the following Chapter are:

Q4.1 How can we model additional document-level and word-level relational information into classical topic models?

Q4.2 Which is the impact of modeling document-level and word-level relational information into topic models?

Methods and results described within this Chapter are based on the following work, in which we show how to incorporate information into classical topic models.

- **Terragni, S.**, Fersini, E., & Messina, E. (2020). *Constrained relational topic models.* Information Sciences, 512, 581-594.
- **Terragni, S.**, Nozza, D., Fersini, E., & Messina, E. (2020). *Which Matters Most? Comparing the Impact of Concept and Document Relationships in Topic Models.* Insights @ EMNLP 2020 (pp. 32-40).

CHAPTER 5: MODELING CONTEXTUAL INFORMATION INTO NEURAL MODELS. This Chapter focuses on one of the contributions of this research work, which is the introduction of the class of Contextualized Topic Models. The models belonging to this class can improve the quality of the topics and use transfer learning to address zero-shot tasks. In this Chapter we aim to answer the following research questions:

Q5.1 How can we incorporate context information into neural topic models?

Q5.2 How can we exploit the cross-lingual capabilities of the multilingual pre-trained representations for topic modeling?

Methods and results described within this Chapter are based on the following work, in which we show how to incorporate pre-trained contextualized representations into neural topic models.

- Bianchi, F., **Terragni, S.**, & Hovy, D. (2021). *Pre-training is a hot topic: Contextualized document embeddings improve topic coherence.* ACL 2021 (pp. 759-766).
- Bianchi, F., **Terragni, S.**, Hovy, D., Nozza, D., & Fersini, E. (2021). *Cross-lingual contextualized topic models with zero-shot learning*. EACL 2021 (pp. 1676-1683)

CHAPTER 6: HYPERPARAMETER OPTIMIZATION FOR THE COMPARISON OF TOPIC MODELS This Chapter focuses on the definition of a general framework based on hyperparameter optimization for comparing topic models.

In this Chapter we aim to answer the following research questions:

Q6.1 Can we determine if a topic model can guarantee an optimal trade-off between different performance measures?

Q6.2 Can a performance measure imply a competing or correlated target for other performance measures?

Results and methods described within this Chapter are based on the following work, in which we propose our comparative framework for topic modeling and the results related to the use of the proposed framework:

- **Terragni, S.**, Fersini, E., Galuzzi, B. G., Tropeano, P., & Candelieri, A. (2021). *OCTIS: Comparing and Optimizing Topic models is Simple!*. EACL 2021: System Demonstrations (pp. 263-270).
- **Terragni, S.**, Fersini, E., & Messina, E. (2021). *Word Embedding-based Topic Similarity Measures.* NLDB 2021 (pp. 33-45), **Best Paper Award**.
- **Terragni, S.**, & Fersini, E.(2021). *An Empirical Analysis of Topic Models: Uncovering the Relationships between Hyperparameters, Document Length and Performance Measures.* RANLP 2021 (pp. 1408-1416).
- **Terragni, S.**, Calendieri, A., & Fersini, E.(2021). *The Role of Hyperparameters in Relational Topic Models: Prediction Capabilities vs Topic Quality.* (under review).

CHAPTER 7: BEYOND SINGLE-OBJECTIVE HYPERPARAMETER OPTIMIZATION This Chapter still focuses on comparing topic models and explores different research directions, originating from the previous Chapter.

In this Chapter we aim to answer the following research questions:

Q7.1: Can we optimize the hyperparameters of a topic model to guarantee an optimal trade-off between different performance measures using multi-objective optimization?

Q7.2: Can we transfer the best hyperparameter configurations from a dataset to an unseen dataset?

Results and methods described within this Chapter are based on the following work:

- **Terragni, S.**, Harrando, I., Lisena, P., Troncy, R., & Fersini, E.(2021). *Multi-Objective Bayesian Optimization for Hyperparameter Transfer in Topic Modeling.* (under review).
- **Terragni, S.**, & Fersini, E.(2021). *OCTIS 2.0: Optimizing and Comparing Topic Models in Italian is Even Simpler!* (to appear).

CHAPTER 8: CONCLUSIONS. Finally, we end this thesis by providing conclusions to this work by highlighting the most critical content. Eventually, we provide possible research directions that may start from the results provided within this thesis.

1.5 REPRODUCIBILITY

Recently, reproducibility has become a significant issue in AI and NLP-related fields (Bianchi and Hovy, 2021). Thus, the experiments run for this thesis can be replicated using codes and models freely available online. In the following, we summarize the links to the repositories for each chapter:

- Chapter 4:
  - Document-Constrained Relational Topic Models: https://github.com/MIND-LAB/Constrained-RTM
  - Entity-Constrained Relational Topic Models: https://github.com/MIND-LAB/EC-RTM
- Chapter 5:
  - Contextualized Topic Models: https://github.com/MilaNLProc/contextualized-topic-models
- Chapter 6 and 7:
  - Optimizing and Comparing Topic Models is Simple (OCTIS): https://github.com/MIND-LAB/octis
  - Comparative Analysis of Classical Topic Models: https://github.com/MIND-LAB/Constrained-RTM

2

PRELIMINARIES

In the following chapter, we introduce the foundations of this research work. This chapter should give the reader most of the required knowledge to access the rest of this work. We begin this chapter with an overview of topic modeling and word and document representations. Then, we provide an overview of probabilistic topic models, which is the focus of this thesis, and we eventually end the chapter with an overview of methods for selecting the hyperparameters of topic models.

2.1 TOPIC MODELING

Topic models are a class of models that provide an automatic way to analyze the main themes of large volumes of texts. A topic model describes a corpus of documents through a set of fixed topics, where each topic is represented by its most significant words. Figure 2.1 sketches how a topic model works, along with its input and outputs.

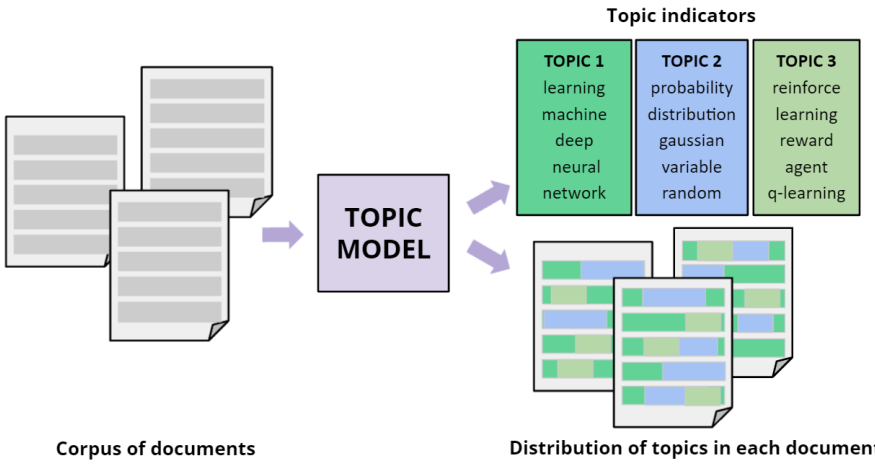


Figure 2.1: Overview of topic modeling. A corpus of documents is given as input and the model returns the list of topics and the topic representations of the documents.

A topic model presents the information in a compact and interpretable form. For example, as seen in the picture, a topic characterized by the words *"learning, machine, deep, neural, network"* can be easily interpreted as a topic related to deep learning, or the words *"probability, distribution, gaussian, variable, random"* are related to probability theory. However, a topic is not just an unordered list of key-

words: each word of the vocabulary has a specific weight, or probability weight, that identifies the importance of the word in the topic.

Not only does a topic model summarize a corpus by lists of coherent keywords, but each document can be described by the discovered topics in different proportions. Indeed, a document is rarely characterized by a single topic, rather it may talk about multiple topics. For example, an NLP paper can contain 30% of linguistics and 70% of computer science.

2.1.1 *Relationship between Topic Modeling and Clustering*

A topic model can be easily inserted into the categories of clustering algorithms (Bishop, 2006). In fact, a topic model allows us to divide the documents into a (usually, fixed) number clusters, i.e. the topics. Moreover, this process is usually done in an unsupervised way, without requiring any prior knowledge on the documents. We also expect that the documents of a cluster are coherent between each other but separated from the rest of documents, reflecting the notions of *cohesion* and *separation* of clustering theory respectively.

We have mentioned that a topic model describes a document as a mixture of different topics with different weights. This form of modeling is the so-called *soft clustering* (which is the opposite of *hard clustering*. In soft clustering, each document (or data point) can belong to more than one cluster.

Despite the similarities between topic modelling and clustering algorithms, topic models offer additional capabilities. A topic model not only provides a soft clustering of the documents, but it provides a way to making sense of the document corpus through the use of topics.

Finally, we will see later in Section 3.4 that the evaluation of topic models takes inspiration also from clustering evaluation. In fact, we expect the topics to be coherent (i.e. their words must be related to each other, similar to the concept of *cohesion* for clutering) and we also expect the topics to be separated from the others (similar to the concept of *separation* in clustering theory).

The two main elements in topic modeling are the documents and its constituents, i.e. the words. To allow the topic model to deal with these elements, we need to find a way to represent them under a computational point of view. This Chapter is therefore organized as follows: we will provide an overview of the main representation methods for words and documents in Section 2.2 and Section 2.3 respectively. In this way, we will have the fundamentals to provide some details on the focus of this thesis, i.e. probabilistic topic models, in

Section 2.4. Since these models are usually controlled by hyperparameters, we will also provide an overview of the main methodologies for estimating the hyperparameters in topic models in Section 2.5.

2.2 WORD REPRESENTATIONS

Language is made of words that under a computational point of view come from a vocabulary and we need to find ways to account for the meaning of these words under a computational point of view. Nowadays, the most common way to represent words in NLP is to use **vectors in a vector space**: words are embedded in a multi-dimensional vector space and we can therefore interpret them as points in a space that can be compared. The process of "embedding" words in the vector space is what brought the community to call these representations *word embeddings*. "Word embeddings" is in fact the general term that is used to refer to this kind of representations.

We can distinguish between two different ways of representing words with vectors: we refer to the first as a **local representation** and to the second one as a **distributed representations** (Ferrone and Zanzotto, 2017). This distinction derives from one of the most vivid debates in the AI field during the '80s on how to store conceptual information inside neural algorithms. Local representations are meant to represent a single concept with the activity of a single neural unit. On the other hand, distributed representations are meant to account for a pattern of activity of more neural units (Hinton et al., 1986).

2.2.1 *Local Representations*

The simplest way to represent words in a way that is interpretable from a machine consists in using **one-hot encoding**. In one-hot encoding, each word is represented by a single and unique vector. One-hot encoding maps the i-th word of the vocabulary V to the vector $\mathbf{w}_i$ in a vector space $\mathbb{R}^n$, where n is the cardinality of the vocabulary V and the i-th element of $\mathbf{w}_i$ is set to 1, while all the other elements are set to zero. We generally refer to this kind of vectors as the *one-hot vectors*. For example, given the words of the vocabulary $V = \{\mathtt{the}, \mathtt{cat}, \mathtt{is}, \mathtt{on}, \mathtt{table}\}$, the word *"cat"* of the vocabulary V can be represented as a vectors of zeros with only one 1 in the position indexed by its own index in the vocabulary $\mathbf{w}_2 = \langle 0, 1, 0, 0, 0\rangle$ (i.e., "cat" is the second element and thus the 1 will be in the second component of the vector).

If we want to represent all the unique words in a text corpus, we will then need a matrix whose dimension is $V \times V$ (where V is the number of unique words). A one-hot encoded representation is simple but results in two main issues. First, the dimension of the matrix grows as the number of unique words increases. Encoding all the

words of the English vocabulary would generate a matrix of at least $170,000 \times 170,000$ entries. Related to this issue, the resulting matrix is extremely sparse (each row is composed of all o-valued entries except for one entry). Each vector is orthogonal to each other, therefore not representing any type of relationship between words. It is instead more convenient that word embeddings reflect and preserve certain specific properties of language. For example, we may agree that the word “cat" is more similar to "dog" than to “Rome". In the next section, we will see how this problem can be addressed by distributional representations.

2.2.2 *Distributional Embeddings*

Distributional semantics is an approach to semantics that advocates a “usage-based” perspective on the computation of word meaning. Distributional semantics is based on the assumption that the statistical distribution and the frequency of usage of words inside textual documents can reveal information about the meaning of words themselves. The intuition that drove the development of the algorithms based on distributional semantics is well described by a famous sentence that was pronounced by J. R. Firth’s *“you shall judge a word by the company it keeps”*(Firth, 1957). In other words, word meaning can be found in the context (Lenci, 2008).

The definition of the concept "context" can vary widely across the different algorithms. The simplest case of context is co-occurrence: a word appears in the context of those words it co-occurs with. We expect the words *“cat”* and *“kitten”* to occur in similar contexts and thus being similar. Let us notice that also the words *cat* and *dog* could co-occur in some contexts, thus making the two words similar, but less similar than *“cat”* and *“kitten”* which co-occur more often. On the other hand, words that occur in different contexts, such as *“smartphone”*, will not be similar to *“cat”*. This effect allows us to define a graded similarity. In other words, the degree of semantic similarity between two words w_1 and w_2 is a function of the similarity of the contexts in which w_1 and w_2 usually appear. We in fact expect that the meaning of the words *“dog”* and *“cat”* to be similar, since both are domestic animals, have four legs, an owner, they eat, and so on.

Models that are based on distributional semantics aim to create representations in which similar vectors should represent similar words (i.e., words that occur in similar contexts). These algorithms take large amounts of text in input to create these vector representations. Figure 2.2 shows an example of what a vector space model built under the distributional hypothesis should create: “cats” and “dogs” are similar words and tend to occur in similar contexts (e.g., those shared by animals, those shared by house pets, etc...) and they tend to share fewer contexts with words like “president”.

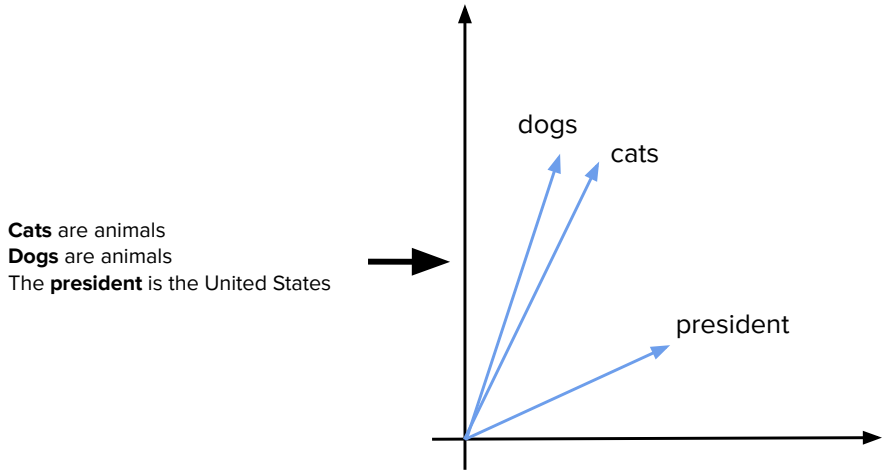


Figure 2.2: An example of word vector representation generated from text.

There are different ways to generate these representations. One of the most famous model that can create distributional representations of words is **Word2vec** (Mikolov et al., 2013). Word2vec is a neural architecture that has been proposed in two different variants: Continuous Bag-of-words (CBOW) and Skip-gram (SG). Both architectures are simple feed-forward neural networks with one hidden layer, and they are trained over a large corpus of text. There are no non-linearities between the layers (except for a softmax function to compute the output scores of the network) and thus the projections are linear. The training examples for the models are extracted from text and are generally based on the concept of target word and context words that appear inside the corpus within a fixed distance from the target word: for example, in a sentence like *"the cat is on the table"*, the word "cat" might be the target word and "the", "is", "on", "the", "table" the context. CBOW gets the context words as input and it aims at predicting the target words. Instead, SG is trained by considering the task in the opposite way: given the target word the model, it aims at predicting the surrounding words of the target. Once the models have been trained, the word embeddings are extracted from the first weight matrix of the neural network.

Word embeddings learned using the Word2vec model exhibit a good capability at capturing syntactic and semantic regularities in a language (Mikolov et al., 2013). In fact, the introduction of Word2vec represents a milestone for the NLP field. Different improved distributional embeddings models have been then proposed across the years (Grave et al., 2018; Pennington et al., 2014) and have become ubiquitous in NLP (Khattak et al., 2019; Rezaeinia et al., 2019; Søgaard et al., 2019). However, these approaches have some limitations. Despite their capabilities of capturing syntactic and semantic regularities, it has been shown that these representations also capture bias in language (Caliskan et al., 2017). Moreover, most of these models also

assign to each word a single vector representation, following that they compress all the senses of a word into a single vector.

2.2.3 *Contextual Embeddings*

The word representations we have seen so far are just static representations of words: each word is associated with a single vector representation, regardless of the context. However, words change their meaning depending on the context in which they appear. Let us consider the following two sentences *"the Broadway play premiered yesterday"* and *"two teams play a football match"*. The word *"play"* in the two sentences has two different meanings and syntactic roles.

Contextualized words embeddings aim at overcoming this issue and capturing word meaning in different contexts. We therefore aim to obtain two different vectors for the same word *"play"*. In particular, we want to obtain a vector representation for the word *"play"* that is dependent on its context. Let be a document composed of $w_1, w_2, \ldots, w_N$ words. Then a context-dependent (or contextualized) vector representation c_k for the k-th word of the document is

$$c_k = f(w_k | w_1, w_2, \ldots, w_N) \in \mathbb{R}^N \tag{1}$$

such that the representation changes for different contexts and f is function that maps the word to a continuous vector representation.

To obtain these vector representations we have to resort to the concept of language modeling. Language modeling is the task of predicting the next word given a sequence of words. For example, given the following sentence *"two teams play a football [BLANK]"*, a language model must predict the word in the *[BLANK]* position, which can be *"match"*. It is intuitive that a language model is therefore required to be able to express syntax (the grammatical form of the predicted word must match its modifier or verb) and to model semantics.

More recent work, namely deep neural language models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and GPT-2 (Radford et al., 2019), have successfully created contextualized word representations. Their internal representations of words are in fact called contextualized word representations because they are a function of the entire input sentence. The success of this approach suggests that these representations capture highly transferable and task-agnostic properties of language (Liu et al., 2019a).

ELMo (Peters et al., 2018) creates contextualized representations of each token by concatenating the internal states of a bidirectional LSTM trained on a bidirectional language modeling task. On the other hand, BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) are transformer-based language models (Vaswani et al., 2017). BERT is bidirectional like ELMo, while GPT-2 is a unidirectional language model. Each transformer layer of BERT and GPT-2 creates a contextu-

alized representation of each token by attending to different parts of the input sentence. BERT – and subsequent iterations on BERT (Liu et al., 2019b; Sanh et al., 2019; Yang et al., 2019) – have achieved state-of-the-art performance on various downstream NLP tasks, ranging from question answering (Liu et al., 2019b; Yang et al., 2019), natural language inference (Yang et al., 2019), and sentiment analysis (Sanh et al., 2019; Yang et al., 2019).

A NOTE ON TRANSFER LEARNING The success of the models and approaches mentioned before is due to their ability to capture semantic and syntactic knowledge and then *transfer* this knowledge to a wide variety of NLP tasks. Transfer learning is a means to extract knowledge from a source setting and apply it to a different target setting (Ruder, 2019). Indeed, there are some properties of language that are task-agnostic and can be highly transferable.

The most successful style of transfer learning in NLP is sequential transfer learning. Sequential transfer learning consists of two stages: a pre-training phase in which general representations are learned on a source task or domain followed by an adaptation phase during which the learned knowledge is applied to a target task or domain. Pre-trained representations are usually obtained from a large unlabelled text corpus using a method of choice (Word2vec, ELMo, BERT, et cetera). While the adaptation consists in adapting the representations to a supervised target task using a smaller set of labelled data. One of the advantages of transfer learning is in fact that we need less training data for the adaptation, given that we already have the knowledge coming from the pre-training phase.

There are two main paradigms for adaptation: *feature extraction* and *fine-tuning*. In feature extraction the model's weights are frozen and the pre-trained representations are used in a downstream model similar to classic feature-based approaches. Alternatively, a pre-trained model's parameters can be unfrozen and fine-tuned on a new task. Pre-trained word vectors (such as Word2vec embeddings) are often fixed and fed into a task specific model. Contextualized word representations have significantly improved over non-contextual vectors and they usually provide significant improvements when fine-tuned with respect to the target task, on the condition that the target task is not too distant from the source task (Peters et al., 2019).

Another exciting advantage of pre-training is related to its multilingual (Pires et al., 2019; Wu and Dredze, 2019) and multimodal capabilities (Radford et al., 2021). In fact, it is possible to jointly train contextual embedding models over multiple languages or multiple modalities (e.g. text and images) without explicit mappings. This would produce a model that is able to perform the so-called *zero-shot cross-lingual (or cross-modal) transfer*. It refers to train a model in a

source language (or modality), often a high resource language, then transfers directly to a target language (modality).

2.3 DOCUMENT REPRESENTATIONS

2.3.1 *Bag of Words*

The Bag-of-Words (BoW) model is a document representation that turns text in natural language into a fixed-length vector. We can obtain the BoW representation of a document by first tokenizing the text, i.e. dividing the words or phrases in tokens. Then we can create a vector, whose length is equal to the number of unique tokens in the texts. The entries of this vector may be binary, thus indicating whether a token occurs (value 1) or not (value 0) in the considered document, or can represent the counts of the tokens in the document. Let us consider the sentences *"The cat is on the table"* and *"The cat and the dog are under the table"*. The vocabulary is composed of the 10 unique words: "The", "cat", "is", "on", "the", "table" "and, "dog", "are", "under". Notice that "The" and "the" are two different words. Considering the previous order of the words, the binary BoW representations of the two sentences are the following vectors: $[1,1,1,1,1,0,0,0,0,0]$ and $[1,1,0,1,1,1,1,1,1,1]$. On the other hand, if we want to consider the counts, we will obtain the following vector representations: $[1,1,1,1,1,0,0,0,0,0]$ and $[1,1,0,1,2,1,1,1,1,1]$.

Let us notice that the BoW model loses the contextual information of a document. the sentences "the department chair couches offers" and "the chair department offers couches" are represented by the same bag of words, but have different meanings. We do not know anymore in which position a given word appeared or which were their surroundings words. Nonetheless, this kind of representation can be useful from the point of view of the computational costs.

2.3.2 *Distributed Representations of Sentences and Documents*

Despite their popularity, bag-of-words representations have two major weaknesses: they lose the ordering of the words and they also ignore semantics of the words. For example, the words "cat," "dog" and "Rome" are equally distant in a bag-of-word representation. This is analogous to the observations we made for the one-hot encoded representations of words. Indeed, as seen before, we can address these problems by resorting to distributed representations of documents. To this end, we can use algorithms that maps a variable-length document to a fixed-length distributed representation.

A simple strategy consists in representing the document as a concatenation or average of its surrounding words, and the resulting vector is used to predict other words in the context (Bengio et al.,

2006). Then the embedded document representation can be exploited in downstream tasks, such as clustering and retrieval.

2.3.3 *Contextualized Document and Sentence Embeddings*

In the previous sections, we have seen that we can derive contextualized representations of words using recent neural language models (Devlin et al., 2019). We can of course learn also contextualized representations of documents, i.e. fixed-length representations that derive from contextualized language models. For some time, transfer learning in NLP was limited to pre-trained word embeddings, but recent work has demonstrated strong transfer task performance using pre-trained sentence embeddings (Cer et al., 2018). The most commonly used approach is to average the BERT (Devlin et al., 2019) output layer or by using the output of the first token (the [CLS] token). However, this common practice yields rather bad sentence embeddings (Reimers and Gurevych, 2019). Other approaches have been investigated. One of the most used is Sentence BERT (Reimers and Gurevych, 2019, SBERT), which is a modification of the pre-trained BERT network that use siamese and triplet network structures.

2.4 PROBABILISTIC TOPIC MODELS

In the introduction of this Chapter, we have provided an overview of topic modeling. Here, we will describe how the elements that compose a topic model can be interpreted in probabilistic terms, originating the prominent class of probabilistic topic models (Blei, 2012; Zhai, 2017).

We have anticipated that topics are not just unsorted lists of keywords. Instead, they are associated with a weight or, rather, a probability weight. We can in fact express **a topic as a multinomial distribution over the vocabulary**, where the most likely words are the representative words of the given topic. We can therefore select the top-n most likely words to represent a topic.

In addition, topic models also provide a lower-dimensional representations of the documents in the space of the topics. Also this representation can be interpreted as a probability distribution: **a document is in fact a multinomial distribution over the topics**. In other words, a document can talk about different topics in different percentages. Reporting the example mentioned in the introduction, an NLP paper can talk about 30% of linguistic and 70% of computer science.

The only observations usually provided to a topic model are the documents and their words. We can imagine a generative process that have generated the words of the documents. A generative process is in fact the imaginary random process by which the model assumes the documents are constructed through the sampling of their words

(observed random variables). The latent topics (latent random variables), which have ideally produced the collection of documents, are inferred by reversing the generative procedure of a text.

2.4.1 *Latent Dirichlet Allocation (LDA)*

To better explore these concepts, we now describe the most well-known topic model Latent Dirichlet Allocation (Blei et al., 2003a, LDA). This model makes the assumptions considered above: it represents the topics as mixtures of words in the vocabulary (i.e. multinomial distributions over the vocabulary) and the documents as mixtures of topics (i.e. multinomial distributions over the topics). It also assumes that each word in a document is associated with a single topic.

LDA is a **probabilistic graphical model**. A graphical model can be represented by a graph, graphically represented in "plate notation". Figure 2.3 reports LDA's representation in plate notation. Here, the nodes of the graph represent the random variables and an edge between two nodes represents the conditional dependency relationships among the variables. Observed variables are represented by shaded circles (e.g. W_{nd} is the variable representing the words for LDA, which are observed, and in fact is represented by a shaded circle). Moreover, if a variable is contained into a plate, then the variables are replicated multiple times (as the number reported on the corner of the plate). For more details on graphical models, we refer to Appendix A.

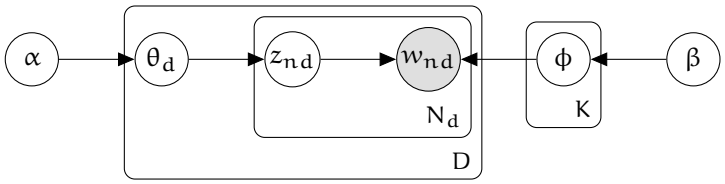


Figure 2.3: LDA in plate notation. The variable w_{nd}, representing the n-th word of document d, is observed, then it is represented by a gray circle. While the other variables are unobserved, thus they are represented by white circles. Variables are repeated if they are included in a rectangle.

More formally, let be K the fixed number of topics, D the documents and V the unique words of the vocabulary. In LDA, the only observations are the words w, and each word is associated with a topic assignment z. The topic assignments z are i.i.d (identically and independently distributed) drawn from a document-topic distribution θ and word tokens are i.i.d. drawn from a topics' distributions over words ϕ. In other words, the words of the documents in LDA are represented as BOWs, because the order does not count. The random

variables θ and ϕ are multinomial distributions and are controlled by the Dirichlet priors α and β respectively.

The generative process of the documents in LDA is the following:

for each topic $k \in K$ **do**
  Draw a distribution over words $\phi_k|\beta \sim \text{Dir}(\beta)$
end for
for each document $d \in D$ **do**
  Draw a vector of topic proportions $\theta_d|\alpha \sim \text{Dir}(\alpha)$
  for each word w in document d **do**
    Draw a topic assignment $z_{nd}|\theta_d \sim \text{Mult}(\theta_d)$, where $z_{nd} \in \{1, \dots, K\}$
    Draw a word $w_{nd}|z_{nd}, \phi_{z_{nd}} \sim \text{Mult}(\phi_{z_{nd}}), w_{nd} \in \{1, \dots, V\}$.
  end for
end for

where $\text{Dir}(\cdot)$ and $\text{Mult}(\cdot)$ represent the Dirichlet and Multinomial distributions respectively. The full joint distribution of LDA, given its hyperparameters, is shown in Equation 2:

$$p(\theta, \mathbf{z}, \boldsymbol{\phi}, \mathbf{w}|\alpha, \beta) \tag{2a}$$
$$= p(\boldsymbol{\phi}|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|z, \phi_z) \tag{2b}$$
$$= \underbrace{\prod_{k=1}^{K} p(\phi_k|\beta)}_{\text{topic plate}} \cdot \overbrace{\prod_{d=1}^{D} p(\theta_d|\alpha) \underbrace{\prod_{n=1}^{N_d} p(z_{nd}|\theta_d)p(w_{nd}|z_{nd}, \phi_{z_{nd}})}_{\text{word plate}}}^{\text{document plate}} \tag{2c}$$

The goal is to compute the posterior distribution of the latent variables, given the observed documents. Therefore, the generative process of a document must be reversed in order to obtain the distribution of the hidden variables:

$$p(\theta, \mathbf{z}, \boldsymbol{\phi}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \boldsymbol{\phi}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{3}$$

The denominator of Equation 3 is intractable to be computed by means of exact inference methods. In fact, the posterior probability of LDA – and any other probabilistic topic model – is usually computed by approximate inference algorithms.

TRAINING AND INFERENCE Different algorithms have been proposed to optimize inference for LDA (Blei et al., 2003a; Griffiths and Steyvers, 2004; Hoffman et al., 2010; Zhai et al., 2012). In the following, we will report the two main foundation methods that are used for training topic models. Although we will see the details of these methods applied for LDA, these methods can be generalized to most of probabilistic models.

- **Collapsed Gibbs Sampling (GS).** Markov Chain Monte Carlo (MCMC) methods are a class of approximate inference algorithms, which are scalable and allow sampling from a large class of distributions. One of the most widely used algorithms that belongs to this category is Gibbs Sampling (GS) (Bishop, 2006).

  The procedure starts from some initial state for the Markov chain and, at each step, it replaces a value for i-th variable with the value drawn from the distribution of that variable conditioned on the values of the remaining variables. This procedure is repeated through all the variables, for T steps. Given $p(\mathbf{z}) = p(z_1, z_2, \ldots, z_M)$ to sample, Gibbs sampling procedure is shown below:

  Randomly initialize $z_i : i = 1, \ldots, M$
  for $t = 1, \ldots, T$ **do**
  Sample $z_1^{(t+1)} \sim p(z_1 | z_2^{(t)}, z_3^{(t)}, \ldots, z_M^{(t)})$
  Sample $z_2^{(t+1)} \sim p(z_2 | z_1^{(t+1)}, z_3^{(t)}, \ldots, z_M^{(t)})$
  $\vdots$
  Sample $z_j^{t+1} \sim p(z_j | z_1^{t+1}, \ldots, z_{j-1}^{(t+1)}, z_{j+1}^{(t)}, \ldots, z_M^{(t)})$
  $\vdots$
  Sample $z_1^{(t+1)} \sim p(z_M | z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{M-1}^{(t+1)})$
  end for

  After sampling a sufficient number of samples, GS is proved to converge to the exact posteriors. However, it is hard to estimate how many iterations are required for the algorithm to converge. Choosing a number of iterations which is too high can lead to a computationally demanding execution for large-scale applications.

  Collapsed Gibbs Sampling (CGS) consists of integrating out the parameters, thus it samples from a distribution with one or more of the conditioned variables integrated out. Collapsed methods, also referred as Rao-Blackwellisation methods (Casella and Robert, 1996), improve performance in terms of velocity in reaching the target distribution.

  CGS can be employed for LDA (Griffiths and Steyvers, 2004), as both document-topic distribution θ_d and word-topic distribution ϕ_t can be calculated using just the topic assignments $\mathbf{z}$.[1] Instead of the full joint distribution $p(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\phi} | \alpha, \beta)$ shown in

1 The topic assignments $\mathbf{z}$ are a sufficient statistics for θ_d and ϕ. In fact $\theta_{dz} = \frac{n_{dz}+\alpha}{n_{.z}+K\alpha}$ and $\phi_{zw} = \frac{n_{zw}+V\beta}{n_{zw}+\beta}$

Equation 2, θ_d and ϕ_t are first integrated out, obtaining the distribution $p(\mathbf{z}, \mathbf{w}|\alpha, \beta)$ as it follows:

$$p(\mathbf{z},\mathbf{w}|\alpha, \beta) \tag{4a}$$

$$= \int_\theta \int_\phi p(\theta, \mathbf{z}, \boldsymbol{\phi}, \mathbf{w}|\alpha, \beta)\, d\theta d\phi \tag{4b}$$

$$= \int_\phi \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{d=1}^{D} \prod_{n=1}^{N_d} p(w_{nd}|z_{nd}, \phi_{z_{nd}})\, d\phi \tag{4c}$$

$$\cdot \int_\theta \prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_{nd}|\theta_d)\, d\theta \tag{4d}$$

Since θ and ϕ only appear in the first and second terms, respectively, these integrals can be performed separately. The joint distribution marginalized over θ and ϕ then becomes:

$$p(\mathbf{z},\mathbf{w}|\alpha, \beta) \tag{5a}$$

$$= \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha_i)}{\prod_{k=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{k=1}^{K} \Gamma(n_{dk} + \alpha_i)}{\Gamma(\sum_{k=1}^{K} n_{dk} + \alpha_i)} \tag{5b}$$

$$\cdot \prod_{k=1}^{K} \frac{\Gamma(\sum_{w=1}^{V} \beta_w)}{\prod_{w=1}^{W} \Gamma(\beta_w)} \frac{\prod_{w=1}^{V} \Gamma(n_{wk} + \beta_w)}{\Gamma(\sum_{w=1}^{V} n_{wk} + \beta_w)} \tag{5c}$$

where $\Gamma(\cdot)$ denotes the gamma function.

The goal of Gibbs sampling is to approximate the distribution $P(z_{nd}|\mathbf{w}, \alpha, \beta)$. The resulting collapsed Gibbs sampling equation for LDA can be written as

$$P(z_{nd} = t|\mathbf{z}^{\neg nd}, w) \propto (N_{dt}^{\neg nd} + \alpha) \frac{N_{tw}^{\neg nd} + \beta}{N_{t\cdot}^{\neg nd} + V\beta} \tag{6}$$

where the superscript $\neg nd$ signifies leaving the nth token of the d-th document out of the calculation. This notation will be used throughout this thesis. For simplicity, the hyperparameters α and β are assumed to be symmetric. Further details about collapsed Gibbs sampling for LDA can be found in (Griffiths and Steyvers, 2004).

- **Variational Inference.** Variational inference is a class of deterministic approximate techniques (Sun, 2013), which are an alternative to MCMC methods. The variational approach aims to approximate a posterior distribution by looking for a distribution from a family of distributions which is tractable and is the closest to the true posterior, where the closeness is generally measured by the relative entropy (or Kullback-Leibler divergence).

  More formally, the aim is to minimize the KL-divergence between the true posterior $p(\mathbf{z}|\mathbf{x}, \alpha)$ and the family of distributions $q(\mathbf{z}|\nu)$:

$$\nu^* = \arg\min_{\nu} \mathrm{KL}\left(q(\mathbf{z}|\nu)||p(\mathbf{z}|\mathbf{x}, \alpha)\right) \tag{7}$$

where $\mathbf{x}$ and $\mathbf{z}$ are arrays of observations and latent variables respectively, α is an array of fixed parameters and ν is an array of free variational parameters. The log-likelihood of p can be limited by using Jensen's inequality:

$$\log p(x) = \log \left[\int_z p(x,z) \right] = \log \left[\int_z p(x,z) \frac{q(z)}{q(z)} \right] \tag{8}$$

$$= \log \left[\mathbb{E}_q \left[\frac{p(x,z)}{q(z)} \right] \right] \geqslant \mathbb{E}_q \left[\log \left[\frac{p(x,z)}{q(z)} \right] \right] \tag{9}$$

Thus Jensen's inequality provides a lower bound, also called *evidence lower bound* (ELBO), over the log-likelihood for an arbitrary variational distribution $q(\mathbf{z}|\nu)$:

$$\log p(\mathbf{x}|\alpha) \geqslant \mathbb{E}_q \left[\log p(\mathbf{x},\mathbf{z}|\alpha) \right] - \mathbb{E}_q \left[\log q(\mathbf{z}|\nu) \right] \tag{10}$$

It can be shown that maximizing the lower bound is equivalent to minimizing the KL-divergence. The optimal values for the parameters ν can be obtained using standard nonlinear optimization techniques (Bishop, 2006).

One issue of the variational inference is the choice of the family of distributions, which are to be simpler than the true posterior. For instance, factorized distributions can be used, i.e. distributions whose hidden variables of interest are forced to be independent. Such case is referred as mean field approximation (Parisi, 1988). Indeed LDA's inference issue can be solved by using a mean field variational approach. By dropping the problematic edges and nodes, the simplified variational distribution is as it follows:

$$q(\theta, z|\gamma, \tau) = q(\theta|\gamma) q(z|\tau) \tag{11}$$

where γ and τ are the variational parameters. The next step is to find the optimal values for γ and τ, by solving the optimization problem explained above.

Variational inference is by far faster than MCMC methods. However a variational method does not reach convergence, since the selected distribution is just an approximation of the true posterior, and finding a proper family of distributions can be difficult.

A relevant feature of probabilistic topic models is that they are modular and can therefore be extended. In the following Chapters, we will see different extensions of topic models originated from LDA.

2.5 HYPERPARAMETER SELECTION

We will now provide the background on an essential topic for this thesis. An important element of topic models is the hyperparameters.

First, it is important to clarify the distinction between parameters and hyperparameters, which are two recurrent terms in machine learning models. A *parameter* is a internal variable of the model that can be estimated or learned from the data. On the other hand, model's *hyperparameters* cannot be learned during training but are set beforehand.

For example, let us consider LDA. The word-topic and the document-topic probability distributions are parameters of the model, because we estimate them during the training. While, the number of topics is a hyperparameter because we have to set it before starting the training.[2] Hyperparameters strongly affect the results and performance of a model. Let us imagine to compare the results of a topic model with 2 topics and a topic model with 500 topics, run on the same corpus. The first model will return coarse-grained and separated topics, while the other model will return finer-grained and possibly overlapping topics. It is then important to carefully select the hyperparameters by adopting an appropriate search strategy, which is computationally tractable and effective.

Let us assume we have to find the optimum of an unknown objective function f. Then, we are considering the problem of finding a global maximizer (or minimizer) of f:

$$\mathbf{x}^* = \arg\max_{x \in \mathcal{X}} f(x) \tag{12}$$

where $\mathcal{X}$ is a design space of interest. This space can be composed of hyperparameters of different types: categorical, continuous or also conditional inputs.

In the following, we will talk about the most well-known techniques for selecting the hyperparameters in machine learning. We focus in particular on Bayesian Optimization, since it is a technique we will extensively use in following Chapters.

2.5.1 *Grid Search*

The traditional way of performing hyperparameter selection is grid search. It consists in an exhaustive search of the hyperparameters through a manually specified subset of the hyperparameter space $\mathcal{X}$. Grid search exhaustively considers all parameter combinations, Then the selected hyperparameter configuration is then one that returned the best results, according to a performance metric.

2 There exist some topic models that are called non-parametric (Paisley et al., 2014; Teh et al., 2004) and are able to estimate the number of topics from the data. Moreover, the (hyper)parameters α and β, which respectively control the document-topic distribution and topic-word distribution, can be considered as both parameters and hyperparameters. They are usually set a priori (we will therefore call them hyperparameters in this specific case), but they can also be estimated from the data using Expectation-Maximization techniques. Throughout this thesis, we will usually consider α and β as hyperparameters.

Grid search is reliable on low-dimensional space (1-dimensional or 2-dimensional), but suffers from the curse of dimensionality (Bergstra and Bengio, 2012). However, this technique can be parallelizable because the hyperparameter settings it evaluates are typically independent of each other.

2.5.2 *Random Search*

Random Search instead selects the hyperparameter configurations to test randomly. This approach generalizes to continuous and mixed spaces. It can outperform grid search (Bergstra and Bengio, 2012). Random search is in fact more efficient than grid search in high-dimensional spaces if the objective function can be approximated by another function with less variables (hyperparameters). If the researcher could know ahead of time which subspaces would be important, then they could design an appropriate grid, however this is not always feasible.

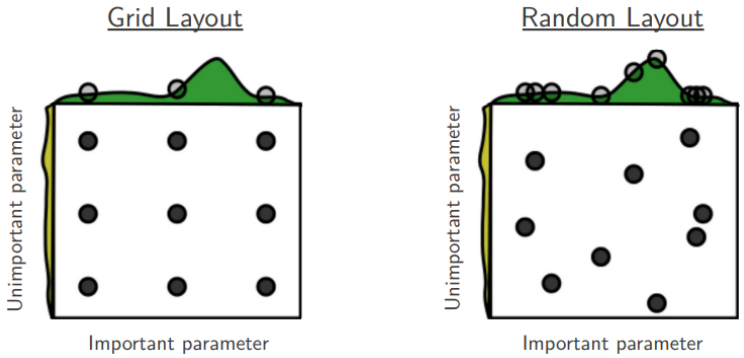


Figure 2.4: Grid and random search of 9 trials for optimizing a function with low effective dimensionality (i.e. it can be approximated by another function with lower number of hyperparameters). Source: (Bergstra and Bengio, 2012).

Figure 2.4 shows a comparison between a grid search approach and a random search approach, when two hyperparameters are involved and one of the two hyperparameters (unimportant parameter) does not change the objective function. It is evident that 6 of the total 9 trials of grid search are indeed ineffective. With random search, all the trials explore distinct values of the objective function. This failure of grid search is the rule rather than the exception in high dimensional hyperparameter optimization.

Random Search is also parallelizable, and additionally allows the inclusion of prior knowledge by specifying the distribution from which to sample.

2.5.3 *Bayesian Optimization*

Bayesian Optimization (Archetti and Candelieri, 2019; Snoek et al., 2012) is a sequential model-based optimization strategy for expensive and noisy black-box functions. The basic idea consists of using all the model's configurations evaluated so far to approximate the value of the performance metric (objective function) and then selects a new promising configuration to evaluate. Figure 2.5 illustrates this sequential process.

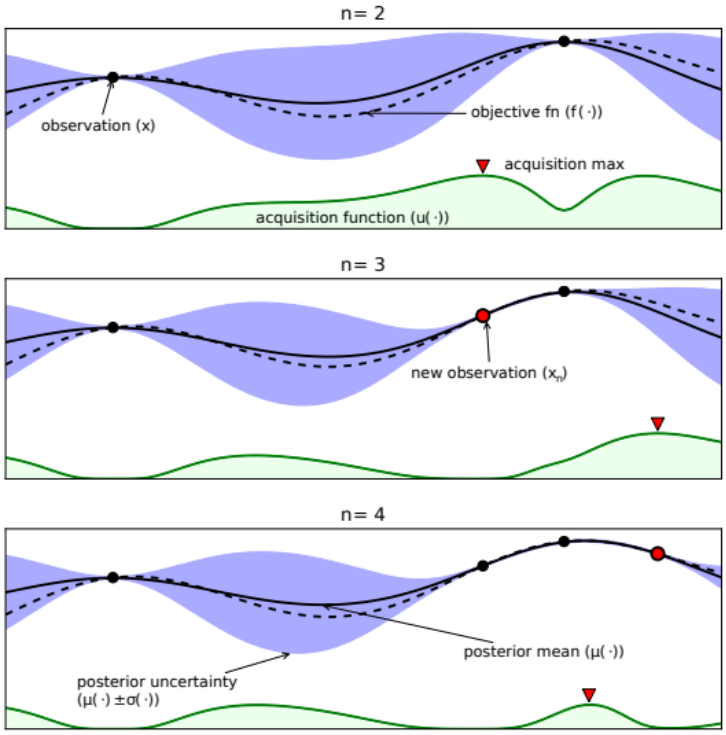


Figure 2.5: Illustration of the Bayesian optimization procedure over three iterations. The plots show the mean and confidence intervals estimated with a probabilistic model of the objective function. The acquisition functions is represented by the lower shaded plots. The acquisition is high where the model predicts a high objective (exploitation) and where the prediction uncertainty is high (exploration). Source: (Shahriari et al., 2016).

The two key components are the *probabilistic surrogate model* aimed at approximating the performance metrics to optimize, and the *acquisition function* that uses the mean of the surrogate model and the confidence (i.e. its standard deviation) to select the next configuration. The approximation is provided by a probabilistic *surrogate model*, which describes the prior belief over the objective function using the observed configurations. The next configuration to evaluate is selected through the optimization of the acquisition function, which leverages the uncertainty in the posterior to guide the exploration.

We report the general BO algorithm in the following pseudo-code.

Table 2.1: A synthetic Bayesian Optimization algorithm.

INPUT:
m_0, number of initial configurations (aka initial design)
M, number of configurations selected through BO

1: define the initial set of configuration $X_{1:m_0}$
2: compute the performance metrics $f_{1:m_0}$
3: $n \leftarrow m_0 + 1$
4: **while** $m \leqslant m_0 + M$ **do**
5: fit the probabilistic surrogate model on $(X_{1:m}, f_{1:m})$
$\Rightarrow$ update $E[x]_m$) and $S[x]_m$
6: compute the acquisition function $\delta_m(x)$, based on $E[x]_m$ and $S[x]_m$
7: select the next configuration according to $x_{m+1} = \text{argmax}\ \delta_m(x)$
8: compute the associated performance metrics f_{m+1}
9: $X_{m+1} \leftarrow X_{1:m} \cup \{x_{m+1}\}$
10: $f_{m+1} \leftarrow f_{1:m} \cup \{f_{m+1}\}$
11: $m \leftarrow m + 1$
12: **endwhile**

OUTPUT:
$x^+ = x_{i^*}$ and $f^+ = f_{i^*}$, where $i^* = \text{argmax}_i\ f_{i=1:m}$

$E[x]_m$ and $S[x]_m$ denote, respectively, the mean and the standard deviation of the prediction provided by the probabilistic surrogate model after m evaluated configurations of the probabilistic topic model under optimization. The next configuration to evaluate is the one maximizing the acquisition function.

2.6 NOTATION

Table 2.2 summarizes the most important mathematical notation that we use in this thesis. Other notations relevant for specific chapters will be introduced when needed.

Note that we will also use subscripts and superscripts to generally identify elements of a sequence (e.g., t_i is the i-th topic of a list of topics).

Symbol	Description	Symbol	Description
D	set of documents	K	fixed number of topics
V	vocabulary	N_d	number of words of document d
ϕ	word-topic distribution	ϕ_k	distribution of topic k over the vocabulary
θ	document-topic distribution	θ_d	distribution of topics in document d
α	Dirichlet hyperparameter for θ	β	Dirichlet hyperparameter for ϕ
$\mathbf{w}$	word variable	w_{nd}	the nth word in document d
$\mathbf{z}$	topic assignments variable	z_{nd}	topic assignment of the nth word of document d
N_{dz}	number of words associated with the topic z in document d	$N_{z.}$	number of words associated with the topic z in the corpus

Table 2.2: Main notations for LDA and its extensions.

3

RELATED WORK

In this Chapter, we provide an overview of the topic models that encode additional information and the state-of-the-art approaches for the evaluation of topic models. In particular, we will focus our attention on two main sub fields: classical probabilistic topic models originated from Latent Dirichlet Allocation and neural topic models, i.e. probabilistic topic models based neural networks. We will therefore describe the extensions of LDA that encode additional information in Section 3.1 and the main approaches of neural topic modeling in Section 3.2. We will also focus on a particular category of topics models which deal with multilingual corpora, called multilingual topic models, in Section 3.3. Finally, we will describe the methods for evaluating a topic model (Section 3.4) and for estimating its hyperparameters (Section 3.5). We will conclude with a summary of the contributions of this thesis in Section 3.6 to outline the main research issues addressed in this thesis.

3.1 BEYOND LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (Blei et al., 2003a, LDA) is the foundation model of numerous topic models extensions (Chauhan and Shah, 2021). (For additional details on LDA, we refer the readers to Section 2.4.) Due to its modularity, LDA can in fact be easily extended: researchers can focus on a single module or building block of the model to extend, keeping the other modules and inference process as they are. As already anticipated, we can encode additional information into LDA. This modification usually allows the model to obtain better topics and better topical representations of the documents.

The extensions of LDA mainly regard the two fundamental elements of the topic model: words and documents. We will therefore describe these two families of topic models the encode word information and document information respectively, focusing on the most relevant approaches for this thesis.

3.1.1 *Modeling Word Information*

LDA assumes that words in a document are identically and independently distributed. In other words, a document is seen as a bag of words. No additional information about words is encoded. However, we do know that words in documents share some kind of relationships, both syntactic and semantic. Several extensions of LDA try to

encode this word information. We can roughly divide the approaches into two categories: topic models that encode word-order (Fei et al., 2014; Gruber et al., 2007; Lindsey et al., 2012; Wallach, 2006; Wang et al., 2007) and syntactic dependencies (Boyd-Graber and Blei, 2008; Griffiths et al., 2004), and models that incorporate domain-specific or semantic information (Andrzejewski et al., 2009, 2011; Chen et al., 2013b; Yang et al., 2015c).

In the following, we consider two of the main approaches for incorporating additional information into topic models, including word information, i.e. Constrained LDA (Yang et al., 2015c) and MetaLDA (Zhao et al., 2017). We will use the notation defined in Section 2.6. Other notation that is relevant to a specific model will be introduced when needed.

CONSTRAINED LATENT DIRICHLET ALLOCATION (CLDA). Constrained LDA (Yang et al., 2015c, CLDA) is an extension of LDA that uses of a potential function to constrain the topics. Topics should reflect the additional information incorporated into the model, e.g. word correlations. We denote the information to incorporate into the model by a set L.[1] Each element $l \in L$ of the information set is introduced into the model by a potential function $f_l(z, w, d)$, which represents a real-valued score for the hidden topic assignment z of the word w in document d.

The information to incorporate L defines the score

$$\xi(\mathbf{z}, L) = \prod_{z \in \mathbf{z}} \exp f_l(z, w, d) \tag{13}$$

that smooths the current topic assignment $\mathbf{z}$. Since CLDA is an extension of LDA, the joint probability distribution of this class of topic models is defined as follows:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{y}, \theta, \boldsymbol{\phi} | \alpha, \beta, \eta, \nu, L) \tag{14a}$$

$$= \underbrace{P(\boldsymbol{\phi}|\beta)}_{\text{Topic plate}} \underbrace{P(\theta|\alpha) P(\mathbf{z}|\theta) P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi})}_{\text{Document plate}} \underbrace{\xi(\mathbf{z}, L)}_{\substack{\text{Potential} \\ \text{function}}} \tag{14b}$$

where the document and topic plate refer to the modules of LDA. The potential function ξ can be factored out of the marginalized joint

1 The paper refers to the information regarding words or documents as to *prior knowledge*. We realize that this term may create ambiguity, since it is often used in different contexts (knowledge transfer, knowledge base, knowledge graph, et cetera). For the sake of uniformity and to avoid ambiguity, we will use the term *information* instead.

distribution, because it does not depend on the distributions ϕ and θ, obtaining the following marginalized joint probability distribution:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{y}|\alpha, \beta, \eta, \nu, L) \tag{15a}$$

$$= \int\int p(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\beta) p(\mathbf{z}|\theta) p(\theta|\alpha) \xi(\mathbf{z}, L) d\theta d\phi \tag{15b}$$

$$= \xi(\mathbf{z}, L) \int\int p(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\beta) p(\mathbf{z}|\theta) p(\theta|\alpha) d\theta d\phi \tag{15c}$$

Potential functions can be defined to constrain elements of the topic model to share or not share the same topics. Here we show the potential function that the authors propose to incorporate domain-specific information as word constraints. Word-related information is represented as word must-link constraints and cannot-link constraints (Andrzejewski et al., 2009). A must-link relation between two words indicates that the two words tend to be related to the same topics, i.e. their topic probabilities should be similar. In contrast, a cannot-link relation between two words indicates that these two words should not both be prominent within the same topic. For example, "quarterback" and "fumble" are both related to American football, so they can share a must-link relation. But "fumble" and "bank" imply two different topics, so they share a cannot-link.

The authors define two information sets for each word w in the vocabulary: a *must-constraint* set L_w^m, containing words that must share the same topics of w, and a *cannot-constraint* set L_w^c, including words that cannot share the same themes of w. Considering the above example, a must-constraint set for the word "fumble" would be

$$L_{\text{fumble}}^m = \{\textit{quarterback}\}$$

and analogously, a cannot-constraint set could be

$$L_{\text{fumble}}^c = \{\textit{bank}\}$$

Given the sets for must-constraints and cannot-constraints for each word of the vocabulary, a potential function of sampling topic t for the word w in document d can be defined as follows:

$$f_m(z, w, d) = \sum_{u \in L_w^m} \log \max(\lambda, n_{u,z}) + \sum_{v \in M_w^c} \log \frac{1}{\max(\lambda, n_{v,z})} \tag{16}$$

The information about the word w will make an impact on the conditional probability of sampling the hidden topic z. Unlike standard LDA where every word's hidden topic is independent of other words given θ, here the CLDA increases the probability that a word w will be drawn from the same topics as those of w's must-link set, and decreases its probability of being drawn from the same topics as those of w's cannot-link word set. Here λ is a hyperparameter that controls the strength of the relationships between the words. . For large λ, the

constraint is inactive for topics except those with the large counts. As λ decreases, the constraint becomes active for topics with lesser word counts.

The advantage of CLDA is that different definitions of the potential functions lead to models that incorporate different types of information. This approach is simple to implement and modular. We will later see how to incorporate document labels into LDA following a similar method.

METALDA. MetaLDA (Zhao et al., 2017) is a model that can leverage arbitrary document and word information encoded in binary form. Therefore, this model belongs both to the family of topic models that encode word information and models that encode document information.

Let us recall that LDA uses the same Dirichlet prior for all the document-topic distributions and the same prior for all the topic-word distributions. In MetaLDA, each document has a specific Dirichlet prior on its topic distribution, which is computed from the meta information of the document (e.g. labels), and the parameters of the prior are estimated during training. Similarly, each topic has a specific Dirichlet prior computed from the word meta information. More specifically, the labels of a document d are encoded in a binary vector $b_d \in \{0,1\}^{L_{doc}}$ where L_{doc} is the total number of unique labels. $b_{dl} = 1$ indicates label l is active in document d and vice versa. Similarly, the L_{word} features of word token v are stored in a binary vector $g_v \in \{0,1\}^{L}_{word}$. Therefore, the document and word meta information are stored in a matrix $B \in \{0,1\}^{D \times L_{doc}}$ and $G \in \{0,1\}V \times L_{word}$ respectively.

At the document level, if two documents have labels in common, their Dirichlet parameter α_d will be more similar, resulting in more similar topic distributions θ_d. Similarly, at the word level, if two words w and w' have similar features, the priors β_{kw} and $\beta_{kw'}$ in topic k will be similar and then we can expect that their ϕ_{kw} and $\phi_{kw'}$ could be more similar. Finally, the two words will have similar probabilities of showing up in topic k. The word-level information that the authors consider is derived from binarized pre-trained word embeddings.

3.1.2 *Modeling Document Information*

Documents in a corpus may be associated with metadata (e.g. labels, the authors, timestamps, links among documents). This information can be introduced into the topic models to encourage documents with the same metadata to be characterized by similar topic distributions. Usually these models are called "supervised topic models", and they can be divided in two categories. In downstream supervised topic

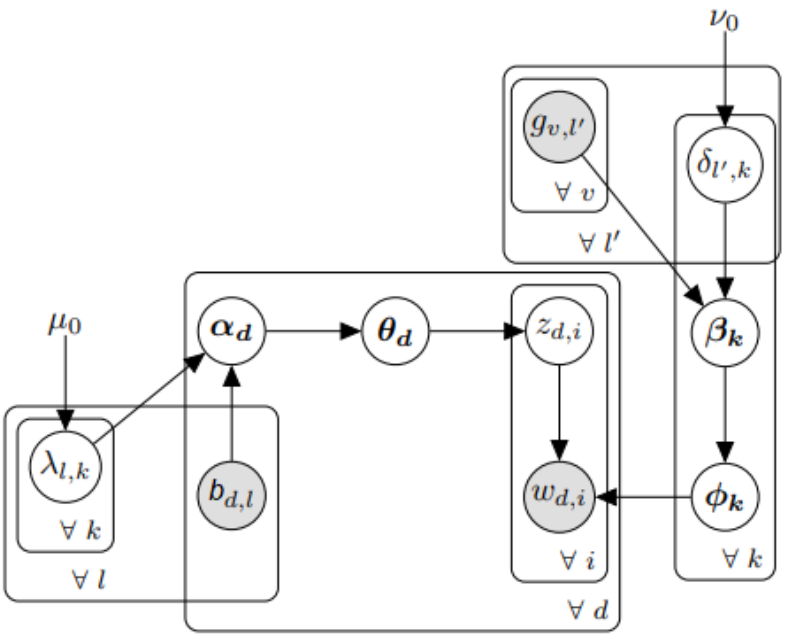


Figure 3.1: MetaLDA in plate notation.

models (Blei and Jordan, 2003; Blei and McAuliffe, 2007; Wang and McCallum, 2006), the response variable is predicted based on the latent representation of the document. Downstream models are typically better at prediction tasks. Supervised LDA (Blei and McAuliffe, 2007) is the most well-know example of downstream supervised approach. It extends LDA by modeling the additional response variable y_d, that can be modeled in different types (e.g. real-values, categoricals or binary), and is conditioned on the topic assignments of the document d.

Upstream supervised topic models go in the opposite direction: the response variable is being conditioned to generate the latent representation of the document (Chang and Blei, 2009; Lacoste-Julien et al., 2008; Li et al., 2015; Mimno and McCallum, 2007; Ramage et al., 2009, 2011; Rosen-Zvi et al., 2004; Yang et al., 2015c; Zhu et al., 2012). These models often lead to more interpretable topics (Yang et al., 2015c).

These supervised approaches encode information of labels or multiple labels (Lacoste-Julien et al., 2008; Li et al., 2015; Ramage et al., 2009, 2011; Zhu et al., 2012), authors (Mimno and McCallum, 2007; Rosen-Zvi et al., 2004) time (Blei and Lafferty, 2006; Wang and McCallum, 2006), or they model relationships among documents (Chang and Blei, 2009; Chen et al., 2013a; Yang et al., 2016a; Zhang et al., 2013).

We will now describe a variant of Constrained LDA that allows to encode document information into topic models.

LABELED LATENT DIRICHLET ALLOCATION (LLDA). Labeled LDA (LLDA), as defined in (Yang et al., 2015b), is a model that introduces information through the use of potential functions. Here, each piece of information $l \in L$ is introduced into the model by a potential function $f_l(z, d)$, which represents a real-valued score for the hidden topic assignment z in document d. The information L defines a score $\xi(\mathbf{z}, L) = \prod_{z \in \mathbf{z}} \exp f_l(z, d)$ that smooths the current topic assignment $\mathbf{z}$. The joint probability distribution of LLDA is then defined as follows:

$$P(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi} | \alpha, \beta, L) \propto P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}) P(\boldsymbol{\phi}|\beta) P(\mathbf{z}|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\alpha) \xi(\mathbf{z}, L) \quad (17)$$

LLDA defines a potential function that models a one-to-one correspondence between a topic and a label associated to a document. This correspondence is modeled with the function $g : \Gamma \mapsto K$ that maps a label to its corresponding topic, where Γ is the set of labels. LLDA introduces document-level information in LDA by including in its joint probability distribution the following potential function:

$$f(z, d) = \begin{cases} 1 & \text{if } z = g(l_d) \\ -\infty & \text{otherwise} \end{cases} \quad (18)$$

where $l_d \in \Gamma$ specifies the document label.

Modeling Relational Information into Topic Models.

A particular category of topic models that encode document-level information consists in those topic models that consider the underlying network structure in a document corpus (e.g. co-authorship networks, social networks, citation networks). These models usually assume that linked documents are more likely to have similar topic representations.

The main class of approaches is represented by Relational Topic Model (Chang and Blei, 2009, RTM) and its extensions (Chen et al., 2013a; Yang et al., 2016a; Zhang et al., 2013) that, grounding on Latent Dirichlet Allocation (Blei et al., 2003a, LDA), model each link as a binary variable considering the existence (or absence) of a link between a couple of documents. Generalized RTM (Chen et al., 2013a) can capture not only same-topic interactions between documents, but all pairwise topic interactions, while Sparse RTM (Zhang et al., 2013) aims at inferring sparse topics for each document by presenting a non-probabilistic formulation of RTM. In (Yang et al., 2016a), RTM also embeds the Weighted Stochastic Block Model (Aicher et al., 2015) to identify groups of documents that are densely connected and should talk about similar topics.

Other approaches are the regularized topic models (He et al., 2017; Mei et al., 2008), which augment the topic model objective function with a network regularization penalty that encourages topic mixtures

of related documents to be similar, and Dirichlet Multinomial Regression (Mimno and McCallum, 2008) and its extensions (Hefny et al., 2013; Wahabzada et al., 2010), which incorporate arbitrary features by considering links as per-document attributes.

We will now focus on two of these main approaches, which will be later used in the following Chapters.

RELATIONAL TOPIC MODELS Relational Topic Model (Chang and Blei, 2009) derives from Latent Dirichlet Allocation and the mixed-membership models (Erosheva et al., 2004). It exploits the former model to represent the content of a document, i.e. words, and the latter to model the network of documents. In particular, links are modeled by a link probability function that depends on the topic assignments z_d of two documents, modeling the idea that documents with similar topic assignments of words are likely to be linked.

Therefore the observed documents' words w_{nd} and binary links $y_{dd'}$ between them are generated by the following process:

for each topic $k \in K$ **do**
  Draw a distribution over words $\phi_k|\beta \sim \text{Dir}(\beta)$
end for
for each document $d \in D$ **do**
  Draw topic proportions $\theta_d|\alpha \sim \text{Dir}(\alpha)$
  for each word $w_{nd} \in d$ **do**
    Draw topic assignment $z_{nd}|\theta_d \sim \text{Mult}(\theta_d)$
    Draw word $w_{nd}|z_{nd}, \phi \sim \text{Mult}(\phi_{z_{nd}})$
  end for
end for
for each pair of document $d, d' \in D$, $d \neq d'$ **do**
  Draw binary link indicator $y_{dd'}|z_d, z'_d \sim \psi(\cdot|z_d, z'_d, \eta)$
end for

where K is the fixed set of topics and D is the set of documents. The random variable w_{nd} represents the n-th word of document d and z_{nd} represents the topic assignment of the n-th word of document d. The variable θ_d is the topic distribution of document d, sampled from a Dirichlet distribution with α prior, and $\phi_{z_{nd}}$ is the word distribution for the topic corresponding to the assignment z_{nd}, sampled from a Dirichlet with prior β. Finally, $y_{dd'}$ is the binary variable representing the link between documents d and d', drawn from a link likelihood function.

Figure 3.2 shows the graphical model of RTM for only a pair of documents, since it is difficult to represent the entire network of documents.

The link likelihood function can be defined in different ways; in this thesis, we consider the sigmoid function, parameterized by coef-

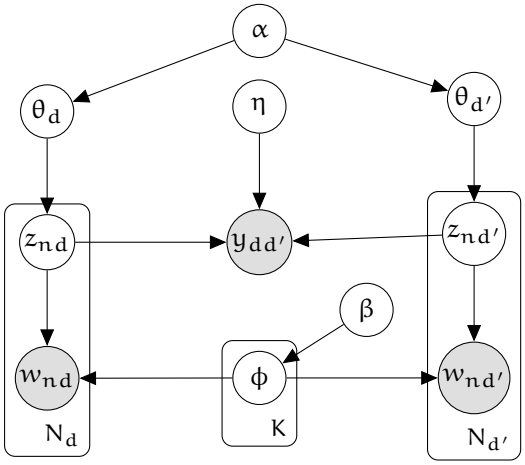


Figure 3.2: RTM in plate notation for a pair of documents. The observed variables are the words w_{nd}, like LDA, and links between documents, represented by $y_{dd'}$.

ficients η and intercept ν. The likelihood that a link y between two documents d and d' exists is then computed as:

$$\psi_\sigma(y = 1) = \sigma(\eta^T(\bar{\mathbf{z}}_d \circ \bar{\mathbf{z}}_{d'}) + \nu) \tag{19}$$

where σ is the sigmoid function, the symbol $\circ$ denotes the Hadamard product (or element-wise product) and $\bar{\mathbf{z}}_d$ is a vector, such that $\bar{\mathbf{z}}_d = \frac{1}{N_d}\sum_{n=1}^{N_d} z_{nd}$, where N_d is the length of document d and z_{nd} denotes the n-th word in document d.

Being an extension of LDA, the joint probability distribution of RTM is composed of the joint distribution of LDA and the term related to the links between documents. We mark the different modules of the model for the sake of clarity. The joint probability distribution is then as follows:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{y}, \theta, \mathbf{\Phi}|\alpha, \beta, \eta, \nu) \tag{20a}$$

$$= P(\boldsymbol{\phi}|\beta)P(\theta|\alpha)P(\mathbf{z}|\theta)P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi})\psi_\sigma(\mathbf{y}|\mathbf{z}, \mathbf{w}, \eta, \nu) \tag{20b}$$

$$= \underbrace{\prod_{k=1}^{K} p(\phi_k|\beta)}_{\text{topic plate}} \cdot \overbrace{\prod_{d=1}^{D} p(\theta_d|\alpha) \underbrace{\prod_{n=1}^{N_d} p(z_{nd}|\theta_d)p(w_{nd}|\phi_{z_{nd}})}_{\text{word plate}}}^{\text{document plate}} \tag{20c}$$

$$\cdot \underbrace{\prod_{\substack{d,d' \in D \\ d' \neq d}} \psi_\sigma(y_{dd'}|z_d, z_{d'}, \eta, \nu)}_{\text{link function}} \tag{20d}$$

Let us notice that the first four factors are the same as LDA's full joint distribution (equation 20c). In the original paper (Chang and Blei, 2009), the term related to the topic plate is missing. This is due to the fact that β does not represent the hyperparameter for the Dirichlet

ϕ, but it is treated as a $K \times V$–matrix parameter to infer. However, for maintaining a coherent notation throughout this thesis, we consider β as the prior of the word-topic distribution ϕ.

Equation 21 represents the collapsed Gibbs sampling equation for solving the problem of inference:

$$p(z_{nd} = t | w, \mathbf{z}^{\neg nd}, y, \alpha, \beta, \nu, \eta) \tag{21a}$$

$$\propto \left(N_{dt}^{\neg nd} + \alpha\right) \frac{N_{tw}^{\neg nd} + \beta}{N_{t\cdot} + W\beta} \tag{21b}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=1}} \sigma\left(\frac{\eta_t}{N_{d\cdot}} \cdot \frac{N_{d't}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}}\right) \tag{21c}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=0}} 1 - \sigma\left(\frac{\eta_t}{N_{d\cdot}} \cdot \frac{N_{d't}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}}\right) \tag{21d}$$

where

- the superscript $\neg nd$ indicates leaving the n-th word token of the d-th document out of the calculation;
- W represents the number of unique words in the vocabulary;
- N_{dz} denotes the number of words associated with the topic z in document d;
- $N_{z\cdot}$ denotes the number of words associated with the topic z in the corpus;
- N_{zw} denotes the number of occurrences of the word w associated with topic z;
- $N_{d\cdot}$ denotes the number of words in document d.

The first term (21b) corresponds to the Gibbs sampling equation for LDA, while the others represent the link probability function when the link is present (21c) and when the link is absent (21d). The last term (21d) is usually omitted, as the absence of a link between d and d' does not imply that there is evidence for $y_{dd'} = 0$. Therefore, absent links are treated as unobserved, also decreasing the computational cost of inference.

RTM can be associated with two different type of tasks, depending on the evidence the user is taking into account. If the words of documents are the evidence, the model can be used for a link prediction task, suggesting citations, hyperlinks, friendships or influencers, according to the specific context. Otherwise, using the links as evidence, RTM can infer keywords from citations or interests of a user from its social connections.

WEIGHTED STOCHASTIC BLOCK RELATIONAL TOPIC MODEL (WSB-RTM). Instead of considering only the links between documents, WSB-RTM (Yang et al., 2016a) assume that groups of strongly connected documents, called blocks, also share similar topics. In particular, the model embeds the Weighted Stochastic Block Model (Aicher et al., 2015, WSBM) to identify L blocks in which documents are densely connected.

WSBM assumes that a document belongs to exactly one block. A matrix $A_{l,l'}$ defines the weight of the link connecting two documents in blocks l and l'. This weight is generated from a Poisson distribution with parameter $\Omega_{l,l'}$ which has a Gamma prior with parameters a and b. WSB-RTM also puts a Dirichlet prior π on each block to capture the block's topic distribution and use it as an informative prior when drawing each document's topic distribution.

Finally, a link between a pair of documents is not only dependent on the topic assignments $\mathbf{z}$ (as in RTM), but also on the word lexical features $\mathbf{w}$, and on the inter-block link rates Ω. We then obtain the following link likelihood function of the link y between a pair of documents d and d':

$$\psi_\sigma(y = 1) = \sigma(\eta^T(\overline{\mathbf{z}}_d \circ \overline{\mathbf{z}}_{d'}) + \tau^T(\overline{\mathbf{w}}_d \circ \overline{\mathbf{w}}_{d'}) + \rho_{l,l'}\Omega_{l,l'}) \tag{22}$$

where w_d is a vector, such that $\overline{\mathbf{w}}_d = \frac{1}{N_d}\sum_n 1(w_{nd} = v)$, w_{nd} denotes the n-th word in document d, and η, τ and ρ are the weight vectors and matrix for topic-based, lexical-based and link rate-based predictions, respectively, and σ is the sigmoid function.

$$\begin{aligned}
&p(\mathbf{w},\mathbf{z},\mathbf{y},\theta,\mathbf{t},\mathbf{\Phi},A,\Omega,\mu,\pi|\alpha,\beta,\eta,\tau,\rho,\Omega,a,b,\gamma) \\
&= \underbrace{P(\pi|\alpha')P(\mu,|\gamma)P(\mathbf{t}|\gamma)}_{\text{Block plate}} \overbrace{P(\theta|\alpha,\mathbf{t},\pi)\underbrace{P(\mathbf{w}|\mathbf{z},\boldsymbol{\phi})P(\mathbf{z}|\theta)}_{\text{Word plate}}}^{\text{Document plate}} \cdot \underbrace{P(\boldsymbol{\phi}|\beta)}_{\text{Topic plate}} \\
&\quad \underbrace{\overbrace{P(A|\mathbf{t},\Omega)P(\Omega|a,b)}^{\text{Inter-blocks plate}}\psi_\sigma(\mathbf{y}|\mathbf{z},\mathbf{w},\eta,\tau,\rho,\Omega)}_{\text{Link probability function}} \\
&= \prod_l^L p(\pi_l|\alpha')p(\mu|\gamma)\prod_d^D p(t_d|\mu)p(\theta_d|\alpha,t_d,\pi_l)\prod_{n=1}^{N_d} p(w_{nd}|\phi_{z_{nd}})\cdot \\
&\quad p(z_{nd}|\theta_d)\prod_{k=1}^K p(\phi_k|\beta)\prod_{l,l'\in L} P(A_{l,l'}|t_d,\Omega_{l,l'})P(\Omega_{l,l'}|a,b)\cdot \\
&\quad \prod_{\substack{d,d'\in D \\ d'\neq d}} \psi_\sigma(y_{dd'}|z_d,z_{d'},w_d,w_{d'},\eta,\tau,\rho,\Omega_{l,l'})
\end{aligned} \tag{23}$$

where μ is the block distribution, controlled by the Dirichlet parameter γ.

3.2 NEURAL TOPIC MODELING

In recent years, neural topic models have gained increasing success and interest (Zhao et al., 2021), due to their flexibility and scalability. Several topic models use neural networks (Gupta et al., 2020; Larochelle and Lauly, 2012; Salakhutdinov and Hinton, 2009) or neural variational inference (Ding et al., 2018; Miao et al., 2017, 2016; Mnih and Gregor, 2014; Srivastava and Sutton, 2017).

Traditional approximate inference methods (e.g. mean-field and collapsed Gibbs) have the drawback that applying them to new topic models, even if there is a small change to the modeling assumptions, requires re-deriving the inference methods, which can be time consuming, and limits the ability of practitioners to freely explore the space of different modeling assumptions. AutoEncoding Variational Bayes (AEVB) (Kingma and Welling, 2014) seems to be a natural choice for topic models, because it trains a neural network that directly maps a document to an approximate posterior distribution, without the need to run further variational updates. Then, from this distribution, we can sample a lower-dimensional document representation. A decoder network (generative model) reconstructs the original input. We also call this architecture Variational AutoEncoder (VAE). In general, the input of these models is the BoW vector representation of the documents.

For additional details on neural topic models, we refer the readers to the work of Zhao et al. (2021). We now focus of the main neural topic modeling approaches of the state of the art.

NEURAL VARIATIONAL DOCUMENT MODEL (NVDM). The Neural Variational Document Model is composed of a Multi-Layer Perceptron (MLP) encoder (inference network) that compresses the input BoW document representation into a continuous latent distribution and a softmax decoder (generative model) reconstructs the document by generating the words indepedently. Each word is generated directly from the dense continuous lower-dimensional document representation, sampled from the learned distribution. Figure 3.3 sketches the architecture of NVDM.

More formally, the authors define a generative model with a latent variable h.[2] Let $d \in \mathbb{R}^V$ be the bag-of-words representation of a document (where V is the size of the vocabulary) and $d_i \in \mathbb{R}^V$ be the one-hot representation of the word at position i. An MLP encoder $q(h|d)$ compresses document representations into continuous hidden vectors ($d \longrightarrow h$). Then, a softmax decoder $p(d|h) = \prod_{i=1}^{N} p(d_i|h)$ recon-

2 This variable corresponds to the topical document representation in LDA, which is usually referred as the document-topic distribution θ. So the values of θ are constrained to be non-negative and they need to sum up to 1. On the other hand, h is unconstrained and continuous. We will therefore refer to θ and h as to different elements.

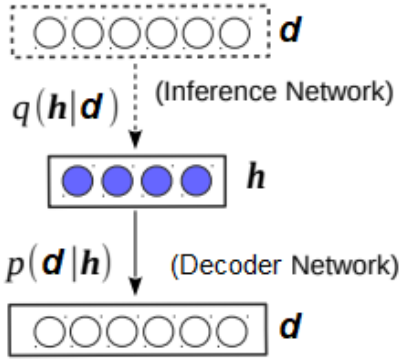


Figure 3.3: High-level schema of Neural Variational Document Model.

structs the documents by independently generating the words ($h \longrightarrow \{d_i\}$). τ parameterizes the generative distribution $p_\tau(d|h)$, while ν are the inference network parameters.

NVDM and LDA share the same generative process, with the exception that NVDM requires a Gaussian prior over the document-topic representation of the documents instead of a Dirichlet prior for computational reasons. The variational lower bound $\mathcal{L}$ is derived as:

$$\mathcal{L} = \mathbf{E}_{q_\nu(h|d)}\Big[\sum_{j=1}^{N_d} \log p_\tau(d_i|h)\Big] - D_{KL}[q_\nu(h|d)||p(h)] \tag{24}$$

where N_d is the number of words in the document and $p(h)$ is a Gaussian prior for h. During training, the model parameters τ together with the inference network parameters ν are updated by stochastic back-propagation based on the samples h drawn from $q_\nu(h|d)$. For the gradients with respect to ν, the authors reparameterize $h = \mu + \sigma \cdot \epsilon$ and sample $\epsilon \sim \mathcal{N}(0, I)$ (Kingma and Welling, 2014). And then the update of ν can be carried out by back-propagating the gradients w.r.t. μ and σ. Based on the samples $h \sim q_\phi(h|d)$, the lower bound $\mathcal{L}$ can be optimised by back-propagating the stochastic gradients w.r.t. θ and ϕ. Since $p(h)$ is a standard Gaussian prior, the Gaussian KL-Divergence $D_{KL}[q_\phi(h|d)||p(h)]$ can be computed analytically to further lower the variance of the gradients.

The conditional probability over words $p_\theta(d_i|h)$ (i.e. the decoder network) is modeled by multinomial logistic regression:

$$p_\theta(d_i|h) = \frac{\exp\{-E(d_i; h, \tau)\}}{\sum_{j=1}^{|V|} \exp\{-E(d_j; h, \tau)\}} \tag{25}$$

where

$$E(d_i; h, \tau) = -h^T R d_i - b_{d_i} \tag{26}$$

where $R \in \mathbb{R}^{K \times |V|}$ represents the topics representations and b_{d_i} represents the bias term. Let us notice that we can extract the topical

words from the matrix R: the words of the vocabulary with the highest weights R_k represents the most significant words of the topic k.

NVDM stands at the basis of different extensions of neural topic models. In the following we will focus on two of the most prominent NVDM extensions.

EMBEDDED TOPIC MODEL (ETM). The Embedded Topic Models (Dieng et al., 2020) aims to combine the benefits of LDA and word embeddings. It represents words and topics in the same embedding space. In particular, it embeds the vocabulary in an L-dimensional space (thus obtaining classical word embeddings). But also a topic k is a vector $\gamma_k \in \mathbb{R}^L$. Therefore γ_k is a topic embedding, i.e. a distributed representation of the k-th topic in the semantic space of words.

In its generative process, the ETM uses the topic embedding to form a topic distribution over the vocabulary. Specifically, ETM uses a loglinear model that takes the inner product of the word embedding matrix and the topic embedding. With this form, the ETM assigns high probability to a word v in topic k by measuring the agreement between the word embedding and the topic embedding. More formally, let ρ the $L \times |V|$ word embedding matrix. In the generative process of ETM, a word w_{nd} is sampled according to $\text{softmax}(\rho^T \gamma_{z_{nd}})$, where z_{nd} is the topic assignment sampled from the document-topic distribution of document d.

Another difference between ETM and NVDM is that the NVDM uses a document real-valued latent vector, instead of a probability latent vector (as LDA). On the contrary, ETM constrains the latent variable h to lie in the simplex (its values are non-negative and sum up to 1).

In addition, ETM can automatically learn the word embedding representations or use pre-trained word embeddings. The use of pre-trained word embeddings allows the model to add general information and improve the coherence of the topics over ETM with learned embeddings.

PRODUCT OF EXPERTS LDA (PRODLDA). ProdLDA addresses two main issues in NVDM. The first challenge is related to the prior over the latent distribution of the document. NVDM uses a Gaussian distribution because it can be reparameterized, as we have seen before. To truly translate LDA into a neural topic model, we should assume a Dirichlet prior over the document. Yet the Dirichlet prior is not a location scale family, and that hinders reparameterization. To address this problem, ProdLDA explicitly approximates the Dirichlet prior using Gaussian distributions. In other words, the authors use an encoder network that approximates the Dirichlet prior $p(\theta|\alpha)$ with a logistic-normal distribution (more precisely, this is softmax-normal distribution).

Another well-known problem of NVDM is the phenomenon of component collapse, in which the encoder network becomes stuck in a bad local optimum in which all topics are identical. To address this issue, the authors used the Adam optimizer, batch normalization and dropout units in the encoder network.

Moreover, LDA and ETM models the distribution $p(\mathbf{w}|\theta, \phi)$ as a mixture of multinomials. (Srivastava and Sutton, 2017) note that this assumption leads to predictions that are never sharper than the components (the topics) that are being mixed. This can result in some topics appearing that are poor quality and do not correspond well with human judgment. To address this issue, the authors replace the word probabilities with a weighted product of experts (Hinton, 2002) which is capable of making sharper predictions than any of the constituents experts by definition. More formally, the word-topic distribution ϕ of LDA becomes an unnormalized weight matrix and therefore the conditional distribution of w_n is defined as

$$p(w_n|\phi, \theta) = \text{Categorical}(\sigma(\phi\theta)) \tag{27}$$

where σ is a sigmoid function. This modification allows the topic model to obtain a drastic improvement in topic coherence.

3.3 MULTILINGUAL TOPIC MODELING

Multilingual topic models are a subset of topic models that aim at finding aligned topics in bilingual or multilingual corpora. Using LDA or a classical topic model to extract topics from a bilingual corpus will result in monolingual topics that do not explicit the relationship among words coming from different languages. Additional information is then required in order to align the topics or the documents. Two main directions address the problem of multilingual topic modeling: at the document level and at the word level.

MULTILINGUAL MODELS AT THE DOCUMENT LEVEL The first strategy is to extract topics from parallel or highly comparable multiligual corpora, under the assumption that translations (or comparable documents) share the same topic distributions. Polylingual Topic Model (PLTM)(Mimno et al., 2009) is the most well-known example and has extensively used and adapted in various ways for different cross-lingual tasks. Models that transfer knowledge on the document level have many variants(Hao and Paul, 2018; Heyman et al., 2016; Krstovski et al., 2016; Liu et al., 2015). PLTM assumes that the tuples of parallel (or closely comparable) documents share the same topic distribution. Although, there is a specific word-topic distribution for each language.

MULTILINGUAL MODELS AT THE WORD LEVEL Another approach consists in modeling the connection between languages through words using multilingual resources (such as dictionaries) (Boyd-Graber and Blei, 2009; Jagarlamudi and Daumé, 2010; Wu et al., 2020; Zhao and Xing, 2006). The use of dictionaries to model similarities across topic-word distributions has been formulated in different ways as well. ProbBiLDA (Ma and Nasukawa, 2017) uses inverted indexing to encode assumptions that word translations are generated from same distributions. (Gutiérrez et al., 2016) use part-of-speech taggers to separate topic words (nouns) and perspective words (adjectives and verbs), developed for the application of detecting cultural differences, such as how different languages have different perspectives on the same topic.

3.4 EVALUATING A TOPIC MODEL

Evaluation of the results of a topic model is an important issue, due to the unsupervised nature of the models. Many studies propose and explore methods to assess a model from a qualitative and quantitative perspective.

3.4.1 *Quantitative evaluation*

Human Evaluation

The simplest way to evaluate if the words of a topics are semantically coherent is to make a group of human evaluators **rate each topic on a 3-point scale** (Doogan and Buntine, 2021; Newman et al., 2010). Alternatively, evaluators can rate the quality of the topics by performing the following two tasks (Chang et al., 2009):

- **Word Intrusion** measures the cohesion of inferred topics. A word is selected at random from a set of words with low probability in a given topic. Human subjects must identify the extraneous word inserted into the topic, i.e. the less semantically coherent word in the topic.
- **Topic Intrusion** measures how well the topic model has decomposed a document as a mixture of topics. Human subjects evaluate the title, a brief snippet from a document and four topics. Three of the topics are the highest probability topics assigned to the document, while the remaining is selected at random from a set of low probability topics for the document. The evaluators must identify the less related topic for the given document.

Automatic Methods

HELD-OUT LOG-LIKELIHOOD AND PERPLEXITY Evaluation of the probability of held-out documents (Wallach et al., 2009) is a method based on likelihood estimation. A given dataset is split between a training set and a testing set. **Log-likelihood** is evaluated given the model trained with the documents belonging to the testing set:

$$\mathcal{L}(\mathbf{w}) = \log p(\mathbf{w}|\boldsymbol{\phi}, \alpha) = \sum_{d}^{D} \log p(\mathbf{w}_d|\boldsymbol{\phi}, \alpha) \tag{28}$$

The parameter θ is omitted because it represents the topic distribution for the documents of the training set. The best model will be the one which gives the highest probability. Another way of assessing a topic model is to split each document of the dataset into two halves and then estimate the log-likelihood of the second half of a document, given the first half. In both cases, computing log-likelihood is intractable; (Wallach et al., 2009) and (Buntine, 2009) explore methods for estimating it efficiently.

Perplexity is an alternative metric derived from log-likelihood and defined as:

$$\text{ppx}(\mathbf{w}) = \exp\left\{-\frac{\mathcal{L}(\mathbf{w})}{\text{number of words}}\right\} \tag{29}$$

Low values of perplexity are preferred here. However, Chang and Blei (2009) show that a model with low held-out perplexity (and consequently high held-out log-likelihood) does not imply more semantically coherent topics. Therefore, perplexity and log-likelihood are not often used nowadays.

TOPIC COHERENCE Topic coherence is a measure for estimating the quality of the inferred topics. It evaluates how related are the most likely words composing a topic, usually the top-10 words. A wide variety of measures have been proposed across the years. Topic coherence metrics are usually based on word co-occurrences. These can be computed on the original dataset ("internal topic coherence") or on an another dataset ("external topic coherence"), e.g. Wikipedia, (Newman et al., 2010; Röder et al., 2015).

Traditional topic coherence measures include the following:

- **Pointwise Mutual Information (PMI)** is a measure commonly used in information theory and statistics. In the field of topic modeling, it is also known with the name of UCI coherence (Newman et al., 2010). PMI is computed for each pair of words w_i and w_j in a topic:

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{30}$$

A pointwise mutual information score is defined for all pairs of N most probable words in the topic t:

$$\text{PMI–score}(t) = \frac{1}{\binom{N}{2}} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \text{PMI}(w_i, w_j) \tag{31}$$

Finally, the results are averaged over all the topics. A model which outputs topics whose PMI-scores are closer to 0 are to be preferred.

- **Normalized Pointwise Mutual Information (NPMI)** is the normalized version of PMI (Aletras and Stevenson, 2013). NPMI for a pair of words in a topic is as follows:

$$\text{NPMI}(w_i, w_j) = \frac{\text{PMI}(w_i, w_j)}{-\log p(w_i, w_j)} \tag{32}$$

Also for this metric, the higher the better. NPMI seems to be better correlated with the human judgment than PMI and it is the most used topic coherence metric.

Aletras and Stevenson (2013) introduced distributional semantic similarity methods for computing coherence, calculating the distributional similarity between vectors for the top-N words of topics using a range of different similarity measures (e.g. cosine). To construct the vector space, they used Wikipedia English as reference corpus and a window of ± 5 words. (Röder et al., 2015) further studied distributional semantic similarity methods proposing different measures and by varying the hyperparameters (e.g. window size) of the measures. They correlated all the metrics with the human judgment and found out that C_V score seems to be more consistent with human judgment compared to other widely used metrics such as PMI. For the sake of completeness, we report this measure, but recent work and the author of the original paper [3] reported that this measure seems to be negatively correlated to other measures (Doogan and Buntine, 2021).

Although NPMI seems the measure to prefer, it is computationally expensive, especially if the co-occurrence probabilities are computed on a large corpus (e.g. Wikipedia). To this end, also measures based on pre-trained word embeddings can be considered. We can consider this **word embedding-based measure** as an external topic coherence, but it is more efficient to compute than Normalized Pointwise Mutual Information on an external corpus. For example, (Belford and Greene, 2019) compute the average pairwise cosine similarity of the word embeddings of the top-10 words in a topic using different pre-trained embeddding spaces. Let be $c(w_i)$ and $c(w_j)$ the word embeddings of the words w_i and w_j respectively (where c is the function that maps the words to their corresponding word embedding), then the word

3 https://github.com/dice-group/Palmetto/issues/12

embedding topic coherence for a topic of N words is computed as follows:

$$\text{PMI–score}(t) = \frac{1}{\binom{N}{2}} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \text{PMI}(c(w_i), c(w_j)) \tag{33}$$

TOPIC DIVERSITY AND SIMILARITY METRICS Topic diversity and topic similarity are the two sides of the same coin. We usually expect the topics to be well-separated from the others (topic diversity); but sometimes we might also be interested in finding out topics which are similar to a given topic (topic similarity). Indeed, we can often convert a topic similarity measure in a topic diversity measure, and vice versa.

Most of the topic similarity measures are based on word tokens and usually adopt a list of top-N terms to estimate if two topics are similar. In this category, we mention the following measures:

- **Average Jaccard Similarity (JS)**. The ratio of common words in two topics can be measured by using Jaccard Similarity (Deng et al., 2012; Tran et al., 2013). The Jaccard Similarity (JS) between two topics t_i and t_j, where each topic is a list of N words $\{w_1, \ldots, w_N\}$, is defined as follows:

  $$\text{JS}(t_i, t_j) = \frac{|t_i \cap t_j|}{|t_i \cup t_j|} \tag{34}$$

  This measure varies between 0 and 1, where 0 means that the topics are completely different, and 1 means that topics are similar to each other.

- **Proportion of Unique Words (PUW)**. Topic diversity can computed as the proportion of unique words in the top-N words of all topics (Dieng et al., 2020). Let K be the number of topics and t_i be the i-th topic descriptor composed of N words. The PUW metric is defined as follows:

  $$\text{PUW} = \frac{\bigcup_{i=1}^{K} |t_i|}{K \cdot N} \tag{35}$$

  This metric ranges between $1/K$ (when all the words are replicated in all the topics) and 1 (when all the words composing the topics are unique), and can be easily converted into a topic similarity measure by computing $1 - \text{PUW}$.

- **Average Pairwise Pointwise Mutual Information (PMI)**. Aletras and Stevenson (2014) present a similarity metric based on Pointwise Mutual Information (PMI). The authors adapt the PMI coherence to measure topic similarity by computing the average pairwise PMI between the words belonging to two topics.

More formally, the PMI between the topics t_i and t_j is defined as:

$$\text{PMI}(t_i, t_j) = \frac{1}{N^2} \sum_{u \in t_i} \sum_{v \in t_j} \text{PMI}(u, v) \tag{36}$$

where N is the number of words of each topic.

- **Rank-biased Overlap (RBO).** To consider the ranking of the words, one can use Rank-Biased Overlap (RBO) (Webber et al., 2010). It is based on a probabilistic model in which a user compares the overlap of two ranked lists (that in our case correspond to two topics) at incrementally increasing depth. The user can stop to examine the lists at a given rank position according to the probability p, enabling therefore the metric to be top-weighted and consequently giving more weight to the top words of a topic. The smaller p, the more top-weighted the metric is. When $p = 0$, only the top-ranked word is considered. The metric ranges from 0 (completely different topic descriptors) to 1 (equal topic descriptors).

  RBO is based on the concept of *overlap at depth* h between two lists, which is the number of elements that the lists share when only the first h words are considered. For example, the overlap at depth 2 between the lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$ is 1. The average overlap is defined as the proportion of the overlap at depth h over h. Therefore, the RBO measure when evaluating two topics is computed as the expected value of the average overlap that the user observes when comparing two lists.

Other topic similarity/diversity approaches are instead based on the probability distribution of the words denoting the topics, i.e. the word-topic distribution usually referred as ϕ. These metrics may be sensitive to the high dimensionality of the vocabulary (Aletras and Stevenson, 2014).

- **Average Log Odds Ratio (LOR)**. Topic similarity can be computed using the average log odds ratio (LOR) (Chaney and Blei, 2012), which is defined as follows:

$$\text{LOR}(\phi_i, \phi_j) = \sum_{w \in V} \mathbb{1}_{\mathbb{R}_{\neq 0}}(\phi_{iw}) \mathbb{1}_{\mathbb{R}_{\neq 0}}(\phi_{jw}) |\log(\phi_{iw} - \beta_{jw}| \tag{37}$$

  where $\mathbb{1}_A(x)$ is an indicator function defined as 1 if $x \in A$ and 0 otherwise. This metric computes the distance between the distributions associated with two topics, so it is a dissimilarity metric.

- **Kullback-Leibler Divergence (KL-DIV).** A widely used measure to determine the similarity between two topics is the Kullback-Leibler Divergence (AlSumait et al., 2009; Sievert and Shirley,

2014), which measures the distance from a given topic's distribution ϕ_k over words to another one. It is defined as follows:

$$KL - DIV(\phi_i, \phi_j) = \sum_{w \in V} \phi_{iw} \log \frac{\phi_{iw}}{\phi_{jw}} \tag{38}$$

Notice that this metric is not symmetric and its domain ranges from 0 (when two distributions are identical) to infinity. In fact, this metric represents a dissimilarity score. Other metrics based on computing the distance between distributions include the Jensen Shannon Divergence and the cosine similarity (Aletras and Stevenson, 2014).

These distribution-based measures suffer from the high dimensionality of the vocabulary, generating solutions that do not strongly correlate with human judgment (Aletras and Stevenson, 2014).

TOPIC SIGNIFICANCE Sometimes a topic model can identify topics which are more significant or relevant than others. To this end, we can use topic significance measures to rank the topics (AlSumait et al., 2009). These measures focus on the distributions of the topics produced by a model. They compute the distance between the discovered topics and three different definitions of "junk topics" in terms of Kullback-Leibler divergence. Topic models that obtain higher values on average are to be preferred..

- **KL-Uniform (KL–U)** measures the distance of each topic-word distribution from the uniform distribution over the words. Significant topics are supposed to be skewed towards a few coherent and related words and distant from the uniform distribution.
- **KL-Vacuous (KL–V)** measures the distance between each topic-word distribution and the empirical word distribution of the whole dataset, also called "vacuous" distribution. The closer the word-topic distribution is to the vacuous distribution of the sample, the lower is its significance.
- **KL-Background (KL–B)** measures the distance of a topic k to a "background" topic, which is a generic topic that is found equally probable in all the documents. Meaningful topics appear in a small subset of the data, thus higher values of KL–B are preferred.

DOCUMENT CLASSIFICATION METRICS. A topic model can be also evaluated on downstream tasks. For example, we can consider the document-topic representations produced by the model and use them as features to train a classifier on an annotated dataset. In this

case, we can use the traditional classification measures. Here we report the most common ones.

Let us consider a multi-class problem with C classes. Let be tp, tn, fp, and fn the number of true positives, true negatives, false positives and false negatives respectively. Then **precision** for a given class i is defined as follows:

$$\textsc{Precision}(i) = \frac{tp}{tp + fp} \tag{39}$$

and **recall** is defined as follows:

$$\textsc{Recall}(i) = \frac{tp}{tp + fn} \tag{40}$$

F-measure, or F1 score, for a given class i is the weighted average of the precision and recall, and it reaches its best value at 1 and its worst score at 0. F-measure of class i is then defined as:

$$\textsc{f-measure}(i) = \frac{2 \cdot \textsc{Recall}(i) \cdot \textsc{Precision}(i)}{\textsc{Recall}(i) + \textsc{Precision}(i)}$$

Given a multi-class problem, we can aggregate the F1 scores with different strategies:

- **Macro-F1**. The average of the F1 score for each class is usually referred to as Macro-F1 or Macro-average F1 score. It is then defined as follows:

  $$\textsc{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} \textsc{f-measure}(i)$$

  where $Classes$ denotes the set of the classes.

- **Micro-F1**. The weighted average of the F1 score for each class (where the weight corresponds to the size of the classes) is called Micro-F1 or Micro-average F1 score, and it is then defined as:

  $$\textsc{Micro-F1} = \frac{1}{D} \sum_{i=1}^{C} |i| \cdot \textsc{f-measure}(i)$$

  where D is the number of instances in the test set and $|i|$ the cardinality of class i.

3.4.2 *Qualitative evaluation*

The most straightforward way to qualitatively evaluate the results of a topic model is to visualize the topics that the model have produced. In particular, we can observe the top-N most likely words of a given topic. Different works provide methods for a user-friendly visualization of the topics, which also take into account the weight of each

word of the vocabulary in the given topic (Chuang et al., 2012; Murdock and Allen, 2015; Sievert and Shirley, 2014).

Some topic models can infer relationships between subsequent words or phrases. This improves the interpretability of topics because a topic is characterized by n-grams instead of single word tokens. We can also enhance the human interpretability of a model that considers topics as unigram distributions over words as well, while preserving the advantages of a bag-of-words formulation. A strategy consists in finding significant phrases related to a topic using multi-word expression discovery techniques (Blei and Lafferty, 2009; Manning and Schütze, 2001).

Otherwise, it is possible to enhance the interpretability of a topic by automatically associating a label to the given topic. The simplest method selects the most likely term in the word distribution ϕ_k of the topic k. Several other approaches select the best candidate label for a topic, using supervised rankers (Lau et al., 2011), approaches based on word embeddings (Bhatia et al., 2016; Kou et al., 2015) or on transformer-based pre-trained models (Popa and Rebedea, 2021).

3.5 HYPERPARAMETER SELECTION IN TOPIC MODELS

Concerning the problem of setting hyperparameters for topic models, researchers have adopted different strategies. They usually select a priori fixed values according to some domain knowledge. For example, in the case of LDA, several approaches (Bao et al., 2009; Daud et al., 2009; Mukherjee and Liu, 2012) fix the values of the hyperparameters α and β according to the work of Griffiths and Steyvers (2004).

However, it has been shown that the same values do not apply to every dataset (Wallach, 2008). Moreover, in most cases, there is no prior knowledge of the distribution of the topics over the corpus and this makes the choice of the hyperparameter configuration difficult. Therefore, researchers usually select the best configuration of the hyperparameters using grid search techniques (Griffiths and Steyvers, 2004; Harrando et al., 2021; Pavlinek and Podgorelec, 2017). These approaches are easy to implement, parallelizable, and accurate in low dimensional spaces, but they suffer from the curse of dimensionality, as the number of the possible configurations grows exponentially with the number of hyperparameters (Bergstra and Bengio, 2012).

Another option is to adopt fixed-point methods for estimating the hyperparameters of a topic model (Asuncion et al., 2009; Wallach, 2008). The inference algorithm alternates between sampling latent topics and inferring model hyperparameters. However, not every type of hyperparameter can be estimated with these methods. With the advent of neural topic modeling, other types of hyperparameters need

to be considered. These are mainly related to the network architecture.

Bayesian Optimization techniques (Archetti and Candelieri, 2019) can be superior to point estimates and grid search techniques (Snoek et al., 2012), and it is designed for expensive objective functions. Yet, a thorough investigation of BO methods in topic modeling is still missing. We refer the reader to Section 2.5.3 for a detailed description of Bayesian Optimization.

3.6 SUMMARY OF THE CONTRIBUTIONS OF THIS THESIS

Contributions of this research work are spread among different levels:

- a methodology for incorporating different types of relational information based on Relationa Topic Models (RTM) (Chang and Blei, 2009) and Constrained LDA (CLDA) (Yang et al., 2015c), which is modular and easy to apply to classical probabilistic topic models (Chapter 4);
- the definition of a class of neural topic models that overcome the BoW limitations of the current models to improve the quality of the topics and address cross-lingual zero-shot prediction tasks (Chapter 5);
- a comprehensive framework for a fair comparison between topic models, based on hyperparameter optimization, which also allows us to investigate the different elements that play a role in the evaluation of the models (Chapter 6);
- a method for an efficient evaluation between topic models based on the transfer of the hyperparameters from a dataset to an unseen one (Chapter 7).

Part II

MODELING OF ADDITIONAL INFORMATION INTO TOPIC MODELS

4

MODELING RELATIONAL INFORMATION INTO CLASSICAL TOPIC MODELS

Most topic models consider the text as a unique source of information. For example, Latent Dirichlet Allocation, which is still one of the most used topic models, assumes that the words in the documents are the only evidence. However, we may know that two documents are written by the same author, or they are associated with the same label. Indeed, in some practical cases, **we have additional information that can be incorporated into the model**. We therefore consider a first distinction between two types of information general and domain-specific.

The first category of information includes taxonomies, vocabularies, knowledge graphs, word embeddings. General information can be collected from publicly available resources. This information can be used for encouraging (or discouraging) two words or documents to share (or not share) the same topics. For instance, the words “search” and “engine” are more likely to be in the same topic than the words “search" and “make-up”. Indeed, if two words are semantically related, they are more likely to share the same topic.

Regarding the second category of additional information, i.e. domain-specific information, rather than just the words of the documents, we may have metadata associated with documents. For example, two related books, because written by the same author, are more likely to share the same topics. More generally, knowing that two documents are related increases our confidence that the documents talk about similar topics. Indeed, we just described a particular type of information, which can be either general or domain-specific, that we will call **relational information. Words or documents often show a relational structure in real-world cases**. For example, two scientific papers may be related by a citation, or two web pages may be related by a hyperlink. We can also use this type of information to improve the quality of the topic models’ results.

In this chapter, we will focus on two types of relational information, namely relationships between documents and between words. Regarding the relationships among documents, we have already mentioned citation networks and hyperlinked web pages, but we may also have co-authorship networks or social networks of friends (where each person is identified by the documents they have written). On the other hand, the relationships among words may include semantic relationships (e.g. synonyms, hyperonyms), syntactic relationships (e.g., word-order and syntax trees), or we may also have domain-specific

information when a domain expert knows that two words must be in the same topic. In general, knowing that two elements (words or documents) are related implies that the related elements are more likely to share the same set of topics. We will therefore consider two categories of topic models: topic models that encode relational information at the document level (*Document-Level Relational Topic Models*) and models that encode relationships at the word level (*Word-level Relational Topic Models*).

Researchers have proposed many topic models that include relational information across the years (Chen et al., 2013a; Guo et al., 2015; Yang et al., 2015a, 2016b; Zhang et al., 2013; Zhu et al., 2013). However, they usually include only one type of relational information, i.e. the links between documents or the relationships between words, disregarding that documents can also provide some other prior information. We will therefore investigate a methodology to incorporate document-level and word-level relationships into classical probabilistic topic models. In addition to this, we will investigate the impact of modeling these different types of relationships into topic models.

RESEARCH QUESTIONS. In this chapter, we will therefore address the following research questions:

Q4.1 How can we incorporate additional document-level and word-level relational information into classical topic models?

Q4.2 What is the impact of modeling document-level and word-level relational information into topic models?

The proposed topic models extend the well-known Document-level Relational Topic Model, i. e., RTM (Chang and Blei, 2009). For a review on RTM, we refer the reader to Section 3.1.2. The next sections are organized as follows. We will define the class of the Constrained Relational Topic Models (CRTM), an extension of RTM, and how to incorporate additional information into RTM (Section 4.1). We will show different variants of Constrained Relational Topic Models, one modeling the relational information among documents and the other modeling relational information among words and named-entities, i. e., *Document Constrained Relational Topic Models* (D-CRTM, Section 4.2) and *Entity Constrained Relational Topic Models* (E-CRTM, Section 4.3) respectively. Since we propose different extensions of RTM, the proposed class of topic models belongs to the family of Document-Level Relational Topic Models. However, our modeling is modular and can be applied to other classical topic models. Indeed, we will show in Section 4.3 that our method for incorporating additional information can be easily applied both to LDA and to RTM.

4.1 MODELING ADDITIONAL INFORMATION INTO DOCUMENT-LEVEL RELATIONAL TOPIC MODELS

Most of the Document-level Relational Topic Models consider only the document network information, disregarding that other types of information deriving from domain-specific information can be encoded as well. Building upon RTM, we introduce the information at the document level in the form of constraints through the definition of a set of potential functions, inspired by Constrained LDA (Yang et al., 2015c, CLDA). For details on CLDA, we refer the reader to Section 3.1.1.

The information to model is denoted by a set L. Each element $l \in L$ of the information set is introduced into the model by a potential function $f_l(z, d)$, representing a real-valued score for the hidden topic assignment z in document d. The overall information L defines a score $\xi(\mathbf{z}, L) = \prod_{z \in \mathbf{z}} \exp f_l(z, d)$ that smooths the current topic assignment $\mathbf{z}$. The joint probability distribution of this class of topic models, to which we will refer as Constrained Relational Topic Models (CRTM), is defined as follows:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{y}, \theta, \boldsymbol{\phi} | \alpha, \beta, \eta, \nu, L) \tag{41a}$$

$$= \underbrace{P(\boldsymbol{\phi}|\beta)}_{\text{Topic plate}} \underbrace{P(\theta|\alpha) P(\mathbf{z}|\theta) P(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi})}_{\text{Document plate}} \underbrace{\psi_\sigma(\mathbf{y}|\mathbf{z}, \mathbf{w}, \eta, \nu)}_{\text{Link function}} \underbrace{\xi(\mathbf{z}, L)}_{\substack{\text{Potential} \\ \text{function}}} \tag{41b}$$

The potential function ξ and the link probability function ψ_σ can be factored out of the marginalized joint distribution, because they do not depend on the distributions ϕ and θ, obtaining the following marginalized joint probability distribution:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{y} | \alpha, \beta, \eta, \nu, L) \tag{42a}$$

$$= \int \int p(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\beta) p(\mathbf{z}|\theta) p(\theta|\alpha) \psi_\sigma(\mathbf{y}|\mathbf{z}, \mathbf{w}, \eta, \nu) \xi(\mathbf{z}, L) d\theta d\phi \tag{42b}$$

$$= \xi(\mathbf{z}, L) \psi_\sigma(\mathbf{y}|\mathbf{z}, \mathbf{w}, \eta, \nu) \int \int p(\mathbf{w}|\mathbf{z}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\beta) p(\mathbf{z}|\theta) p(\theta|\alpha) d\theta d\phi \tag{42c}$$

The main goal of CRTM is to estimate the posterior distribution $P(\mathbf{z}|\mathbf{w}, \mathbf{y}) = P(\mathbf{w}, \mathbf{z}, \mathbf{y}) / \sum_{\mathbf{z}} P(\mathbf{w}, \mathbf{z}, \mathbf{y})$. Since the evaluation of the denominator is intractable, we need to use an approximate inference

method for the inference. In particular, we use a collapsed Gibbs sampler that leads to the following estimation:

$$P(z_{nd}|w_{nd}, \mathbf{z}^{\neg nd}, \mathbf{y}, \alpha, \beta, \eta, \nu, L) \tag{43a}$$

$$= \frac{P(\mathbf{w}, z_{nd}, \mathbf{z}^{\neg nd}, \mathbf{y}|\alpha, \beta, \eta, \nu, L)}{P(\mathbf{w}, \mathbf{z}^{\neg nd}, \mathbf{y}|\alpha, \beta, \eta, \nu, L)} \tag{43b}$$

$$= \frac{P(\mathbf{w}, z_{nd}, \mathbf{z}^{\neg nd}|\alpha, \beta)}{P(\mathbf{w}, \mathbf{z}^{\neg nd}|\alpha, \beta)} \prod_{\substack{d' \neq d \\ y_{dd'}=1}} \frac{\psi_\sigma(y_{dd'} = 1|z_{nd}, \mathbf{z}_d^{\neg nd}, \mathbf{z}_{d'}, \eta, \nu)}{\psi_\sigma(y_{dd'} = 1|\mathbf{z}_d^{\neg nd}, \mathbf{z}_{d'}, \eta, \nu)} \tag{43c}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=0}} \frac{\psi_\sigma(y_{dd'} = 0|z_{nd}, \mathbf{z}_d^{\neg nd}, \mathbf{z}_{d'}, \eta, \nu)}{\psi_\sigma(y_{dd'} = 0|z_d^{\neg nd}, \mathbf{z}_{d'}, \eta, \nu)} \cdot \frac{\xi(\mathbf{z}^{\neg nd}, \mathbf{z}_{nd}, L)}{\xi(\mathbf{z}^{\neg nd}, L)} \tag{43d}$$

$$\propto \left(N_{dz_{nd}}^{\neg nd} + \alpha\right) \frac{N_{z_{nd}w}^{\neg nd} + \beta}{N_{z_{nd}\cdot} + W\beta} \tag{43e}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=1}} \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{43f}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=0}} 1 - \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{43g}$$

$$\cdot \exp\left(f_l(z_{nd}, d)\right) \tag{43h}$$

where

- the superscript $\neg nd$ indicates leaving the n-th token of the d-th document out of the calculation;
- W represents the number of unique words in the vocabulary;
- N_{dz} denotes the number of words associated with the topic z in document d;
- $N_{z\cdot}$ denotes the number of words associated with the topic z in the corpus;
- N_{zw} denotes the number of occurrences of the word w associated with topic z;
- $N_{d\cdot}$ denotes the number of words in document d.

Equation (43e) corresponds to the Gibbs sampling of the standard LDA (Griffiths and Steyvers, 2004), equation (43f) represents the sigmoid link likelihood function of RTM when a link exists, and equation (43g) denotes the link function of RTM when a link is absent,

plus the incorporation of additional information by means of the potential function. Let us notice that equations (43f) and (43g) related to RTM deal with directed graphs, however it can be easily adapted to deal with undirected networks, similarly to all the models that extend RTM.

In the following sections, we will describe how to represent the additional information set L and how to define potential functions to incorporate document-level or word-level relationships in the form of constraints.

4.2 MODELING DOCUMENT-LEVEL RELATIONAL INFORMATION INTO DOCUMENT-LEVEL RELATIONAL TOPIC MODELS

We will now focus on how to incorporate additional information as document constraints. The introduction of document constraints instead of document labels can be more realistic in some cases. For instance, labels may be unknown, but a user may know whether two documents belong or do not belong to the same class (Basu et al., 2004). This formulation is also more general, as document constraints imply labels, but vice versa does not hold. Therefore, we will propose the class of models *Document Constrained Relational Topic Models* (D-CRTM), which make use of potential functions, inspired by must-link and cannot-link constraints described in (Andrzejewski et al., 2009), that allow us to incorporate document constraints in RTM.

4.2.1 *Definition of Information Sets*

We define two information sets for each document d: a *must-constraint* set L_d^m, containing documents that must share the same topics of d, and a *cannot-constraint* set L_d^c, including documents that cannot share the same themes of d. For example, a must-constraint set for the book titled *Emma* and written by Jane Austen could be

$$L^m_{Emma} = \{\textit{Sense and Sensibility}, \textit{Pride and Prejudice}\}$$

which contains a set of books written by the same author. Analogously, a cannot-constraint set could be

$$L^c_{Emma} = \{\textit{Divine Comedy}\}$$

which denotes a book that has not been written by Jane Austen.

In the following, we will propose two potential functions, which, once instantiated in D-CRTM, will lead to the Unnormalized and Normalized D-CRTM.

4.2.2 *Unnormalized Document Potential Function*

We can encode document relationships by modeling the relationship that exists between the words of two constrained documents. In particular, we assume that if two documents are must-constrained (i.e. they must share the same set of topic assignments), then the words in the documents must have similar topic distributions, i.e. $p(z_d|w, d) \approx p(z_{d'}|w', d')$, where w are the words of document d, and w' are the words of document d'. In other words, we model the idea that the more the words of the documents belonging to the set L_d^m are assigned to topic t, the higher the value of the potential function $f_l(z = t, d)$ is. Analogously, a cannot-constraint between two documents indicates that their words should not share the same set of topics. Therefore, if many words of two cannot-constrained documents are assigned to the same topic, then the value of the potential function will be low.

To model the previous ideas, we define the following potential function, named *unnormalized potential function*, as it takes into account the absolute value of the document-topic counts. It is defined as follows:

$$f_l(z, d) = \sum_{\substack{d' \in D \\ d' \in L_d^m}} \log \max(\lambda, N_{d'z}) + \sum_{\substack{d' \in D \\ d' \in L_d^c}} \log \frac{1}{\max(\lambda, N_{d'z})} \tag{44}$$

where λ is the hyper-parameter which controls the strength of each $l \in L$. Larger values of λ imply that the constraint is active only for those topic assignments that have large counts. The value of λ must be set for each piece of information according to the domain expert's confidence. The conditional probability of topic z, including the defined document constraint potential function, can be estimated as:

$$P(z_{nd}|w, \mathbf{z}^{\neg nd}, \mathbf{y}, \alpha, \beta, \eta, \nu, L) \propto \tag{45a}$$

$$\left(N_{dz_{nd}}^{\neg nd} + \alpha\right) \frac{N_{z_{nd}w}^{\neg nd} + \beta}{N_{z_{nd}\cdot}^{\neg nd} + W\beta} \tag{45b}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=1}} \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{45c}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=0}} 1 - \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{45d}$$

$$\cdot \prod_{\substack{d' \in D \\ d' \in L_d^m}} \max(\lambda, N_{d'z_{nd}}) \prod_{\substack{d' \in D \\ d' \in L_d^c}} \frac{1}{\max(\lambda, N_{d'z_{nd}})} \tag{45e}$$

where, similarly to Equation 4.1, the term (45b) corresponds to the Gibbs sampling of the standard LDA, equation (45c) represents the

sigmoid link likelihood function of RTM when a link exists, and equation (45d) denotes the link function of RTM when a link is absent, and (45e) corresponds the incorporation of additional information through the Unnormalized potential function.

The selection of the correct values for λ is not trivial due to the different lengths of the documents involved in a constraint. For example, if we choose a value for λ that is too large, a document with a number of words less than λ will not affect the probability $p(z_{nd} = t | w_{nd}, \mathbf{z}^{\neg nd}, \mathbf{y}, L)$, even if all the words of the documents are assigned to topic t.

To smooth the effect of the hyperparameter λ, we propose a potential function that takes into account the length of the document in the following.

4.2.3 *Normalized Document Potential Function*

The potential function that we propose hereby considers the proportion of words in a document assigned to the same topic, rather than the absolute values of the document-topics counts. We define the potential function $f_l(z, d)$ as follows:

$$f_l(z, d) = \sum_{\substack{d' \in D \\ d' \in L_d^m}} \log\left(\frac{N_{d'z}}{N_{d'\cdot}} + 1\right) - \sum_{\substack{d' \in D \\ d' \in L_d^c}} \log\left(\frac{N_{d'z}}{N_{d'\cdot}} + 1\right) \tag{46}$$

The conditional probability of topic z estimated by D-CRTM, including the defined document constraint potential function, can be specified as follows:

$$P(z_{nd} | w, \mathbf{z}^{\neg nd}, \mathbf{y}, \alpha, \beta, \eta, \nu, L) \tag{47a}$$

$$\propto \left(N_{dz_{nd}}^{\neg nd} + \alpha\right) \frac{N_{z_{nd}w}^{\neg nd} + \beta}{N_{z_{nd}\cdot} + W\beta} \tag{47b}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=1}} \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{47c}$$

$$\cdot \prod_{\substack{d' \neq d \\ y_{dd'}=0}} 1 - \sigma\left(\frac{\eta_{z_{nd}}}{N_{d\cdot}} \cdot \frac{N_{d'z_{nd}}}{N_{d'\cdot}} + \sum_{k=1}^{K} \eta_k \frac{N_{dk}^{\neg nd}}{N_{d\cdot}} \frac{N_{d'k}}{N_{d'\cdot}} + \nu\right) \tag{47d}$$

$$\cdot \prod_{\substack{d' \in D \\ d' \in L_d^m}} \left(1 + \frac{N_{d'z_{nd}}}{N_{d'\cdot}}\right) \prod_{\substack{d' \in D \\ d' \in L_d^c}} \frac{1}{1 + \frac{N_{d'z_{nd}}}{N_{d'\cdot}}} \tag{47e}$$

In the following sections, we present an experimental investigation to evaluate the capabilities of D-CRTM to discover hidden topics in different collections of networked documents.

4.2.4 *Experimental Setting*

To evaluate the performance of the proposed models, we conduct several experiments using document labels as additional information and we compare the performance of the D-CRTMs with different baseline models' performance.

BASELINE MODELS. We validate D-CRTM-N and D-CRTM-U by comparing the results on benchmark datasets against the following models:

- LDA: Latent Dirichlet Allocation (Blei et al., 2003a) using collapsed Gibbs sampling.
- RTM: standard RTM (Chang and Blei, 2009) that models only the links between documents through the binary variable $\mathbf{y}$, without incorporating any other kind of domain-specific information.
- Bi-RTM: RTM for bidimensional networks, where the first dimension is intended to represent the links of the document network, modeled by the binary variable $\mathbf{y}$, and the second dimension is designed to represent the must- and cannot-constraints between documents, modeled by an additional binary variable $\mathbf{c}$. Its joint probability distribution is the following:

$$\begin{aligned} &p(\mathbf{w}, \mathbf{z}, \mathbf{y}, \mathbf{c}, \theta, \mathbf{\Phi} | \alpha, \beta, \eta, \nu, \eta', \nu') \\ &= \prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(w_{nd}|\Phi_{z_{nd}}) p(z_{nd}|\theta_d \prod_{k=1}^{K} p(\Phi_k|\beta) \\ &\cdot \prod_{\substack{d,d' \in D \\ d' \neq d}} \psi_\sigma(y_{dd'}|z_d, z_{d'}, \eta, \nu) \\ &\cdot \prod_{\substack{d,d' \in D \\ d' \neq d}} \psi_{\sigma'}(c_{dd'}|z_d, z_{d'}, \eta', \nu') \end{aligned} \tag{48}$$

  where η' and ν' are respectively the coefficient and the intercept for the sigmoid function $\psi_{\sigma'}$ that models the likelihood that a constraint $c_{dd'}$ between two documents d and d' exists.

For the sake of completeness, we also compared the proposed D-CRTM models with a supervised model. In particular, we considered Labeled LDA (LLDA) as defined in Section 3.1.2.

Let us notice that in some realistic cases, we only know that two documents belong or do not belong to the same class, rather than knowing that to which class a document belongs.

BENCHMARK DATASETS. Since D-CRTM deals with networks of documents and domain-specific information, the chosen datasets for the validation phase must have two main features: an underlying document-relational structure (e.g. citation links) and some domain-specific information available (e.g. document labels) to derive the semi-supervised constraints. Table 4.1 contains some statistics about the selected benchmarks. Cora (McCallum et al., 2007) and M10 (Lim and Buntine, 2014) are two datasets composed of 2708 and 4427 scientific publications respectively, whose links are represented by citations. WebKB[1] is a dataset composed of 877 universities web pages whose relationships are hyperlinks from a web page to another.

Dataset	#docs	#links	Density	Type of link	#classes	#unique words
Cora	2708	5430	$2.87 \cdot 10^{-4}$	citation	7	1752
M10	4427	5627	$7.41 \cdot 10^{-4}$	citation	9	1592
WebKB	877	1131	$14.72 \cdot 10^{-4}$	hyperlink	5	1830

Table 4.1: Statistics of the benchmark datasets Cora, WebKB, and M10.

The three benchmarks have been pre-processed: words are stemmed, stopwords and the least and most frequent words are removed. Only the documents that link another document or are linked by a document at least once are considered. Prior information has been introduced in terms of constraints using a percentage of the possible constraints between documents. In particular, if two documents d and d' randomly chosen share the same class label, we expect that their words are assigned to similar topics, therefore a must-constraint is introduced (i.e. document d is added to the must-constraint set of d' and document d' is added to the must-constraint set of d). Concerning LLDA, we first define a mapping between the set of topics and the set of labels. Two documents d and d' are randomly drawn and the labels l_d and $l_{d'}$ are incorporated as domain-specific information, according to Equation (18).

4.2.4.1 *Performance Measures*

Each dataset is divided into a training set and a test set. The models are evaluated on the test set by measuring their performance on a document classification task. The K-dimensional representation of each document output by the considered topic model, i.e. the document-topic distribution θ, is used to train a linear Support Vector Machine (SVM) classifier that predicts the document classes. For the experimental evaluation, we considered both micro-F1 and macro-F1 measures as defined in 3.4.1.

1 http://www.cs.cmu.edu/~WebKB/ILP-data.html

4.2.4.2 *Parameter settings*

Each experiment, with a given set of parameters, has been repeated 100 times. The performance measures have been averaged by the number of the samples, thus obtaining an average micro-F1 and macro-F1 measure. The hyperparameters α and β have been set equal to $50/K$ and 0.1 respectively (as reported in (Griffiths and Steyvers, 2004)), for all the considered models. The selected value of λ for D-CRTM-U is 1. Each model has been trained for 1,500 Gibbs iterations.

The models have been validated by setting the number of topics equal to the number of classes of the dataset and by varying the quantity of information, i.e. the number of possible constraints, during the training phase and, in a second stage, during the testing phase.

The maximum quantity of prior information in terms of constraints is $\frac{D(D-1)}{2}$ (where D is the number of documents), which represents the maximum number of possible pairs among all the documents of the dataset. The quantity of information introduced into the models is expressed as a percentage, preferring low values to maintain the typical semi-supervised scenario. Thus, given a percentage p, the number of constraints introduced into the model will be $p \cdot \frac{D(D-1)}{2}$, rounded down to the nearest integer. When the percentage of incorporated information is equal to 0%, then D-CRTM and LLDA correspond to RTM and LDA, respectively.

We used Support Vector Machines (SVM) to predict the ground truth labels from the document-topic distribution of the documents. In particular, we used the LibSVM implementation[2] for inducing the linear SVM classifier.

The code of the proposed models is available at `https://github.com/MIND-Lab/Constrained-RTM`.

4.2.5 *Results*

In the following, we consider the performance of each model with an increasing percentage of prior information introduced only in the training phase. In particular, we consider an experimental setting with zero knowledge (0.0%), which corresponds to models that do not encode any constraint (i.e. LDA and RTM), and we represent them in the plots using the lines. The other models, i.e. BiRTM, D-CRTM-U, D-CRTM-N, and LLDA, are reported with different percentages of information, and the bar plots represent them.

Let us notice that, in these experiments, zero additional information is incorporated in the testing phase, as it often happens in realistic cases.

2 LibSVM library: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

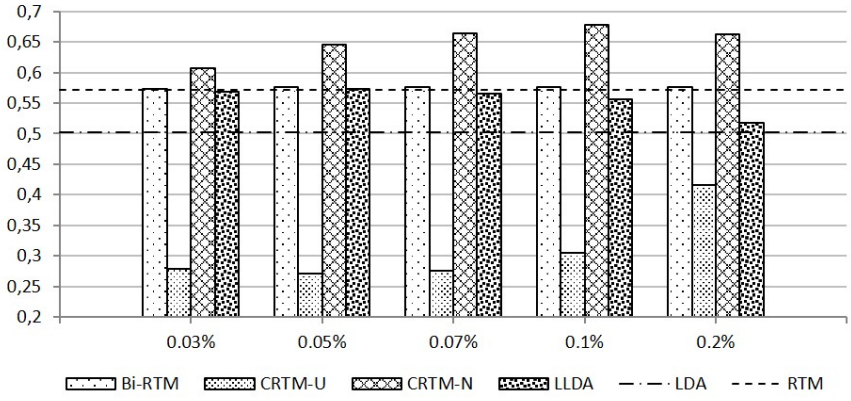


Figure 4.1: Micro-F1 performance of the compared models on Cora.

QUANTITATIVE RESULTS. Figure 4.1 shows the performance of the models measured using the micro-F1 on the dataset Cora, where the number of constraints that are randomly selected ranges from 0.03% and 0.2% of the number of possible constraints. D-CRTM-N outperforms the other models, increasing its performance as more additional information is introduced, and it seems to decrease its performance for larger quantities of constraints. We can also notice that, while the performance of Bi-RTM is invariant for the quantity of domain-specific information, LLDA gets at first an improvement with a small contribution of additional information, then its performance decreases for larger values. The performance of D-CRTM-U is worse than the baselines LDA and RTM. This behavior may be related to the fact that documents in Cora are long (the average length of a document is 68.9 words). Therefore the value of the potential function, which depends on the number of words associated with the current topic, will be very high, allowing a small contribution to the rest of the sampling.

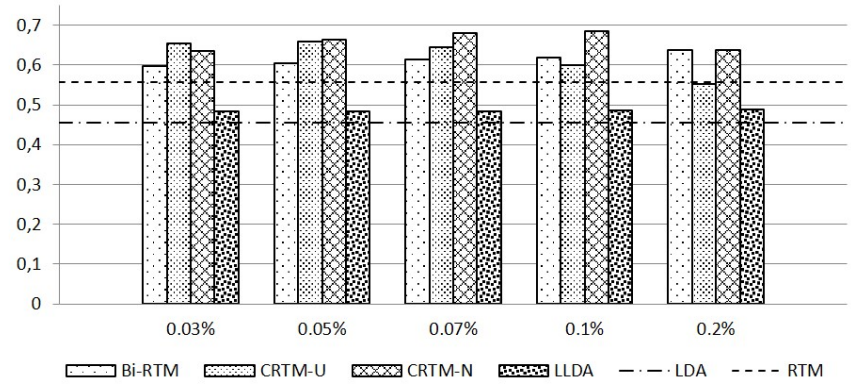


Figure 4.2: Micro-F1 performance of the compared models on M10.

In Figure 4.2, we show the results for dataset M10. D-CRTM-N has a similar behavior with respect to the previous experiments, while D-CRTM-U gets an improvement with a small insertion of constraints. This is due to the lengths of the documents in M10, which are short (the average length of documents in M10 is 6.3 words). This makes the introduction of the constraints more smoothed than in Cora. However, for larger quantities of additional information, the average performance of D-CRTM-U gets worse. The introduction of the labels allows LLDA to obtain a small improvement with respect to LDA, meaning that associating each word of a labeled document to the same topic does not improve the generalization capabilities of the model. Bi-RTM has a higher performance as the quantity of additional information increases, although it requires many constraints and its best performance is still lower than D-CRTM-N.

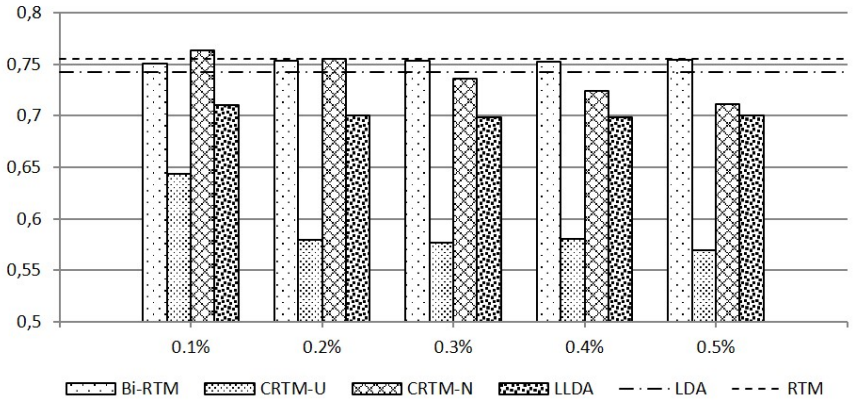


Figure 4.3: Micro-F1 performance of the compared models on WebKB.

Figure 4.3 shows the performance of the models for the dataset WebKB. D-CRTM-N still has the same behavior as the previous datasets, obtaining the best performance. Bi-RTM has a constant trend, while the other two models get worse performances with respect to LDA and RTM. The behavior of D-CRTM-U is similar to the one obtained in Cora. In fact, also WebKB is composed of long documents. On the other hand, LLDA has a lower performance with respect to Bi-RTM, D-CRTM-N, and LDA.

We report in the following the results of the considered models on the different datasets by introducing domain-specific information both in the training and testing set. In particular, each combination of values of percentage in training and testing has been considered. To provide a concise visualization of the performance of the models, the results have been averaged, and therefore Figure 4.4 reports the best average performance for each model.

The two D-CRTMs significantly outperform Bi-RTM and the baselines LDA and RTM (with confidence of 95%). In particular, the two proposed models have similar performance on M10 and WebKB, while

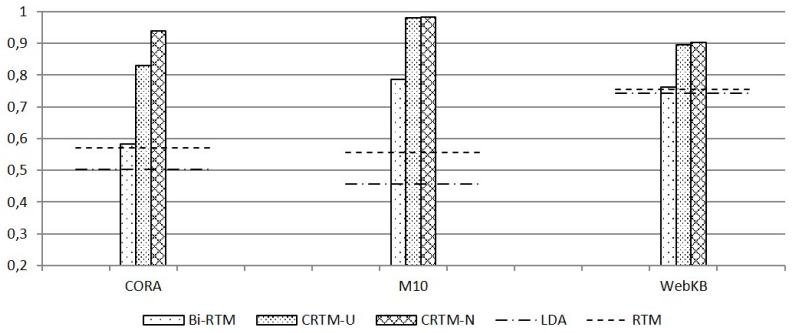


Figure 4.4: Micro-F1 measure of the models across all the datasets. The plot shows the best performance of the average behavior of the models, considering different percentages of constraints introduced in the training phase and in the test phase.

D-CRTM-N outperforms its counterpart D-CRTM-U, because it can handle the documents' length issue of the Cora dataset. Bi-RTM outperforms standard RTM and LDA, however in Cora and WebKB the improvement in the performance is small, meaning that modeling the document constraints using the link likelihood function $\psi_{\sigma'}$ may not be a promising solution.

We do not report LLDA in this evaluation because, in LLDA, all of the words of a labeled document are associated with the same topic. This has the trivial effect of automatically labeling each document affected by a constraint in the test set with the correct class. D-CRTMs still have very promising results, and they can be applied in more realistic cases, i.e. when we do not know the exact labels of documents, but we know that two documents belong to the same class. In this scenario, LLDA cannot be used.

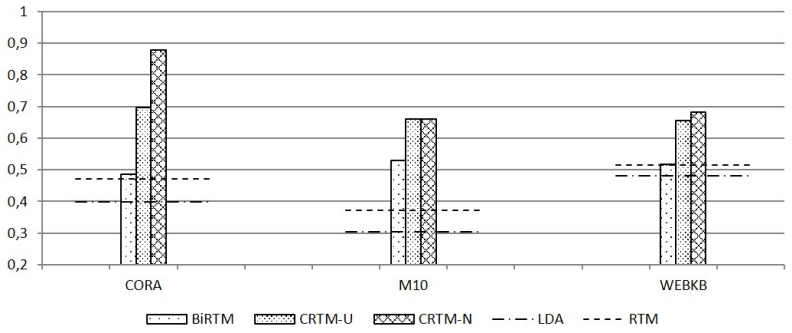


Figure 4.5: Macro-F1 measure of the models across all the datasets. The plot shows the best performance of the average behavior of the models, considering different percentages of constraints introduced in the training phase and in the test phase.

We show a further comparison in Figure 4.5, where the results are reported in terms of macro-F1 measure, with additional information introduced both in training and testing. We can easily notice that

macro-F1 values are lower than the micro-F1 ones, highlighting that all the models are negatively affected by the class/topic size. This means that all the LDA-based models tend in general to fit better those classes with higher cardinality at the expenses of the minority classes.

This behavior is mainly motivated by the symmetric and positive (> 1) values of the hyper-parameter α that regulates the corresponding document-topic distribution θ. In fact, this setting implies having the same prior distribution of topics (and classes) for each document, originating therefore a posterior topic/classes distribution that is almost uniform and consequently balanced among different classes. In Chapter 6 we will show how the choice of the hyperparameters have an impact on the performance of the topic models.

Even if D-CRTM is sensitive to the hyper-parameter α, it still outperforms the other baselines. The promising performance in terms of macro-F1 is mainly due to its ability to smooth the posterior topic distributions by the introduction of constraints.

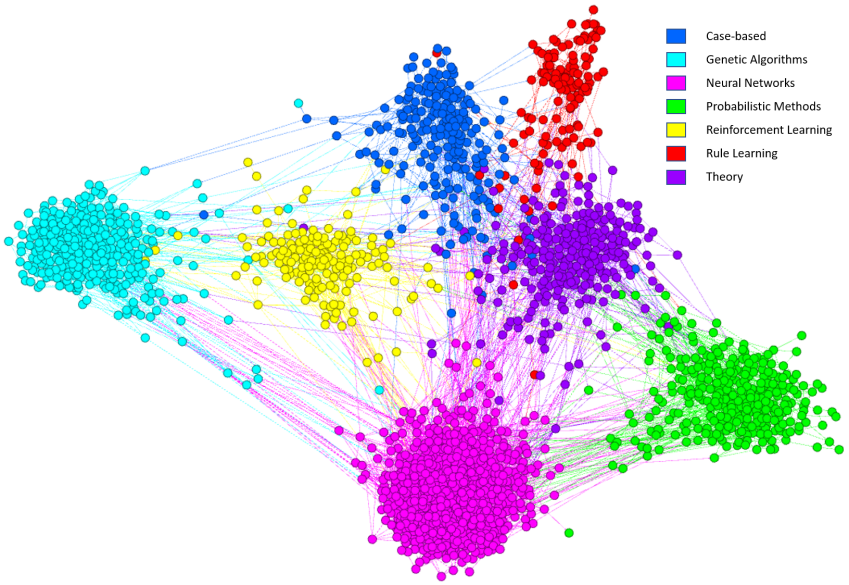


Figure 4.6: An example of the Cora network used during the training phase. Edges represent either citation links or must-constraint, where 0.2% of additional information is incorporated. Nodes are colored with respect to their actual class.

OBSERVATIONS ON THE NETWORK STRUCTURE. To show the complexity of the network obtained by the combination of links and must-constraints, we illustrate an example of the Cora benchmark. Figure 4.6 shows an instance of a document network when 0.2% of additional information is introduced during the training phase. In particular, each node denotes a document, whose color represents the actual document class. The edges denote either a citation link or

a must-constraint. Since must-constraints are allowed only between documents of the same class, this type of relationships forms seven connected components that are visible by observing the network. On the other hand, citations can exist either between same-class documents or documents belonging to different classes.

The density of the citation network, together with the density of the constraints, can have an impact on the classifier's performance, which decreases when too much additional information is introduced. To better clarify this issue, we consider the ego network of the document 40886 (where 40886 is the original identifier of the document in the Cora dataset), as illustrated in Figure 4.7.

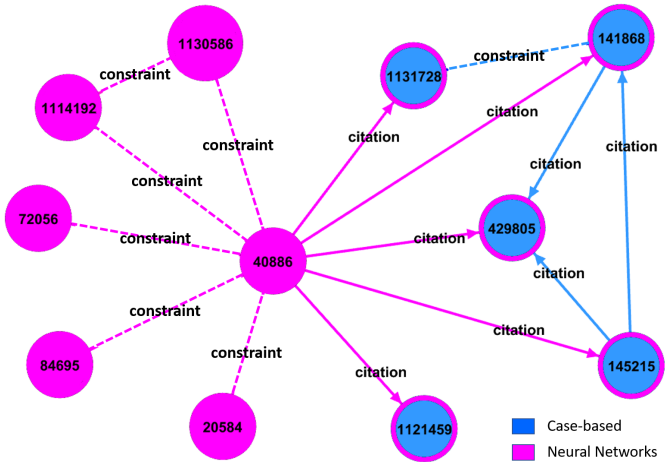


Figure 4.7: Ego network of document 40886 of the Cora dataset. The nodes are labeled by the original identifiers of the dataset and edges are labeled by the relationship type (citations are denoted by a straight line and must-constraint by a dashed line). The node color represents the actual class of a document, and the color of the outline denotes the predicted class.

In particular, the color of the node represents the actual class of a document, while the color of the outline denotes the predicted class (e.g. documents 40886 and 429805 are classified as belonging to the class "Neural Network", but the first is correctly predicted while the second is misclassified).

As expected, the proposed model D-CRTM-N encourages all the purple nodes to have similar topic distributions, and the classifier correctly predicts that all the documents belong to the class "Neural Network". Analogously, all the blue nodes are encouraged by both the must-constraints and the citation links to have similar topic distributions and are assigned to the *same* class, though the predicted class is not correct. This error is likely due to the presence of citation links between documents of different classes (e.g. the citation between documents 40886 and 429805) combined with the must-constraints. If a document is misclassified, a must-constraint may propagate this error

to all the other documents that are must-constrained to the misclassified ones.

This error could be reduced through the use of cannot-constraints, which can be incorporated if two documents belong to different classes. In this way, a cannot-constraint between two documents would allow the topic distributions to be dissimilar, originating therefore a correct classifier's prediction.

4.3 MODELING WORD-LEVEL RELATIONAL INFORMATION INTO DOCUMENT-LEVEL RELATIONAL TOPIC MODELS

We have shown in the previous part of the Chapter that incorporating additional information related to documents can help to enhance the topical representation of the documents, providing improvements for document classification tasks. Since the discovered topic words, derived from the word distributions, are strongly connected with the topical representations of the documents, one may wonder if the incorporated relationships affect also the quality of the resulting topics.

Moreover, previous work proved that the introduction of additional relational information between words improves the coherence of the discovered topics (Chen et al., 2013b,c; Yang et al., 2015c). This type of relationship is commonly viewed as related to the concept of synonym, but this is not always the case in a real-world scenario because of word ambiguity. Following this intuition, it is thus important to take into consideration the concept behind the word alongside the word itself for understanding its relationship with other words, because it would permit to associate the same topic to words that are actually related and not only synonyms. For example, it would be possible to grasp that the word "engine", when associated with the concept of "search engine", is distant from "motor", but similar to "information retrieval". Few works investigate the use of named entities in topic models (Allahyari and Kochut, 2016; Kim et al., 2012; Wang et al., 2017), but none of them addresses the problem as a relational setting. The constrained functions we have defined before can be indeed applied to word tokens too (Yang et al., 2015c).

Therefore, in this section, we would like to investigate the role of two different types of relational information: (1) concept relationships between words and named entities and (2) document-level relationships extracted by a document network. The impact of these two types of relational information is evaluated by extending traditional topic models using different potential functions.

We therefore propose *Entity Constrained Latent Dirichlet Allocation (E-CLDA)* and *Entity Constrained Relational Topic Models (E-CRTM)*, two classes of models aimed at incorporating entity-entity and entity-word relationships in traditional topic models. Following the previous work (Terragni et al., 2020; Yang et al., 2015c), we constrain the

joint distribution of LDA and RTM through the use of potential functions that model entity-entity and/or entity-word relationships.

4.3.1 *Definition of Information Sets*

We define the vocabulary E containing the unique named entities of the corpus, and the vocabulary W containing the unique words. We derive the vocabulary Γ as the union of the word and named entity vocabularies. Similarly to D-CRTM, relationships are denoted by the set of information L and each piece of information $l \in L$ is incorporated by a potential function $f_l(z, u)$, which represents a real-valued score for the hidden topic assignment z of the word or named entity token u.

We derive the information L from the similarities of embeddings in a word and entity embedding space derived from Skip-Gram (Mikolov et al., 2013). Given a word (and entity) embeddings training set composed of a large but finite set Λ, the word (and entity) embeddings model can be expressed as a mapping function $C' : \Gamma \mapsto \mathbb{R}^t$. For each token $u \in \Gamma$, we define a *must-constraint* set L_u^m, containing words and named entities that are likely to share the same themes of u. L_u^m is defined as:

$$L_u^m = \{v \in \Gamma | sim(C'(u), C'(v)) > \epsilon_m\} \tag{49}$$

where sim is the cosine similarity between two vectors, and ϵ_m is a given threshold. We also define a *cannot-constraint* set L_u^c, that contains the words and named entities that are not likely to share the same themes of u. L_u^c is defined as:

$$L_u^c = \{v \in \Gamma | sim(C'(u), C'(v)) < \epsilon_c\} \tag{50}$$

where ϵ_c is a given threshold.

An example of a must-constraint set for the named entity

$$L^c_{\substack{\text{Artificial} \\ \text{Neural Network}}} = \{\textit{Artificial neuron}, \textit{perceptron}\}$$

which contains named entities that are likely to be assigned to the same topic. Analogously, an example of a cannot-constraint set for the same named entity is:

$$L^c_{\substack{\text{Artificial} \\ \text{Neural Network}}} = \{\textit{Olympic\_Games}, \textit{Athlete}, \textit{medallist}\}$$

which denotes named entities related to *sports* and not to Machine Learning.

We report the joint distribution of the proposed models. Entity Constrained Latent Dirichlet Allocation (E-CLDA) defines the following joint probability distribution:

$$P(\mathbf{u}, \mathbf{z}, \theta, \mathbf{\Phi}|\alpha, \beta, L) \propto \tag{51a}$$

$$\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(u_{nd}|\Phi_{z_{nd}}) p(z_{nd}|\theta_d) \tag{51b}$$

$$\prod_{k}^{K} p(\Phi_k|\beta) \cdot \xi(\mathbf{z}, L) \tag{51c}$$

where, differently from the joint probability distribution of LDA (Equation 2), the vocabulary set is Γ and there is also the term representing the potential function.

Similarly, the joint probability distribution of Entity Constrained Relational Topic Models (E-CRTM) is defined as follows:

$$P(\mathbf{u}, \mathbf{z}, \mathbf{y}, \theta, \mathbf{\Phi}|\alpha, \beta, \eta, \nu, L) \propto \tag{52a}$$

$$\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(u_{nd}|\Phi_{z_{nd}}) p(z_{nd}|\theta_d) \tag{52b}$$

$$\prod_{k}^{K} p(\Phi_k|\beta) \prod_{\substack{d,d' \in D \\ d' \neq d}} \psi_\sigma(y_{d,d'}|z_d, z_{d'}, \eta, \nu) \cdot \xi(\mathbf{z}, L) \tag{52c}$$

where ψ_σ is the link probability function as defined in Equation 19.

4.3.2 *Entity-Entity Potential Function*

We specify an entity-entity potential function that models the relationships between named entities. Let $N_{ze'}$ be the maximum between 1 and the topic-entities counts, i.e. the number of occurrences of e' assigned to topic z. The function $f_l(z, u)$ is as follows:

$$f_l(z, u) = \begin{cases} \sum_{\substack{e' \in L_u^m \\ e' \in E}} \log N_{ze'} + \sum_{\substack{e' \in L_u^c \\ e' \in E}} \log \frac{1}{N_{ze'}} & \text{if } u \in E \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

The function increases the probability that the entity u will be assigned to the same topics as those of the entities belonging to L_u^m. Similarly, the potential function decreases the probability that a named entity u will be drawn from the same topics as those of entities contained in the L_u^c.

The models that can encode the Entity-Entity (EE) potential function will be referred to as E-CLDA-EE and E-CRTM-EE.

4.3.3 *Entity-Word Potential Function*

Let $N_{zw'}$ be the maximum between 1 and the topic-word counts, i.e. the counts of word w' assigned to topic z. The following potential function deals with relationships between entities and word tokens:

$$f_l(z,u) = \begin{cases} \sum\limits_{\substack{w' \in L_u^m \\ w' \in W}} \log N_{zw'} + \sum\limits_{\substack{w' \in L_W^c \\ w' \in W}} \log \frac{1}{N_{zw'}} & \text{if } u \in E \\ \sum\limits_{\substack{e' \in L_u^m \\ e' \in E}} \log N_{ze'} + \sum\limits_{\substack{e' \in L_u^c \\ e' \in E}} \log \frac{1}{N_{ze'}} & \text{if } u \in W \end{cases} \tag{54}$$

The potential function models the following cases:

- if u is a named entity, then we consider only the words that are contained in u's must- and cannot-constraint sets, i.e. L_u^m and L_u^c;
- if u is a word, then we consider only the named entities that are contained in u's must- and cannot-constraint sets, i.e. L_u^m and L_u^c.

The models encoding Entity-Word (EW) relationships are named E-CLDA-EW and E-CRTM-EW.

4.3.4 *Experimental Setting*

Since our objective is to evaluate the contribution of the different incorporated relationships, we consider the proposed models (i.e., E-CLDA-EE, E-CLDA-EW and E-CRTM-EE, E-CRTM-EW), and also the baselines LDA and RTM. We also consider two neural counterparts of LDA and RTM: Stacked Variational Auto-Encoder (SVAE) (Bai et al., 2018; Miao et al., 2016) and Neural Relational Topic Model (NRTM) (Bai et al., 2018).

The code of the proposed models is available at `https://github.com/MIND-Lab/ec-rtm`.

DATASETS We perform the experimental investigation on two relational datasets Cora (McCallum et al., 2005) and WebKB[3]. Table 4.2 reports the basic statistics of the datasets.

DATASET PREPROCESSING. The identification of named entities in the text is typically performed through a series of techniques that refer to the task of Named Entity Recognition (NER) (Fersini et al., 2014; Li et al., 2020; Ritter et al., 2011). Once the named entities are recognized, the next step is to associate them to unambiguous concepts, as

3 `www.cs.cmu.edu/~WebKB/ILP-data.html`

Datasets	#Docs	#Links	Document Type	Link Type
Cora	2,708	5,278	Title+Abstract	Citation
WebKB	877	1,608	Webpage	Hyperlink

Table 4.2: Statistics of benchmark datasets Cora and WebKB.

for example resources in a Knowledge Base. This process is known as the task of Named Entity Linking (NEL) (Cucerzan, 2007; Dredze et al., 2010; Nozza et al., 2019).

Here we use the DBPedia Spotlight tool (Mendes et al., 2011) (with confidence = 0.5 and support = 0.0) to identify named entities in the text and associate them to DBPedia units. We added the prefix "NE/" to each identified entity to discriminate it from words. Then, we apply a common pre-processing technique to the text. We lowercase the text, remove English stopwords and words occurring less than 10 times, and filter out documents composed of less than 2 words. Details on the vocabulary composition are reported in Table 4.3. We consider only must-constraints, which have been extracted from Wikipedia2Vec (Yamada et al., 2018).

	Processed corpus			**Unprocessed corpus**
	# unique entities	**# unique words**	**# unique entities and words**	**# unique words**
Cora	384	2,675	3,059	3,012
WebKB	355	1,874	2,229	2,247

Table 4.3: Summary of the vocabularies for the benchmark datasets before and after the pre-processing phase.

HYPERPARAMETERS. Each experiment, with a given set of parameters, is repeated 100 times. The performance measures are averaged over the number of samples. The hyperparameters α and β are set equal to 50/K and 0.1 respectively (as reported in (Griffiths and Steyvers, 2004)) for all the considered models. All the compared models are trained for 1,500 Gibbs iterations.

In our evaluation, we consider only must-constraint relations that entities and words can generate. To select the most appropriate value for the threshold ϵ_m, we studied the performance of the topic coherence of our models by varying the value of the parameter. The final values for the models with the potential functions EE and EW are 0.8 and 0.7 for the dataset Cora and 0.6 and 0.6 for WebKB. The selected value for λ is 1.

	TD			*NPMI*			C_V		
	10	30	50	10	30	50	10	30	50
LDA	0.816	0.736	0.654	0.098	0.080	0.071	0.399	0.389	0.386
RTM	0.814	**0.747**	0.666	0.099	0.082	0.071	0.348	0.391	0.392
E-CLDA-EE	0.814	0.742	0.659	0.098	0.079	0.069	0.397	0.390	0.389
E-CLDA-EW	0.817	0.740	0.660	0.094	0.079	0.070	0.395	0.389	0.387
E-CRTM-EE	0.817	0.747	**0.675**	**0.099**	0.081	0.071	0.402	0.394	0.392
E-CRTM-EW	0.820	0.746	0.671	0.098	**0.082**	**0.072**	0.340	0.392	0.392
SVAE	**0.893**	0.694	0.577	-0.099	-0.095	-0.096	**0.456**	**0.456**	**0.453**
NRTM	0.857	0.525	0.381	-0.083	-0.082	-0.082	0.442	0.447	0.446

Table 4.4: Topic diversity and coherence performance on the Cora dataset with the number of topics equal to 10, 30, 50.

	KL–U			*KL–V*			*KL–B*		
	10	30	50	10	30	50	10	30	50
LDA	1.855	1.572	1.259	1.226	1.231	1.059	0.052	0.119	0.168
RTM	2.001	2.046	1.820	1.357	1.563	1.460	0.095	**0.207**	**0.283**
E-CLDA-EE	1.845	1.520	1.375	1.225	1.238	1.066	0.052	0.119	0.167
E-CLDA-EW	1.800	1.518	1.381	1.230	1.236	1.065	0.052	0.119	0.168
E-CRTM-EE	2.033	**2.082**	**1.849**	**1.362**	1.564	**1.472**	0.095	0.205	0.280
E-CRTM-EW	**2.079**	1.990	1.643	1.361	**1.565**	1.470	**0.096**	0.206	0.282

Table 4.5: KL-* performance on the Cora dataset with the number of topics equal to 10, 30, 50.

METRICS We use *KL-U*, *KL-V*, and *KL-B* to measure semantic importance and identify junk and insignificant topics (AlSumait et al., 2009). We also measure how different the topics are from each other by computing the Percentage of Unique Words (**PUW**) (Dieng et al., 2020) on the 10-top words of the topics. Finally, we consider two metrics of topic coherence, i.e. **NPMI** (Aletras and Stevenson, 2013) and $\mathbf{C_V}$ (Röder et al., 2015) that measure how much the 10-top words of a topic are related to each other. The scores are computed using the Palmetto toolkit[4] and Wikipedia[5] as reference corpus. We refer the reader to Section 3.4 for additional details on the evaluation metrics.

4.3.5 *Results*

QUANTITATIVE RESULTS Tables 4.4, 4.5, 4.6 and 4.7 show the performance of the models in terms of all the considered scores over an increasing number of topics on the datasets.[6] Results show that

4 http://github.com/dice-group/Palmetto
5 English Wikipedia dump of the 23rd of May, 2019.
6 Computing the KL- metrics is impractical for SVAE and NRTM since they do not model word- and document-topic distributions.

	TD			*NPMI*			C_V		
	10	30	50	10	30	50	10	30	50
LDA	0.761	0.617	0.538	0.039	0.040	0.030	0.378	0.379	0.379
RTM	0.760	0.608	0.532	0.043	0.043	0.036	0.377	0.380	0.380
E-CLDA-EE	0.769	0.623	0.542	0.043	0.041	0.033	0.379	0.380	0.381
E-CLDA-EW	0.764	**0.651**	**0.547**	0.042	0.038	0.033	0.376	0.381	0.382
E-CRTM-EE	0.760	0.612	0.536	**0.048**	**0.043**	**0.039**	0.377	0.382	0.381
E-CRTM-EW	0.759	0.639	0.543	0.045	0.042	0.036	0.377	0.382	0.384
SVAE	**0.829**	0.563	0.454	-0.116	-0.110	-0.112	**0.460**	0.450	0.452
NRTM	0.734	0.360	0.283	-0.114	-0.117	-0.119	0.454	**0.455**	**0.458**

Table 4.6: Topic diversity and coherence performance on the WebKB dataset with the number of topics equal to 10, 30, 50.

	KL–U			*KL–V*			*KL–B*		
	10	30	50	10	30	50	10	30	50
LDA	1.695	1.256	1.130	1.054	0.943	0.775	0.069	0.142	0.199
RTM	**1.986**	1.795	1.430	**1.202**	1.239	1.109	**0.119**	0.225	**0.303**
E-CLDA-EE	1.643	1.289	1.061	1.055	0.948	0.780	0.069	0.143	0.200
E-CLDA-EW	1.736	1.345	1.075	1.062	0.981	0.784	0.069	0.138	0.198
E-CRTM-EE	1.867	**1.944**	1.468	1.199	1.246	1.119	0.118	**0.226**	0.303
E-CRTM-EW	1.979	1.786	**1.646**	1.199	**1.294**	**1.127**	0.117	0.217	0.302

Table 4.7: KL-* performance on the WebKB dataset with the number of topics equal to 10, 30, 50.

models that consider relational information generally obtain higher performance than their non-relational counterparts. Differently, introducing the concept constraints in E-CRTM-EE and E-CRTM-EW models does not seem to provide significant improvements with respect to RTM. This can be motivated by the fact that the constraint sets additionally included in the E-CRTM models are already captured in the word-topic distribution obtained by RTM.
Different behaviors can be observed for the C_V scores, for which NRTM and SVAE obtain significantly higher performance. This opposite trend with respect to the other topic scores can be explained by the fact that C_V rewards the presence of rare words even if they are contained in junk topics as stated by the author of (Röder et al., 2015)[7].

Models	**Top-10 words**
LDA*	problem genetic algorithms problems programming search optimization fitness population space
RTM*	genetic control programming fitness reinforcement population algorithms paper environment behavior
E-CRTM-EE	NE/Genetic_programming programs NE/Genetic_algorithm population fitness genetic evolutionary program NE/Evolution strategies
E-CRTM-EW	NE/Genetic_programming NE/Genetic_algorithm population fitness genetic evolutionary NE/Evolution encoding operator operators
SVAE	koza NE/Multidisciplinary_design_optimization splice bitsback NE/Genetic_programming fitness orientation NE/-Ploidy NE/Exon coded
NRTM	genetic reactive NE/Genetic_programming NE/Case casebased neuroevolution ssa NE/Genetic_algorithm coevolutionary problemsolving

Table 4.8: Example of the topic “Genetic Programming" in Cora.

QUALITATIVE RESULTS In Table 4.8, we show the top-10 words for Cora concerning an example topic “Genetic Programming" for E-CRTM-EE, E-CRTM-EW, LDA, RTM, SVAE, and NRTM. To analyze if the named entity annotation can contribute to topic interpretability, we report the words of LDA and RTM (referred to as LDA* and RTM*) run on Cora composed of words only. As expected from the quantitative results, the topics extracted by the proposed models do not significantly differ from RTM*, further demonstrating the hypothesis that the imposed constraints were already captured by the original model.
Qualitative considerations can be made regarding the exploitation of

7 https://bit.ly/3jApSAC

the novel entity-level modeling of the documents. While this representation leads to topics containing explicit concepts (e.g., “NE/Genetic_programming"), topics obtained by RTM* seem to be equally interpretable because they can identify named entities in the form of distinct words (e.g., “genetic, programming, algorithm"). Moreover, the difference in representation is only evident when named entities are composed of two or more words (e.g., “NE/Evolution" and “evolution" are equivalent). The benefit of applying NEEL techniques for recognizing named entities in topics may come in handy for automatically providing links to KB (such as Wikipedia), at the computational cost of discovering named entities. Moreover, the proposed potential functions would allow users to artificially manipulate the model to derive explanations for the topic assignments or force entities in the same topic based on human domain knowledge.

Regarding SVAE and NRTM, their topics seem hard to interpret from a qualitative perspective, confirming the results of the quantitative evaluation.

4.4 SUMMARY OF THIS CHAPTER

In the following, we give a short summary of this chapter. In the introductory section of the Chapter, we reported the following research questions:

Q4.1 How can we incorporate document-level and word-level relational information into classical topic models?

Q4.2 What is the impact of modeling document-level and word-level relational information into topic models?

To answer question Q4.1, we have defined potential functions to model the relationships between documents (Section 4.2) and between words and named entities (Section 4.3) in the form of constraints. We have proposed two novel definitions of potential functions for modeling document-level relationships, originating the models D-CRTM-U and D-CRTM-N. These models produce accurate document representations, which improve the performance of the classical topic models in document classification tasks.

We have also compared the performance of document-level relational topic models and word-level relational topic models, to estimate the impact of these types of information on the quality of the topics (Q4.2). Our results show that incorporating document relationships can effectively help the model discover more coherent topics. However, the incorporation of word-level relationships seems to be not as effective. A qualitative inspection suggests that the word-level relationships are already captured by the topic model and are therefore superfluous.

Let us notice that our modeling is flexible, modular and easy to implement. It can be applied to other topic models, as long as they belong to the category of probabilistic graphical models. Indeed, as we have shown in Section 4.3, our method can be easily applied both to LDA and to RTM.

5

MODELING CONTEXTUAL INFORMATION IN NEURAL MODELS

In recent years, Neural Topic Models (Dieng et al., 2020; Zhao et al., 2021) have gained increasing popularity due to their flexibility and scalability. **Most of these neural models still use Bag-of-Words (BoW) document representations as input. These representations, though, disregard the syntactic and semantic information of the words in a document**, the two main linguistic avenues to coherent text. In other words, the models based on bag of words represent the input in an inherently incoherent manner. Although the bag-of-words assumption makes sense from a point of view of computational efficiency, it is unrealistic.

These observations are not novel in the topic modeling community. As Wallach (2006) clearly exemplify, the sentences "the department chair couches offers" and "the chair department offers couches" are represented by the same bag of words, but describe different topics. Knowing the context of the word "chair" makes it easier to assign the correct topic. In this Chapter, we will therefore investigate methods to model the context information into neural topic models to overcome the limitations of the bag-of-words assumption.

From the Bigram Topic Model (Wallach, 2006), which incorporates the notion of word order into LDA, other works tried to relax the BoW assumption or enrich the word representations in classical topic models. For example, in Chapter 4 we have seen how to incorporate relationships between word tokens and named-entities in the text. Other approaches use word relationships derived from external knowledge bases (Chen et al., 2013b; Yang et al., 2015c), or pre-trained word embeddings (Das et al., 2015; Dieng et al., 2020; Nguyen et al., 2015; Zhao et al., 2017). However, **enriching the representation of the BoW in neural topic models is still an underexplored path**. There exists work on incorporating external information, e.g., via word embeddings (Dieng et al., 2020; Gupta et al., 2019, 2020), in neural topic models, but static word embeddings do not take into consideration the context of the considered words.

Meanwhile, pre-trained language models are becoming ubiquitous in Natural Language Processing, precisely for their ability to capture context and the relationships of the words in a sentence. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), the most prominent architecture in this category, allows us to extract pre-trained word and sentence representations. Their use as input has advanced state-of-the-art performance across many

tasks. Consequently, BERT representations are used in a diverse set of NLP applications (Nozza et al., 2020; Rogers et al., 2020a). Also, topic models could benefit from the advantages that come from the use of contextualized representations.

In addition, pre-trained representations are becoming extremely popular for their transfer learning capabilities. Multilingual and multimodal models (Bianchi et al., 2021a; Radford et al., 2021; Yang et al., 2020) can provide representations that can be applied to a wide variety of tasks in a few or zero-shot fashion. For example, we may use multilingual embeddings to address cross-lingual tasks (Hu et al., 2020), including dependency parsing (Schuster et al., 2019), named entity recognition (Rahimi et al., 2019), sentiment analysis (Barnes et al., 2018), and question answering (Artetxe et al., 2020). **Traditional topic methods are language-specific and cannot be used in a transferable manner.** They rely on a fixed vocabulary specific to the training language. Therefore, currently available topic models suffer from two limitations: (i) they cannot handle unknown words by default, and (ii) they cannot easily be applied to other languages - except the one in the training data - since the vocabulary would not match. Training on several languages together, though, results in a vocabulary so vast that it creates problems with parameter size, search, and overfitting (Boyd-Graber et al., 2014). Traditional topic modeling provides methods to extract meaningful word distributions from "unstructured" text but requires language-specific bag-of-words (BoW) representations (Boyd-Graber and Blei, 2009; Jagarlamudi and Daumé, 2010).

A cross-lingual setup proves ideal for transfer learning: provided that the gist of topics is the same across languages, we can learn this gist on texts in one language and then apply it to others. This setup is zero-shot learning: we can train a model on one language and test it on several other languages to which the model had no access during training. Being able to exploit these multilingual representations in topic modeling can indeed open the field to new exciting directions.

RESEARCH QUESTIONS. This Chapter will address the following research questions:

Q5.1 How can we model context information into neural topic models?

Q5.2 How can we exploit the cross-lingual capabilities of the multilingual pre-trained representations for topic modeling?

This Chapter is organized as follows. In Section 5.1 we will present the class of contextualized topic models, a family of topic models that incorporate context information in the form of contextualized document embeddings. In Section 5.2, we will describe in detail the so-called Combined Topic Model, a contextualized topic model that

combines the input BoW representation with the corresponding contextualized document representations. We will then propose a variant of contextualized topic models, i. e., Zero-shot Topic Model, in Section 5.3. This model can exploit multilingual representations for predicting the topics of documents in unseen languages.

5.1 MODELING CONTEXTUAL INFORMATION INTO NEURAL TOPIC MODELS

We introduce the class of Contextualized Topic Model (CTM) to investigate the incorporation of contextualized representations in topic models. This class is built around two main components: (i) the neural topic model ProdLDA (Srivastava and Sutton, 2017) and (ii) Sentence BERT (SBERT) representations (Reimers and Gurevych, 2019). However, the method is agnostic about the choice of the topic model and the pre-trained representations, as long as the topic model extends an autoencoder and the pre-trained representations embed the documents.

PRODUCT-OF-EXPERTS LDA (PRODLDA). CTMs extend ProdLDA (introduced in Section 3.2). This neural variational framework trains a neural inference network to directly map the BoW document representation into a continuous latent representation. Then, a decoder network reconstructs the BoW by generating its words from the latent document representation. The framework explicitly approximates the Dirichlet prior using Gaussian distributions, instead of using a Gaussian prior like Neural Variational Document Models (Miao et al., 2016). Moreover, the authors replace the word probabilities with a weighted product of experts (Hinton, 2002). This modification allows the topic model to obtain a drastic improvement in topic coherence.

SENTENCE BERT (SBERT). The other main component that characterizes Contextualized Topic Models is the contextualized document embeddings. We use the sentence embeddings derived from SBERT (Reimers and Gurevych, 2019)[1], a recent extension of BERT that allows the quick generation of sentence embeddings. In particular, SBERT is a modification of the pretrained BERT network that uses siamese network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. SBERT adds a pooling operation to the output of BERT to derive a fixed-sized sentence embedding: we obtain the resulting sentence embedding from the computation of the mean of all the BERT's output vectors.

Although this approach allows us to quickly obtain document embeddings, it has one limitation: if a document is longer than SBERT's

1 https://github.com/UKPLab/sentence-transformers

sentence-length limit, the rest of the document will be lost. However, we will see later that encoding only the first tokens is sufficient to enrich the document BoW representation and obtain more coherent topics.

In the following sections, we will detail the two variants of Contextualized Topic Models we have defined: the Combined Topic Model (Section 5.2) and the Zero-Shot Topic Model (Section 5.3). We release Contextualized Topic Models as a Python library.[2]

5.2 COMBINED TOPIC MODELS

The first variant of Contextualized Topic Models that we propose is the so-called Combined Topic Models, which combines the input BoW document representations with the corresponding contextualized representations by concatenating the two representations. The document representations are projected through a hidden layer with the same dimensionality as the vocabulary size, concatenated with the BoW representation. The rest of the architecture remains invariant. The model will then learn to reconstruct the BoW representation given the topical representation (sampled representation) of the document. Figure 5.1 sketches the architecture of our model.

By concatenating the two representations, we encourage the documents with similar contextualized representations to be close to each other, and therefore generate better topical representations.

5.2.1 *Experimental Setting*

Our objective is to show that the CombinedTM can improve the quality of the topics, thanks to the incorporation of contextualized representations. In the following, we will present the experimental setting of our experiments.

Dataset	Docs	Vocabulary	Avg (Std) Document Length
20Newsgroups	18,173	2,000	50.21 (140.30)
Wiki20K	20,000	2,000	15.63 (4.59)
StackOverflow	16,408	2,303	5.02 (1.76)
Tweets2011	2,471	5,098	8.56 (3.17)
GoogleNews	11,108	8,110	6.23 (1.86)

Table 5.1: Statistics of the datasets used.

2 The library is available at the following link: https://github.com/MilaNLProc/contextualized-topic-models

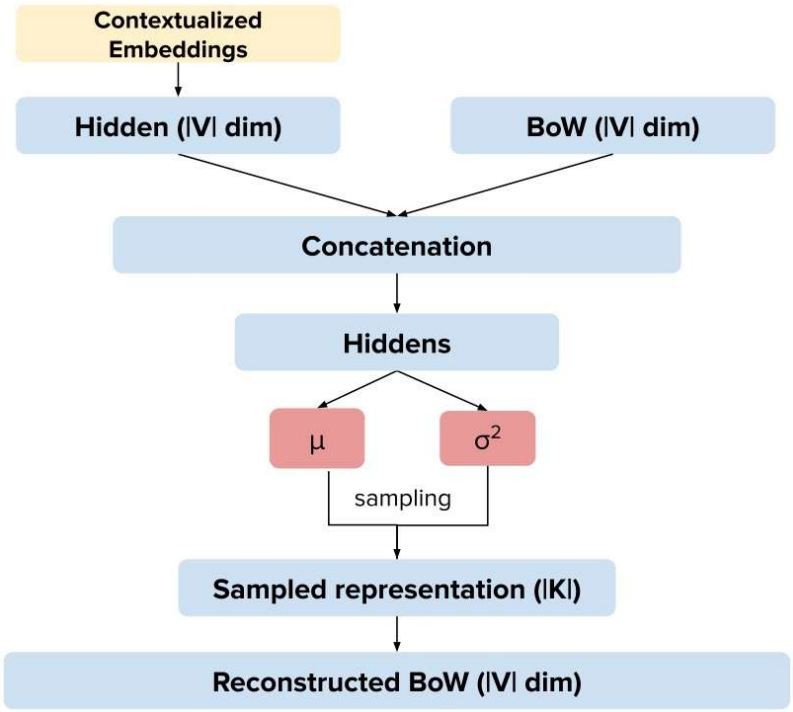


Figure 5.1: High-level schema of the architecture of CombinedTM.

DATASETS. We evaluate the models on the following five datasets: 20NewsGroups[3], Wiki20K (a collection of 20,000 English Wikipedia abstracts), Tweets2011[4], Google News (Qiang et al., 2019), and the StackOverflow dataset (Qiang et al., 2019). The latter three are already pre-processed. We use a similar pipeline for 20NewsGroups and Wiki20K: removing digits, punctuation, English stop-words, and infrequent words. We derive SBERT document representations from unpreprocessed text for Wiki20k and 20NewsGroups. For the others, we use the pre-processed text.[5] See Table 5.1 for dataset statistics. The sentence encoding model used is the pre-trained RoBERTa model fine-tuned on SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and the STSb (Cer et al., 2017) dataset.[6]

METRICS. We evaluate each model on three different metrics: two for topic coherence (NPMI) and an external word embedding-based topic coherence (WE-TC, as defined in Section 3.4.1) and the Inversed Rank-Biased Overlap metric (IRBO, Section 3.4.1) to quantify the diversity of the topic solutions.

3 http://qwone.com/~jason/20Newsgroups/
4 https://trec.nist.gov/data/tweets/
5 This can be sub-optimal, but many datasets in the literature are already pre-processed.
6 stsb-roberta-large

- **NPMI** is computed on the original corpus. As (Ding et al., 2018) pointed out, though, topic coherence computed on the original data is inherently limited. Coherence computed on an external corpus, on the other hand, correlates much more to human judgment, but it may be expensive to estimate.
- **WE-TC** provides an additional measure of how similar the words in a topic are. We follow (Ding et al., 2018) and first compute the average pairwise cosine similarity of the word embeddings of the top-10 words in a topic, using (Mikolov et al., 2013) embeddings. Then, we compute the overall average of those values for all the topics. We can consider this measure as an external topic coherence, but it is more efficient to compute than Normalized Pointwise Mutual Information on an external corpus.
- **IRBO** evaluates how diverse the topics generated by a single model are. We define IRBO as the reciprocal of the standard RBO (Terragni et al., 2021b; Webber et al., 2010). RBO compares the 10-top words of two topics. It allows disjointedness between the lists of topics (i.e., two topics can have different words in them) and uses weighted ranking. I.e., two lists that share some of the same words, albeit at different rankings, are penalized less than two lists that share the same words at the highest ranks. IRBO is 0 for identical topics and 1 for completely different topics.

All the metrics are computed on the 10-most likely words of the topics.

MODELS. Our main objective is to show that contextual information increases coherence. To show this, we compare our approach to ProdLDA (Srivastava and Sutton, 2017, the model we extend)[7], and the following models: (ii) Neural Variational Document Model (NVDM) (Miao et al., 2016); (iii) the Embedded Topic Model (ETM) (Dieng et al., 2020), MetaLDA (MLDA) (Zhao et al., 2017) and (iv) LDA. For a detailed descriptions of the neural models, we refer to Section 3.2.

HYPERPARAMETERS CONFIGURATIONS. We train all models with similar hyperparameter configurations. The inference network for both our method and ProdLDA consists of one hidden layer and 100-dimension of softplus units, which converts the input into embeddings. This final representation is again passed through a hidden layer before the variational inference process. We follow (Srivastava and Sutton, 2017) for the choice of the parameters. The priors over the topic and document distributions are learnable parameters. For LDA,

7 We use the implementation of (Carrow, 2018).

the Dirichlet priors are estimated via Expectation-Maximization. In detail:

- *ProdLDA*: We use the implementation made available by Carrow (2018) since it is the most recent and with the most updated packages (e.g., one of the latest versions of PyTorch). We run 100 epochs of the model. We use ADAM optimizer. The inference network is composed of a single hidden layer and 100-dimension of softplus units. The priors over the topic and document distributions are learnable parameters. Momentum is set to 0.99, the learning rate is set to 0.002, and we apply 20% of drop-out to the hidden document representation. The batch size is equal to 200. More details related to the architecture can be found in the original work (Srivastava and Sutton, 2017).

- *Combined TM*: The model and hyperparameters are the same used for ProdLDA with the difference that we also use SBERT features in combination with the BoW: we take the SBERT embeddings, apply a (learnable) function/dense layer $\mathrm{R}^{1024} \rightarrow \mathrm{R}^{|V|}$ and concatenate the representation to the BoW. We run 100 epochs of the model.

- *LDA*: We use Gensim's[8] implementation of this model. The hyperparameters α and β, controlling the document-topic and word-topic distribution respectively, are estimated from the data during training.

- *ETM*: We use the implementation available at `https://github.com/adjidieng/ETM` with default hyperparameters.

- *MetaLDA*: We use the authors' implementation available at `https://github.com/ethanhezhao/MetaLDA`. As suggested, we use the Glove embeddings to initialize the models.[9] The parameters α and β have been set to 0.1 and 0.01 respectively.

- *NVDM*: We use the implementation available at `https://github.com/ysmiao/nvdm` with default hyperparameters, but using two alternating epochs for encoder and decoder.

5.2.2 *Results*

We divide our results into two parts: we first describe the results for our quantitative evaluation, and we then explore the effect on the performance when we use two different contextualized representations.

8 `https://radimrehurek.com/gensim/models/ldamodel.html`

9 We used the 50-dimensional embeddings from `https://nlp.stanford.edu/projects/glove/`.

Model	NPMI	WE-TC	IRBO	NPMI	WE-TC	IRBO
	Wiki20k			20NewsGroup		
Ours	**0.1823**	0.1980	**0.9950**	0.1025	0.1715	0.9917
ProdLDA	0.1397	0.1799	0.9901	0.0632	0.1554	**0.9931**
MLDA	0.1443	**0.2110**	0.9843	**0.1300**	0.2210	0.9808
NVDM	-0.2938	0.0797	0.9604	-0.1720	0.0839	0.9805
ETM	0.0740	0.1948	0.8632	0.0766	**0.2539**	0.8642
LDA	-0.0481	0.1333	0.9931	0.0173	0.1627	0.9897
	GoogleNews			Tweets2011		
Ours	**0.1207**	**0.1325**	**0.9965**	**0.1008**	**0.1493**	0.9901
ProdLDA	0.0110	0.1218	0.9902	0.0612	0.1327	0.9847
MLDA	0.0849	0.1219	0.9959	0.0122	0.1272	**0.9956**
NVDM	-0.3767	0.1067	0.9648	-0.5105	0.0797	0.9751
ETM	-0.2770	0.1175	0.4700	-0.3613	0.1166	0.4335
LDA	-0.3250	0.0969	0.9774	-0.3227	0.1025	0.8169
	StackOverflow					
Ours	**0.0280**	0.1563	0.9805			
ProdLDA	-0.0394	0.1370	**0.9914**			
MLDA	0.0136	0.1450	0.9822			
NVDM	-0.4836	0.0985	0.8903			
ETM	-0.4132	**0.1598**	0.4788			
LDA	-0.3207	0.1063	0.8947			

Table 5.2: Averaged results over 5 numbers of topics. Best results are marked in bold.

QUANTITATIVE EVALUATION. We compute all the metrics for 25, 50, 75, 100, and 150 topics. We average results for each metric over 30 runs of each model (see Table 5.2). As a general remark, our CombinedTM provides the most coherent topics across all corpora and topic settings, even maintaining a competitive diversity of the topics. This result suggests that the incorporation of contextualized representations can improve a topic model's performance.

LDA and NVDM obtain low coherence. This result has also also been confirmed by (Srivastava and Sutton, 2017). ETM shows good external coherence (WE-TC), especially in 20NewsGroups and StackOverflow. However, it fails at obtaining a good NPMI coherence for short texts. Moreover, IRBO shows that the topics are very similar to one another. A manual inspection of the topics confirmed this problem. MetaLDA is the most competitive of the models we used for com-

Wiki20K	25	50	75	100	150
Ours	**0.17**♣	**0.19**♣	**0.18**♣	**0.19**♣	**0.17**♣
MLDA	0.15	0.15	0.14	0.14	0.13
StackOverflow					
Ours	**0.05**	**0.03**♣	**0.02**♣	**0.02**♣	**0.02**♣
MLDA	**0.05**♣	0.02	0.00	-0.02	0.00
GoogleNews					
Ours	**-0.03**♣	**0.10**♣	**0.15**♣	**0.18**♣	**0.19**♣
MLDA	-0.06	0.07	0.13	0.16	0.14
Tweets2011					
Ours	**0.05**♣	**0.10**♣	**0.11**♣	**0.12**♣	**0.12**♣
MLDA	0.00	0.05	0.06	0.04	-0.07
20NewsGroup					
Ours	0.12	0.11	0.10	0.09	0.09
MLDA	**0.13**♣	**0.13**♣	**0.13**♣	**0.13**♣	**0.12**♣

Table 5.3: Comparison of NPMI between CombinedTM (ours) and MetaLDA over various choices of topics. Each result averaged over 30 runs. ♣ indicates statistical significance of the results (t-test, p-value < 0.05).

parison. This may be due to the incorporation of pre-trained word embeddings into MetaLDA. Our model provides very competitive results, and the second strongest model appears to be MetaLDA. For this reason, we provide a detailed comparison of NPMI in Table 5.3, where we show the average coherence for each number of topics; we show that on 4 datasets over 5 our model provides the best results, but still keeps a very competitive score on 20NewsGroup, where MetaLDA is best.

USING DIFFERENT CONTEXTUALIZED REPRESENTATIONS. Contextualized representations can be generated from different models and some representations might be better than others. Indeed, one question left to answer is the impact of the specific contextualized model on the topic modeling task. To answer this question we re-run all the experiments with CombinedTM but we used different contextualized sentence embedding methods as input to the model.

We compare the performance of CombinedTM using two different models for embedding the contextualized representations found in the SBERT repository:[10] *stsb-roberta-large* (Ours-R), as employed in the

10 https://github.com/UKPLab/sentence-transformers

previous experimental setting, and using *bert-base-nli-means* (Ours-B). The latter is derived from a BERT model fine-tuned on NLI data. Table 5.4 shows the coherence of the two approaches on all the datasets (we averaged all results). In these experiments, RoBERTa fine-tuned on the STSb dataset has a strong impact on the increase of the coherence. This result suggests that including novel and better contextualized embeddings can further improve a topic model's performance.

	Wiki20K	**SO**	**GoogleNews**	**Tweets2011**	**20NewsGroup**
Ours-R	**0.18**	**0.03**	**0.12**	**0.10**	**0.10**
Ours-B	**0.18**	0.02	0.08	0.06	0.07

Table 5.4: NPMI performance of CombinedTM using different contextualized encoders.

5.3 ZERO-SHOT CONTEXTUALIZED TOPIC MODELS FOR CROSS-LINGUAL PREDICTIONS

In Section 5.2, we have combined the contextualized representations with the input BoW representation of a neural topic model. One may wonder what happens if we instead *replace the input BoW with* the contextualized embeddings.

It is true that traditional neural topic models, such as ProdLDA (Srivastava and Sutton, 2017) and NVMD (Miao et al., 2016), take in input the document BoW representations, which provide valuable symbolic information; however, this information's structure is lost after the first hidden layer in any neural architecture. We, therefore, hypothesize that contextual information can replace the input BoW representation.

Moreover, instead of using monolingual pre-trained representations as in Section 5.2, we can use multilingual representations. This additional change allows us to address the two limitations mentioned in the introduction of the chapter. In particular, (i) our approach solves the problem of dealing with unseen words at test time since we do not need them to have a BoW representation; moreover, (ii) the model infers topics on unseen documents in languages other than the one in the training data. The inferred topics consist of tokens from the training language and can be applied to any supported test language. In Figure 5.2, we sketch the architecture of our contextualized neural topic model. The final *reconstructed BoW* layer is still a component of our model: the BoW representation is necessary for the model's training to obtain the topic indicators (i.e., the most likely words representing a topic).

We refer to this model as ZeroShotTM. This model can be applied to new languages after training is complete and does not require

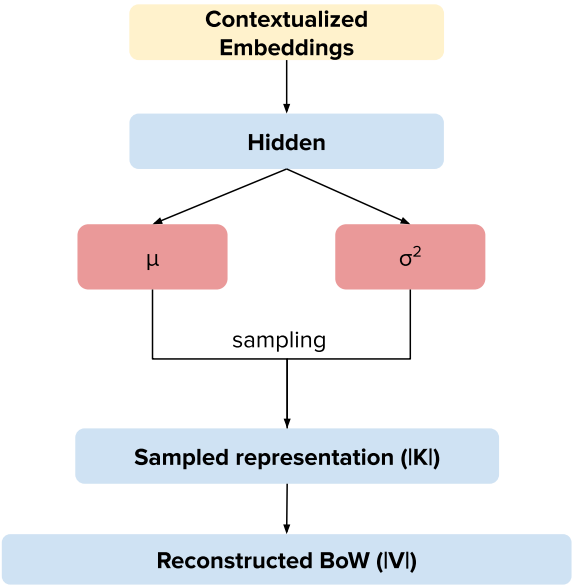


Figure 5.2: High-level schema of the architecture of ZeroShotTM.

external resources, alignment, or other conditions. Nonetheless, the flexibility of the input means our model will benefit from any future improvement of language modeling techniques. The multilingual capabilities of ZeroShotTM are extremely useful in low-resource settings in which there is little data available for the new languages. Because multilingual contextualized representations exist for multiple languages, it allows zero-shot modeling in a cross-lingual scenario. Indeed, ZeroShotTM is language-independent: given a contextualized representation of a new language as input,[11] it can predict the topic distribution of the document. The predicted topic descriptors, though, will be from the training language.

5.3.1 *Experimental Setting*

Our experiments evaluate two main hypotheses: (i) we can define a topic model that does not rely on the BoW input but instead uses contextual information; (ii) the model can tackle zero-shot cross-lingual topic modeling.

MODELS. Regarding the first setting, our main objective is to show that even if we remove the input BoW, we can keep a stable coherence. To show this, we compare our approach with CombinedTM (the model previously defined), ProdLDA (Srivastava and Sutton, 2017), and LDA.

11 As long as a multilingual model - like multilingual BERT - covers it.

DATASETS. We use datasets collected from English Wikipedia abstracts from DBpedia.[12] The first dataset (W1) contains 20,000 randomly sampled abstracts. The second dataset (W2) contains 100,000 English documents. We use 99,700 documents as training and consider the remaining 300 documents as the test set. We collect the 300 respective instances in Portuguese, Italian, French, and German. This collection creates a test set of comparable documents, i.e., documents that refer to the same entity in Wikipedia, but in different languages.

We extract only the first 200 tokens of each abstract to reduce the length limit's effects in the tokenization process. In particular, we use the SBERT embeddings using a multilingual model,[13] on this unpreprocessed text. We then remove English stop-words and use the most frequent remaining 2,000 words to create the English vocabulary for BoW model comparisons.

METRICS. Regarding the first experimental setting, we use NPMI coherence (Lau et al., 2014b) to validate the model. While regarding the quantitative evaluation of the cross-lingual experiments, we define the following metrics.

We expect the topic distributions over a set of comparable documents (e.g., in English and Portuguese) to be similar to each other. We compare the topic distributions of each abstract in a test language with the topic distribution of the respective abstract in English, which is the training language. Note that the English test document is also unseen, i.e., the training data does not include it. We evaluate our model on three different metrics. The first metric is **matches**, i.e., the percentage of times the predicted topic for the non-English test document is the same as for the respective test document in English. The higher the scores, the better.

To also account for similar but not exactly equal topic predictions, we compute the **centroid embeddings** of the five words describing the predicted topic for both English and non-English documents. Then we compute the cosine similarity between those two centroids (CD).

Finally, to capture the **distributional similarity**, we also compute the KL divergence between the predicted topic distribution on the test document and the same test document in English. Here, lower scores are better, indicating that the distributions do not differ by much.

HYPERPARAMETER SETTING.

- ProdLDA: As for the previous experiments, we use the implementation made available by (Carrow, 2018). We train the model for 100 epochs. We use ADAM optimizer (with a learning rate equal to 2e-3). The inference network is composed of a single

12 https://wiki.dbpedia.org/downloads-2016-10

13 We use the *distiluse-base-multilingual-cased* embeddings for this experiment available on the authors' repository.

hidden layer and 100-dimension of softplus units. The priors over the topic and document distributions are learnable parameters. Momentum is set to 0.99, the learning rate is set to 0.002, and we apply 20% of drop-out to the hidden document representation. The batch size is equal to 200. More details related to the architecture can be found in the original work (Srivastava and Sutton, 2017).

- ZeroShot TM: The model and the hyperparameters are the same for ProdLDA. The model is trained for 100 epochs. We use ADAM optimizer.
- Combined TM: The model and the hyperparameters are the same used for ProdLDA. The model is trained for 100 epochs. We use ADAM optimizer.
- LDA: We use Gensim's[14] implementation of this model. The hyperparameters α and β, controlling the document-topic and word-topic distribution respectively, are estimated from the data during training.

5.3.2 *Results on Hypothesis 1: Topic Quality*

First, we want to check if **ZeroShotTM** maintains comparable performance to other topic models; if this is true, we can then explore its performance in a cross-lingual setting. Since we use only English text, in this setting we use English representations.[15]

Model	NPMI (50)	NPMI (100)
ZeroShotTM	0.1632	0.1381
Combined TM	0.1544	**0.1409***
ProdLDA	**0.1658**	0.1285
LDA	-0.0246	-0.0757

Table 5.5: NPMI Coherences on W1 dataset. * denotes the statistically significant results (t-test).

We compute the topic coherence (Lau et al., 2014a) via NPMI for 50 and 100 topics averaging models' results over 30 runs for all the considered models. We report the results in Table 5.5. ZeroShotTM obtains comparable results to CombinedTM and ProdLDA in this setting. Contextualized embeddings *can* replace BoW input representations without loss of coherence.

14 https://radimrehurek.com/gensim/models/ldamodel.html
15 We use the *bert-base-nli-mean-tokens* model.

5.3.3 *Results on Hypothesis 2: Zero-shot Cross-Lingual Topic Modeling.*

ZeroShotTM can be used for zero-shot cross-lingual topic modeling. We evaluate multilingual topic predictions on the multilingual abstracts in W2. We use SBERT to generate multilingual embeddings as the input of the model.

Lang	Mat25↑	KL25↓	CD25↑	Mat50↑	KL50↓	CD50↑
IT	75.67	0.16	0.84	62.00	0.21	0.75
FR	79.00	0.14	0.86	63.33	0.19	0.77
PT	78.00	0.14	0.85	68.00	0.19	0.79
DE	79.33	0.15	0.85	64.33	0.20	0.77
Lang Avg	**78.00**	**0.15**	**0.85**	64.41	0.20	0.77
Ori Avg	76.00	**0.15**	0.84	**69.00**	**0.19**	**0.79**
Uni	4.00	0.75	—	2.00	0.85	—

Table 5.6: Match, KL, and centroid similarity for 25 and 50 topics on various languages on W2.

Quantitative Evaluation

Since the predicted document-topic distribution is subject to a stochastic sampling process, we average it over 100 samples to obtain a better estimate.

AUTOMATIC EVALUATION We use two baselines: the first one (Ori) consists of performing topic modeling on documents translated into English via DeepL.[16] Let us notice that, while this is an easily accessible baseline, automatic translation may be expensive and may introduce bias in the representations (Hovy et al., 2020). We compare the predicted topics of each translated document to the ones predicted for the original English document (as done above). The second baseline is a uniform distribution (Uni): we compute all the metrics over a uniform distribution (this baseline gives a lower bound).

Table 5.6 shows the evaluation results of our model in the zero-shot context. Note that because we trained on English data, the topic descriptors are in English. Topic predictions are significantly better than the uniform baselines: more than 70% of the times, the predicted topic on the test set matches the topic of the same document in English. The CD similarity suggests that even when there is no match, the predicted topic on the unseen language is at least similar to the one on the English testing data. Simultaneously, the predictions for

16 https://www.deepl.com/

the contextualized model are in line with the ones obtained using the translations (Ori Avg), showing that our model is capable of finding good topics for documents in unseen languages without the need for translation.

HUMAN EVALUATION. We rated the predicted topics for 300 test documents in five languages (thus, 1500 docs including English) on an ordinal scale from 0-3. A 0 rate means that the predicted topic is wrong, a 1 rate means the topic is somewhat related, a 2 rate means the topic is good, and a 3 rate means the topic is entirely associated with the considered document.

Language	**Average Topic Quality**
English	2.35
Italian	2.29
French	2.22
Portuguese	2.26
German	2.19
Average	2.26

Table 5.7: Average topic quality (out of 3).

Table 5.7 shows the results per language. We evaluate the inter-rater reliability using Gwet AC1 with ordinal weighting (Gwet, 2014). The resulting value of 0.88 indicates consistent scoring.

Qualitative Evaluation

In Table 5.8, we show some examples of topic predictions on test languages. Our model predicts the main topic for all languages, even though they were unseen during training.

The predicted topic is generally consistent with the text. I.e., the topics are easily interpretable and give the user a coherent impression. In some circumstances, noise biases the results: dates in the abstract tend to make the model predict a topic about time. Another interesting case is the abstract of the artist Joan Brossa, who was both a poet and a graphic designer. In the English and Italian abstract, the model has discovered a topic related to writing. In contrast, in the Portuguese abstract, the model has found a topic related to art, which is still meaningful.

5.4 SUMMARY OF THIS CHAPTER

In the following, we give a short summary of this chapter. In the introduction of the Chapter, we started with the following research questions:

Q5.1 How can we model context information into neural topic models?

Q5.2 How can we exploit the cross-lingual capabilities of the multilingual pre-trained representations for topic modeling?

To answer Question Q5.1, we have defined the class of Contextualized topic models, a family of models that incorporates context information in the form of pre-trained contextualized embeddings. These models guarantee an improvement in the coherence with respect to state-of-the-art topic models. Moreover, results that compare different types of contextualized representations suggest that including novel and better contextualized embeddings can further improve a topic model's performance.

Concerning question Q5.2, we have shown that ZeroShotTM, a component of the family of CTMs that replaces the input BoW representations with contextualized representations, can exploit multilingual embeddings to address the task of cross-lingual topic modeling. The model can indeed be trained on a corpus in a language and then can predict the topics of documents in unseen languages.

These results pave the way to different research directions. This model can be applied in low-resource settings, when we aim to predict the topics of a set of documents but the documents in the considered language are very few. The proposed model is also language-independent: we can obtain the topics of documents without the need to know or understand the target language. In addition, our results suggest us that we can exploit the transfer learning capabilities of document embeddings to other contexts: for example, we could use multimodal representations, e.g. Image-Text representations (Radford et al., 2021), to learn topics of a corpus of documents and zero-shot predict the topics of instances of another modality (e.g. images).

Lang	Sentence	Predicted Topic
EN	Blackmore's Night is a British/American traditional folk rock duo [...]	rock, band, bass, formed
IT	I Blackmore's Night sono la band fondatrice del renaissance rock [...]	rock, band, bass, formed
PT	Blackmore's Night é uma banda de folk rock de estilo renascentista [...]	rock, band, bass, formed
EN	Langton's ant is a two-dimensional Turing machine with [...]	mathematics, theory, space, numbers
FR	On nomme fourmi de Langton un automate cellulaire [...]	mathematics, theory, space, numbers
DE	Die Ameise ist eine Turingmaschine mit einem zweidimensionalen [...]	mathematics, theory, space, numbers
EN	The Journal of Organic Chemistry, colloquially known as JOC or [...]	journal, published, articles, editor
IT	Journal of Organic Chemistry è una rivista accademica [...]	journal, published, articles, editor
PT	Journal of Organic Chemistry é uma publicação científica [...]	journal, published, articles, editor
EN	The Pirate Party Germany (German: Piratenpartei Deutschland) [...]	political, movement, party, alliance
PT	Piratenpartei Deutschland (Partido Pirata da Alemanha, [...]	political, movement, party, alliance
DE	Die Piratenpartei Deutschland (Kurzbezeichnung Piraten, [...]	political, movement, party, alliance
EN	Joan Brossa [...] was a Catalan poet, playwright, graphic designer [...]	book, french, novel, written
IT	Fu l'ispiratore e uno dei fondatori della rivista "Dau al Set"[...]	book, french, novel, written
PT	Joan Brossa i Cuervo [...] foi um poeta, dramaturgo, artista plástico [...]	painting, art, painter, works

Table 5.8: Examples of zero-shot cross-lingual topic classification in various languages with ZeroShotTM.

Part III

TOPIC MODELS' EVALUATION

6

HYPERPARAMETER OPTIMIZATION FOR THE COMPARISON OF TOPIC MODELS

Although topic models are used in a vast range of applications, from text exploratory purposes to information retrieval tasks (Boyd-Graber et al., 2017), most of the investigations disregard the main elements that influence the results generated by the models and, in particular, what is their effect on the performance. Several works explore topic modeling over a range of different models, topics, and measures, but usually focus on classical topic models (Greene et al., 2014; Stevens et al., 2012), e.g. Latent Dirichlet Allocation (Blei et al., 2003b), and solely on a single evaluation measure (O'Callaghan et al., 2015; Stevens et al., 2012). Doan and Hoang (2021) recently made an effort to benchmark neural topic models, however, they seem to disregard the importance of the hyperparameter selection.

In fact, the evaluations of topic models are usually limited to the comparison of models whose hyperparameters are fixed. Yet, the hyperparameters that control the models can have a great impact on their performance. Therefore, fixing them prevents researchers from discovering the best topic model on a given dataset. In the latest years, Neural Topic Models (NTM) (Dieng et al., 2020; Zhao et al., 2021) have gained popularity. The problem of finding the best hyperparameter configuration has become even more compelling, since topic models based on neural networks are usually controlled by a high number of hyperparameters. **It is then critical to carefully select the hyperparameters by adopting a search strategy that is computationally tractable and effective from a quantitative and qualitative perspective.**

To this end, Bayesian Optimization (Archetti and Candelieri, 2019, BO) seems to be an excellent solution to discover an optimal set of hyperparameters for a topic model. This method for finding the global optimum of expensive objective functions assumes that the function is unknown (also called "black-box"). We can indeed express a model just in terms of its inputs, hyperparameters and output, and this allows us to use Bayesian Optimization for any type of hyperparameters (binary, categoricals, or continuous) and objective function.

In addition, **exploring different hyperparameter configurations allows us to empirically investigate the relationships among the different metrics, hyperparameters, models and datasets.** For example, we know that the Dirichlet prior over the document-topic distribution in LDA controls the sparsity of the distribution. Tackling this hyperparameter will lead to document representations dominated by

a few high-peaked topics or otherwise dominated by many topics but less likely. This can have an impact on the performance of downstream tasks which use the topical representations of the documents as features. We also expect that varying the value of a hyperparameter will have different effects on datasets characterized by short texts or long texts, and it will also have an effect on other performance metrics. For example, a topic model that produces good document representations with a given hyperparameter value may not be able to produce coherent and diverse topics simultaneously.

RESEARCH QUESTIONS. In this chapter, we will therefore address the following research questions:

Q6.1 Can we determine if a topic model can guarantee an optimal trade-off between different performance measures?

Q6.2 Can a performance measure imply a competing or correlated target for other performance measures?

This Chapter is therefore organized as follows; in Section 6.1 we present how we can apply Bayesian Optimization to solve the problem of hyperparameter tuning in topic models. In Section 6.2 we present the comparative framework OCTIS 1.0. We will then show different directions of the use of Bayesian Optimization through a comparative analysis of classical topic models in Section 6.3 and of neural topic models in Section 6.4. Finally, we will conclude the Chapter with some remarks and future directions in Section 6.5.

6.1 BAYESIAN OPTIMIZATION FOR TOPIC MODELING

The hyperparameters are fundamental ingredients in topic models. Considering that the hyperparameter configuration of a topic model can have a strong effect not only on the prior distribution of the parameters in the model, but also on their posterior distribution (George et al., 2017), it is important to choose them carefully.

Bayesian Optimization is an excellent fit for our requirements. BO is suitable for expensive and noisy objective functions (such as topic models). Moreover, BO treats the objective function as a black box, a system or function solely viewed in terms of its inputs and outputs and whose internal workings are invisible. In the case of topic models, the black box takes as input a dataset and a set of hyperparameters values and returns the score of the chosen objective function (e.g. topic coherence), computed on the output of the topic model (i.e. the top-t topic words the document-topic distributions, and the topic-word distribution). This means that we do not have to care of the shape or features of the objective function, but instead, a black-box approach generalizes to different objective functions.

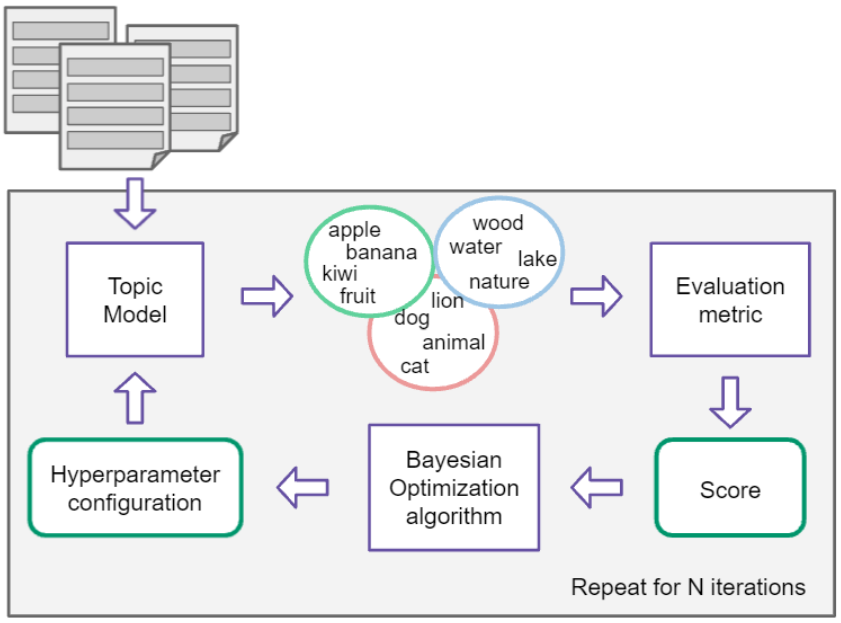


Figure 6.1: Illustration of the Bayesian Optimization process applied to topic modeling.

Figure 6.1 depicts the process of Bayesian Optimization applied to topic models. As mentioned before, the topic model takes as input the document corpus and a proposed hyperparameter configuration, and it returns the topics. This output will be used to compute the score of the chosen objective function and the score will be used from BO (along with all the past evaluation scores) to propose the next hyperparameter configuration to evaluate.

The reader can refer to the pseudo-code reported in Table 2.1 for a reference of the BO algorithm, and we refer to Section 2.5.3 for additional details on Bayesian Optimization.

6.2 OCTIS: OPTIMIZING AND COMPARING TOPIC MODELS IS SIMPLE!

In this section we present OCTIS (Optimizing and Comparing Topic models Is Simple)[1], a unified and open-source evaluation framework for training, analyzing, and comparing topic models, over several datasets and evaluation metrics. To guarantee a fair comparison among the models, we find the optimal hyperparameter configuration of the models according to a Bayesian Optimization (BO) strategy, as explained in Section 6.1.

1 A video demonstration is also available at https://youtu.be/nPmiWBFFJ8E.

6.2.1 *System design and architecture*

The proposed framework follows an object-oriented paradigm, providing all the tools for running a whole topic modeling pipeline. The main functionalities of the proposed OCTIS are related to dataset pre-processing, training topic models, estimating evaluation metrics, hyperparameter optimization, and interactive web dashboard visualization. Figure 6.2 summarizes the workflow involving the first four modules (the dashboard interacts with all of them).

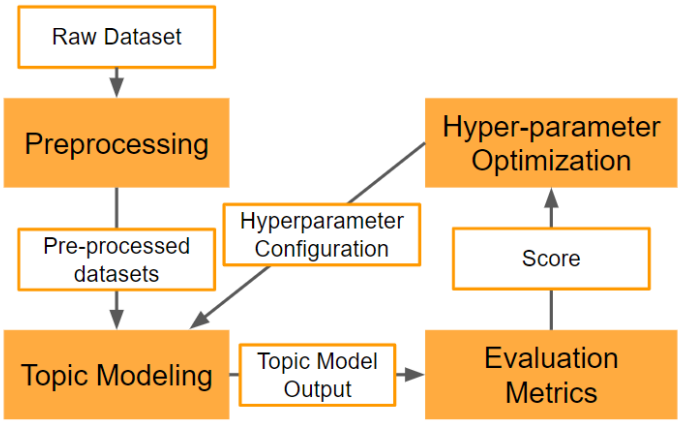


Figure 6.2: Workflow of the OCTIS framework.

The framework can be used both as a python library and as a dashboard. The python library offers more advanced functionalities than the ones available in the dashboard. The modules that comprise the OCTIS framework are detailed in the following sections.

DATASETS AND PRE-PROCESSING The first step of the topic modeling pipeline is the pre-processing of the input dataset. OCTIS includes the following pre-processing utilities:

- reducing the text to lowercase;
- punctuation removal;
- lemmatization;
- stop-words removal;
- removal of unfrequent and most frequent words (according to a specified word frequency or document frequency threshold);
- removal of documents with few words (according to a specified word frequency or document frequency threshold).

These utilities include the most common techniques for pre-processing text for topic modeling. However, some of these features may not

be appropriate for specific domains and languages, e.g. requiring language-specific or domain-specific stop-words.

OCTIS currently provides the following pre-processed datasets: 20 NewsGroups [2], M10 (Lim and Buntine, 2015), DBLP [3] and BBC News (Greene and Cunningham, 2006). Moreover, we build and include two Italian datasets from the Italian version of the Europarl dataset[4] and from the Italian abstracts of DBPedia.[5] In particular, to build these datasets we randomly sample 5000 documents from Europarl and we randomly sample 1000 Italian abstracts for 5 DBpedia types (event, organization, place, person, work), for a total of 5000 abstracts.

We report the statistics of the datasets in Table 6.1. Following the original paper, we split the datasets into three partitions: training (75%), validation (15%), and testing (15%).

Dataset	Domain	Language	# Docs	Avg. (Std.) # words in docs	# Unique words
20 Newsgroups	Forum posts	English	16309	48.02	1612
BBC News	News	English	2225	120.12	2949
M10	Scientific papers	English	8355	5.91	1696
DBLP	Scientific papers	English	54595	5.4	1513
DBPedia	Abstracts	Italian	4251	5.48 (11.76)	2047
Europarl	Proceedings	Italian	3616	20.63 (19.33)	2000

Table 6.1: Statistics of the pre-processed datasets.

The datasets already available in OCTIS, and accessible through the web dashboard, have been pre-processed according to the length and features of the documents. In particular, we removed the punctuation, we lemmatized the text and filtered out the stop-words. We removed the words that have a word frequency less than 0.5% for 20 Newsgroups and BBC News and less than 0.05% for DBLP and M10. We removed the words with a document frequency higher than the 50% and less than the 0.1% for Europarl and 0.2% for DBPedia. Subsequently, we removed the documents with less than 5 words for 20 Newsgroups, BBC News, Europarl and DBpedia and less than 3 words for the other datasets (M10 and DBLP).

Although OCTIS already provides some datasets, a user can upload and pre-process any dataset using the python library according to their needs.

2 http://people.csail.mit.edu/jrennie/20Newsgroups/
3 https://github.com/shiruipan/TriDNR/tree/master/data
4 https://www.statmt.org/europarl/
5 https://www.dbpedia.org/resources/ontology/

TOPIC MODELS OCTIS integrates both classical topic models and neural topic models. In particular, the following traditional and neural approaches are available to be trained, optimized, and compared (the models that are available in the web dashboard are marked with ⋆):

- Latent Dirichlet Allocation⋆ (Blei et al., 2003b, LDA);[6]
- Non-negative Matrix Factorization⋆ (Lee and Seung, 2000, NMF);[6]
- Latent Semantic Analysis⋆ (Hofmann, 1999, LSI);[6]
- Hierarchical Dirichlet Process (Teh et al., 2004, HDP);[6]
- Neural LDA⋆ (Srivastava and Sutton, 2017);[7]
- Product-of-Experts LDA⋆ (Srivastava and Sutton, 2017, ProdLDA);[7]
- Embedded Topic Models⋆ (Dieng et al., 2020, ETM);[8]
- Contextualized Topic Models (Bianchi et al., 2021b,c, CTM).[9]

Moreover, we defined a standard interface for allowing a user to integrate their topic model's implementation. A topic model viewed as a black box: it takes as input a dataset and a set of hyperparameters values and returns the top-t topic words, the document-topic distributions, and the topic-word distribution in a specified format.

EVALUATION METRICS. The proposed framework provides several evaluation metrics. A metric can be used as the objective targeted by a Bayesian Optimization strategy, or to monitor the behavior of a topic model while the model is optimized on a different objective. The performance of a topic model can be evaluated by investigating different aspects, according to an evaluation metrics. The available evaluation metrics include topic coherence (6), topic significance (3), topic diversity (6), topic similarity (7) and classification metrics (4), for a total of 26 evaluation metrics. In Appendix B, we propose novel topic similarity measures based on word embeddings.

HYPERPARAMETER OPTIMIZATION. OCTIS uses Bayesian Optimization to tune the hyperparameters of the topic models. If any of the available hyperparameters is selected to be optimized for a given evaluation metric, BO explores the search space to determine the optimal settings. Since the performance estimated by the evaluation metrics can be affected by noise, the objective function is computed as the median of a given number of *model runs* (i.e., topic models run with

6 https://radimrehurek.com/gensim/
7 https://github.com/estebandito22/PyTorchAVITM
8 https://github.com/adjidieng/ETM
9 https://github.com/MilaNLProc/contextualized-topic-models

Features	OCTIS	Gensim	STTM	PyCARET	MALLET	TOMODAPI
Pre-processing tools	√	√		√	√	√
Pre-processed datasets	√	√	√	√		√
Classical topic models	√	√	√	√	√	√
Neural topic models	√					√
Coherence metrics	√	√	√	√	√	
Diversity metrics	√					
Significance metrics	√					
Classification metrics	√		√	√	√	√
Hyper-parameters tuning	BO	MLE		grid-search	MLE	
Usage	script, web dashboard	script	command line	script	command line	script, API
Programming Language	Python	Python	Java	Python	Java	Python

Table 6.2: Comparison between OCTIS and the most well-known topic modeling libraries.

the same hyperparameter configuration) computed for the selected evaluation metric.

We integrated into OCTIS most of the BO algorithms of the Scikit-Optimize library (Head et al., 2018) to provide a robust and efficient BO implementation. We integrated Gaussian Process and Random Forest as surrogate models, while we included Probability of Improvement, Expected Improvement, and Upper Confidence Bound as acquisition functions (Candelieri and Archetti, 2019; Frazier, 2018).

Instead of performing BO, a user can also use a random search technique to find the best hyperparameter configuration. Since the Bayesian Optimization requires some initial configurations to fit the surrogate model, the user can provide the initial configurations, according to their domain knowledge. Alternatively, a user can perform a pure exploration of the search space using a random sampling strategy. Different algorithms are available (e.g. Uniform Random Sampling or Latin Hypercube sequence) for sampling the initial configurations.

6.2.2 *Existing frameworks*

The existing topic modeling frameworks usually provide topic modeling algorithms, while disregarding other essential aspects of the whole topic modeling pipeline: pre-processing, evaluation, compari-

son, and visualization of the results and, most importantly, the hyperparameter selection. In the following, we outline the existing frameworks, highlighting their advantages and limitations. Table 6.2 summarizes the main features of the existing topic modeling frameworks and compares them with OCTIS.

MALLET(McCallum, 2002) and gensim[6] are the most known topic modeling libraries and include several classical topic models. They provide pre-processing methods and the estimation of the hyperparameters using maximum likelihood estimation (MLE) techniques. These libraries do not include the recently proposed neural topic models, and they just provide topic coherence metrics. STTM (Qiang et al., 2018) is a java library that provides a set of topic models that are specifically designed for short texts, providing several evaluation metrics.

ToModAPI (Lisena et al., 2020) is a python API that allows for training, inference, and evaluating different topic models, also including some of the most recent. However, it does not provide a method for finding the best hyperparameter configuration of topic models. Instead, a tool that allows for optimizing the hyperparameter of a machine learning model is PyCARET (Ali, 2020). However, it employs a grid-search technique to tune the hyperparameters. This approach can be very time-consuming if the number of hyperparameters is high and the search space is huge (Bergstra and Bengio, 2012).

OCTIS stands at the union of the features of the existing frameworks: we integrated both classical and recent neural topic models, providing pre-processing methods, evaluation metrics, and the possibility of optimizing the hyperparameters. Finally, a user-friendly graphical interface to launch one or more hyperparameter optimization experiments on a given topic model and on a specific dataset has been provided.

6.2.3 *System usage*

OCTIS has been designed to be used as a python library by advanced users, as well as a simple web dashboard by anyone.

EXAMPLE OF A USE CASE FOR THE PYTHON LIBRARY. The lines of code below executes an optimization experiment that will provide an optimal configuration of the hyperparameters α and β for LDA with 25 topics by maximizing the diversity of the topics.

```
# loading of a pre-processed dataset
dataset = Dataset()
dataset.load("path/to/dataset")

#model instantiation
lda = LDA(num_topics=25)
```

```
#definition of the metric
td = TopicDiversity()

#definition of the search space
search_space = {
  "eta": Real(low=0.01, high=5.0),
  "alpha": Real(low=0.01, high=5.0)
}

#define and launch optimization
optimizer=Optimizer()
opt_result = optimizer.optimize(model, dataset, td, search_space)
```

6.2.4 *Web-based dashboard*

The dashboard includes a set of simple but useful operations to conduct an experimental campaign on different topic models. Here we briefly explain the four main functionalities of the dashboard.

EXPERIMENT CREATION. First, a user can define an optimization experiment by selecting the dataset, the topic model, the corresponding hyperparameter to optimize, the evaluation metric to be considered by the BO (possibly other extra metrics to evaluate), and the settings of the optimization process.

MANAGEMENT OF THE EXPERIMENTS' QUEUE. The user can monitor the queue of the experiments and see the corresponding progress. The user can also pause, restart, or delete an experiment that has been launched before. Additionally, the user can easily change the order of the queue of the experiments, by allowing a given run to be executed before others.

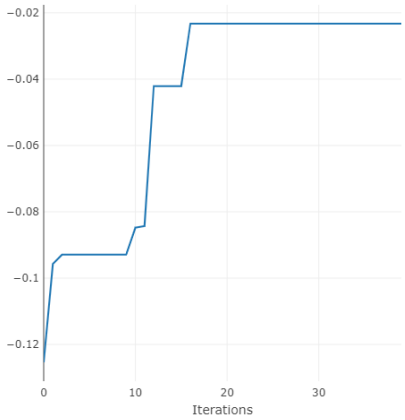


Figure 6.3: Example of the best-seen evolution for an optimization experiment.

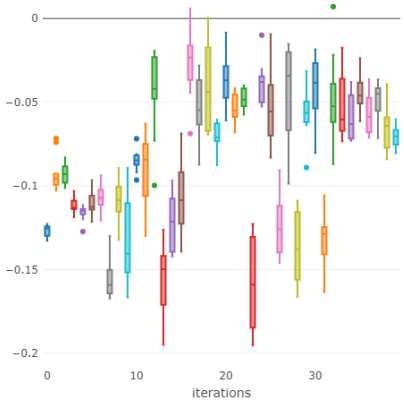


Figure 6.4: Example of box plot of an optimization experiment.

COMPARISON OF THE TOPIC MODELS. The user can select the models to be analyzed and compared. At the first stage, one can observe the progress of the BO iterations, observing a plot that contains at each iteration the best-seen evaluation, i.e. the median at each iteration of the metric that has been optimized (see Figure 6.3). Alternatively, a user can visualize a box plot at each iteration (see Figure 6.4) to understand if a given hyperparameter configuration is noisy (high variance) or not.

Figure 6.5: Example of word cloud of a topic.

ANALYSIS OF A SINGLE EXPERIMENT. A user can further inspect the results of a specific topic model on a given dataset with respect to the considered metrics, by analyzing a single experiment.

Here, a user can visualize all the information and statistics related to the experiment, including the best hyperparameter configuration and the best value of the optimized metric. They can also have an outline of the statistics of the other extra metrics that they had chosen to evaluate.

We provide three different plots for inspecting the output of a single run of a topic model. Figure 6.5 shows the word cloud obtained from the most relevant words of a given topic, scaled by their probability. Focusing on the distributions inferred by a topic model, Figure 6.6 shows the topic distribution of a document, and Figure 6.7

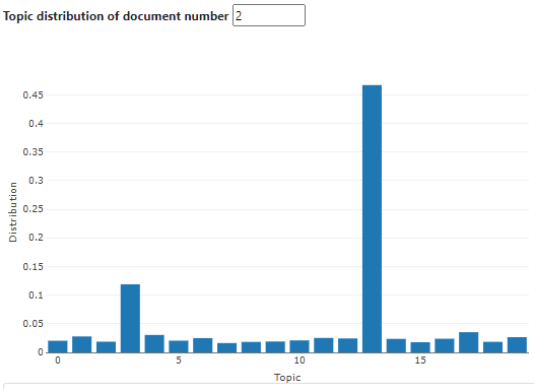


Figure 6.6: Example of distribution of the topics in a selected document.

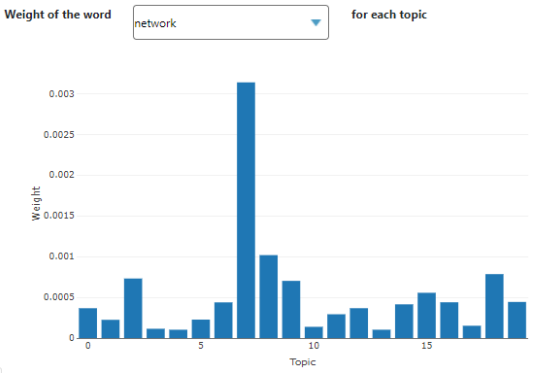


Figure 6.7: Example of the weight of the word “network" for each document.

represents an example of the weight of a selected word of the vocabulary for each topic.

6.3 COMPARATIVE ANALYSIS OF RELATIONAL AND SEMI-SUPERVISED TOPIC MODELS

In this Section, we will show how we can exploit Bayesian Optimization to draw empirical observations on the performance of the models and evaluation metrics. In particular, we want to investigate if an optimal configuration of hyperparameters enables the models to simultaneously obtain good performance in terms of prediction capabilities and significant topics from a qualitative perspective.

We focus our investigation on optimal hyperparameter settings related to a set of the most recent semi-supervised and document-level relational topic models, previously defined in Chapters 3 and 4. As previously seen, these approaches jointly model a network of documents (e.g. citations) and additional knowledge (e.g. labels) to address a document classification task. To investigate the problem of hyperparameters selection, we use a Bayesian Optimization (BO) approach.

We therefore consider the following models:

- D-CRTM-N, defined in Section 4.2;
- D-CRTM-U, defined in Section 4.2;
- Bi-RTM, defined in Section 4.2;
- LLDA (Yang et al., 2015c), defined in Section 3.1.2;
- RTM (Chang and Blei, 2009), defined in Section 3.1.2;
- WSB-RTM (Yang et al., 2016a), defined in Section 3.1.2;
- LDA (Blei et al., 2003b);

We recall that D-CRTM-N, D-CRTM-U, and Bi-RTM are extensions of RTM that can incorporate some prior knowledge related to the documents' labels. In D-CRTM-N and D-CRTM-U, prior knowledge is incorporated using constraints between documents: if two documents share the same label, then the documents are related and are more likely to share the same topics. While Bi-RTM uses an additional binary variable to model the idea that two documents that share the same label are more likely to share the same topics. We also consider an additional extension of RTM, known as WBS-RTM (Yang et al., 2016a). This model uses the weighted stochastic block model (Aicher et al., 2015) to identify blocks, i.e. subnetworks, in which documents are densely connected.

We consider LDA (Blei et al., 2003a) as a fully unsupervised baseline, RTM as a baseline that models only the network structure, and Labeled-LDA (LLDA) (Yang et al., 2015c) as a baseline that takes only the labels into account.

HYPERPARAMETERS OPTIMIZATION. To validate the hypothesis that optimal settings lead topic models to simultaneously obtain good performance in terms of prediction capabilities and coherent topics, a single-objective BO approach has been adopted to optimize the hyperparameters.

All the topic models' hyperparameters have been optimized via BO, having a Random Forest (RF) as a probabilistic surrogate model and Upper Confidence Bound (UCB) as acquisition function, using the mean of the RF and the confidence (i.e. its standard deviation) to select the next configuration.

$$\mathrm{UCB}_m(x) = \mathrm{E}[x]_m + \varphi_m^{1/2}\mathrm{S}[x]_m \tag{55}$$

where $\mathrm{E}[x]_m$ and $\mathrm{S}[x]_m$ are, respectively, the mean and the standard deviation of the prediction provided by the probabilistic surrogate model after m evaluated configurations of the topic model under optimization. The next configuration to evaluate is the one maximizing UCB_m.

In our comparative evaluation, the performance metric to be optimized is the median of micro-F1 over 30 runs for each given model. The number of initial configurations, used to fit the initial surrogate model, is given by the initial configurations reported in Section 4.2 plus 4 randomly sampled via Latin Hypercube Sampling for each topic model, a statistical method that improves on the coverage of the search space.

In Table 6.3, we report a summary of the models, specifying if they consider the network structure (N) and/or they incorporate label-related knowledge (L), together with the hyperparameters that we consider in our evaluation.

Table 6.3: Summary of the models and hyperparameters.

Models	N	L	Hyperparameters
LDA (Blei et al., 2003a)			α, β
LLDA (Yang et al., 2015b)		✓	α, β, p
RTM (Chang and Blei, 2009)	✓		α, β
WSB-RTM (Yang et al., 2016a)	✓		$\alpha, \beta, \alpha', a, b, \gamma$, blocks
Bi-RTM (Terragni et al., 2020)	✓	✓	α, β, p
CRTM-N (Terragni et al., 2020)	✓	✓	α, β, p
D-CRTM-U (Terragni et al., 2020)	✓	✓	$\alpha, \beta, p, \lambda$

We remark that α is a Dirichlet parameter controlling how the topics are distributed over a document and, analogously, β is a Dirichlet parameter controlling how the words of the vocabulary are distributed in a topic. The parameter p represents the quantity of prior knowledge, expressed as a percentage. We follow the procedure described in (Terragni et al., 2020) to define the quantity of prior knowledge. Finally, λ is an integer hyperparameter, that controls the strength of the constraints introduced into D-CRTM-U. We set the value of α to vary between 0 and $50/K$ (where K is the number of topics), following (Griffiths and Steyvers, 2004). We let β range between 0 and 1, and we let p range between 0 and 0.5. Finally, λ ranges between 1 and the average document length of the documents of the given dataset.

6.3.1 *Experimental Setting*

BENCHMARK DATASETS We consider the same datasets previously described for the experimental campaign of the document constrained relational topic models in Section 4.2. Table 6.4 shows the statistics about the selected benchmarks.

Following previous work, each dataset has been divided into a training set and a test set. We train a linear Support Vector Machine (SVM) to predict the document's class using the document-topic distribution θ of each document as its K-dimensional representation.

Dataset	#total docs	#training docs	#testing docs	#total links	#training link	#testing link	#classes	#unique words
Cora	2708	2031	677	3448	3054	394	7	1752
M10	4427	2966	1461	3057	2425	632	9	1592
WebKB	877	612	265	1516	1006	510	5	1830

Table 6.4: Statistics of the benchmark datasets.

PERFORMANCE MEASURES The evaluation metric for this task is the **Micro-F1**, defined as the weighted average of the f-measure for each class. For determining the interpretability of the extracted topics, we consider five additional metrics: KL-B, KL-U, KL-V, NPMI coherence and PUW diversity (as defined in Section 3.4).

6.3.2 *Experimental Results*

HYPERPARAMETER OPTIMIZATION DRIVEN BY MICRO-F1 In Table 6.5, we report the Micro-F1 results by comparing the best initial configuration and the optimal configuration identified by BO, also reporting the corresponding 95% confidence intervals. We show the median of Micro-F1 over 30 different runs, as it is the target performance metric to optimize. The leftmost columns of the table report the Micro-F1 values, both in terms of initial mean (μ), final mean (μ^*), initial median ($\widetilde{\mu}$), and final median ($\widetilde{\mu}^*$). We also report in Table 6.6 the hyperparameter configurations discovered by the single-objective BO approach for each model and dataset.

As expected, we can observe in Table 6.5 that, in most of the cases, a global optimization search strategy improves the results in the context of semi-supervised relational topic models for a classification task. For the Cora and M10 datasets, all the models obtain an improvement that mostly ranges from 1% to 7% for $\widetilde{\mu}^*$ with respect to $\widetilde{\mu}$. Concerning the hyperparameters α and β, it seems that lower values are often preferred. In a classification task, we expect that the probability θ has a very skewed shape, where the most probable topic will then reflect the class to predict. With low values of α, the documents are expected to follow a distribution that has few but high peaks. Analogously, word-topic distributions that are skewed towards few relevant words are preferred.

For the algorithms characterized by two hyperparameters, specifically RTM and LDA, we report a 3-dimensional representation of the Micro-F1, as approximated by a Gaussian Process regression model, with Squared Exponential kernel, fitted on the set of evaluated hyperparameters configurations. We have used a GP, instead of the RF, just to have a smoother approximation of Micro-F1 to depict. A 3-dimensional representation (Figures 6.8, 6.9 and 6.10) is reported for each dataset, and separately for LDA and RTM. As the main result,

		Micro-F1			
Dataset	Model	μ	μ^*	$\widetilde{\mu}$	$\widetilde{\mu^*}$
Cora	D-CRTM-U	0.6144 ± 0.0060	**0.6780** ± 0.0167	0.6152	**0.6928**
	CRTM-N	0.6702 ± 0.0103	**0.6882** ± 0.0145	0.6765	**0.7061**
	Bi-RTM	0.6220 ± 0.0056	**0.6228** ± 0.0083	0.6196	**0.6278**
	WSB-RTM	0.6108 ± 0.0176	**0.6213** ± 0.0093	0.6108	**0.6256**
	LLDA	0.6517 ± 0.0062	**0.6648** ± 0.0046	0.6529	**0.6669**
	RTM	0.6158 ± 0.0057	**0.6235** ± 0.0062	0.6160	**0.6256**
	LDA	0.5498 ± 0.0055	**0.5966** ± 0.0078	0.5510	**0.6012**
M10	D-CRTM-U	0.6840 ± 0.0035	**0.7194** ± 0.0089	0.6858	**0.7238**
	CRTM-N	0.7164 ± 0.0030	**0.7285** ± 0.0238	0.7139	**0.7673**
	Bi-RTM	0.7251 ± 0.0077	**0.7347** ± 0.0174	0.7286	**0.7536**
	WSB-RTM	0.3483 ± 0.0184	**0.6810** ± 0.0059	0.3737	**0.6851**
	LLDA	0.4317 ± 0.0453	**0.4968** ± 0.0347	0.5212	**0.5359**
	RTM	0.6236 ± 0.0070	**0.6805** ± 0.0033	0.6280	**0.6797**
	LDA	0.5330 ± 0.0039	**0.5915** ± 0.0057	0.5339	**0.5941**
WebKB	D-CRTM-U	0.7577 ± 0.0050	**0.7585** ± 0.0049	0.7585	0.7585
	D-CRTM-N	0.7652 ± 0.0059	**0.7658** ± 0.0061	0.7623	**0.7698**
	Bi-RTM	0.7546 ± 0.0060	**0.7600** ± 0.0054	0.7547	**0.7604**
	WSB-RTM	0.7125 ± 0.0051	**0.7453** ± 0.0061	0.71320	**0.7472**
	LLDA	0.7491 ± 0.0047	**0.7492** ± 0.0059	0.7472	**0.7509**
	RTM	0.7489 ± 0.0048	**0.7600** ± 0.0061	0.7509	**0.7585**
	LDA	0.74201 ± 0.0035	**0.7532** ± 0.0064	0.7434	**0.7547**

Table 6.5: Document classification results in terms of Micro-F1. μ and μ^* denote the initial and the final mean of the Micro-F1. $\widetilde{\mu}$ and $\widetilde{\mu}^*$ denote the initial and the final median of the Micro-F1. Bold values denote the best results both for the mean and median Micro-F1 measure.

we can observe that Micro-F1 does not depend on β. Moreover, it is worth noticing that the shape of LDA and RTM for the datasets Cora and M10 are quite similar, while for the WEBKB dataset we notice a different behavior. As we can also observe in Table 6.7, the models obtain the highest performance on the dataset WEBKB with α values higher than 1.

TOPIC ANALYSIS Concerning the topic quality, we analyze the performance of the considered models over the three datasets in Table 6.7, using the qualitative metrics previously presented (NPMI, PUW, KL-U, KL-V and KL-B). We report the results of the initial configuration (I) and final configuration (F) for each model. Focusing on the scores of the KL-B metric, we can first notice that in most of the cases the best configuration identified by BO leads the models to obtain better performance with respect to its initial counterpart. This improvement is likely related to the target performance of BO: if we

Dataset	Model	α	β	λ	p	α′	a	b	γ	blocks
Cora	D-CRTM-U	1.353	0.082	48	0.308	-	-	-	-	-
	D-CRTM-N	0.345	0.062	-	0.155	-	-	-	-	-
	Bi-RTM	0.253	0.247	-	0.246	-	-	-	-	-
	WSB-RTM	0.020	0.517	-	-	9.982	8.967	0.302	3.569	42
	LLDA	0.015	0.669	-	0.467	-	-	-	-	-
	RTM	0.218	0.496	-	-	-	-	-	-	-
	LDA	0.043	0.944	-	-	-	-	-	-	-
M10	D-CRTM-U	0.107	0.641	5	0.155-	-	-	-	-	-
	D-CRTM-N	0.012	0.015	-	0.133 -	-	-	-	-	-
	Bi-RTM	0.003	0.705	-	0.483-	-	-	-	-	-
	WSB-RTM	0.704	1.023	-	-	1.653	2.9341	0.537	3.036	11
	LLDA	2.984	0.002	-	0.2435	-	-	-	-	-
	RTM	0.031	0.094	-	-	-	-	-	-	-
	LDA	0.013	0.9508	-	-	-	-	-	-	-
WebKB	D-CRTM-U	9.868	0.100	79	0.197	-	-	-	-	-
	D-CRTM-N	8.535	0.003	-	0.113	-	-	-	-	-
	Bi-RTM	9.064	0.119	-	0.372	-	-	-	-	-
	WSB-RTM	1.114	1.330	-	-	0.896	2.837	9.723	0.557	5
	LLDA	5.033	0.763	-	0.015	-	-	-	-	-
	RTM	7.257	0.080	-	-	-	-	-	-	-
	LDA	3.843	0.085	-	-	-	-	-	-	-

Table 6.6: Hyperparameter configurations identified by the single-objective optimization experiments for document classification.

want to maximize the performance for the classification task, we then intervene on the document-topic distribution θ. Since θ is driven to be spiked in correspondence of a given topic, any topic will never be a "background" topic (i.e. equally likely over all the documents). Therefore, when optimizing with respect to prediction capabilities, the models are able to also improve the KL-B metric ensuring a good topic quality.

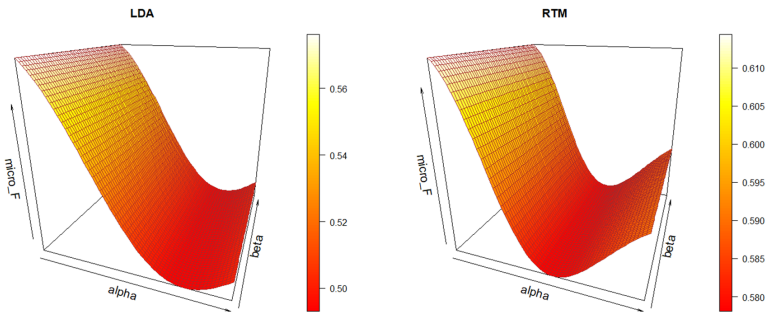


Figure 6.8: Approximated Micro-F1 for LDA (left) and RTM (right) for the Cora dataset.

The other metrics KL-U, KL-V, PUW and NPMI seem to have an opposite behavior with respect to KL-B: in most of the cases, the

	Models	Configuration	NPMI	PUW	KL-U	KL-V	KL-B
Cora	D-CRTM-U	F	0.105	0.563	1.342	0.551	**0.898**
		I	**0.112**	**0.743**	**1.695**	**0.897**	0.323
	D-CRTM-N	F	**0.122**	0.564	1.342	0.593	**1.372**
		I	0.107	**0.662**	**1.639**	**0.800**	0.213
	BIRTM	F	**0.115**	0.669	1.466	0.730	0.889
		I	0.113	**0.676**	**1.511**	**0.736**	**0.901**
	WSB-RTM	F	0.115	0.614	1.199	0.485	**1.512**
		I	**0.121**	**0.699**	**1.208**	**0.578**	0.769
	LLDA	F	0.076	0.404	0.883	0.154	**1.243**
		I	**0.077**	**0.422**	**0.887**	**0.166**	1.160
	RTM	F	0.114	0.661	1.340	0.599	**0.978**
		I	**0.115**	**0.694**	**1.632**	**0.840**	0.777
	LDA	F	0.112	0.613	1.106	0.425	**1.414**
		I	**0.113**	**0.753**	**1.609**	**0.837**	0.352
M10	D-CRTM-U	F	0.186	**0.850**	**1.635**	**0.985**	0.160
		I	**0.190**	0.838	1.601	0.951	**0.176**
	D-CRTM-N	F	0.187	**0.903**	**1.977**	**1.327**	0.139
		I	**0.191**	0.855	1.651	1.016	**0.155**
	BIRTM	F	0.185	0.872	1.715	1.070	**0.157**
		I	**0.187**	**0.874**	**1.758**	**1.109**	0.144
	WSB-RTM	F	0.191	**0.856**	**1.833**	**1.168**	0.253
		I	0.191	0.784	0.900	0.437	**0.691**
	LLDA	F	0.185	0.799	1.077	0.539	**0.298**
		I	**0.187**	**0.835**	**1.285**	**0.707**	0.259
	RTM	F	**0.003**	0.708	1.891	1.808	**1.637**
		I	-0.036	**0.978**	**2.798**	**1.892**	0.033
	LDA	F	-0.016	0.729	0.704	0.406	**1.966**
		I	**0.027**	**0.808**	**1.305**	**0.774**	0.521
WebKB	D-CRTM-U	F	0.186	**0.850**	**1.635**	**0.985**	0.160
		I	**0.190**	0.838	1.601	0.951	**0.176**
	D-CRTM-N	F	0.187	**0.903**	**1.977**	**1.327**	0.139
		I	**0.191**	0.855	1.651	1.016	**0.155**
	BIRTM	F	0.185	0.872	1.715	1.070	**0.157**
		I	**0.187**	**0.874**	**1.758**	**1.109**	0.144
	WSB-RTM	F	0.191	**0.856**	**1.833**	**1.168**	0.253
		I	0.191	0.784	0.900	0.437	**0.691**
	LLDA	F	0.185	0.799	1.077	0.539	**0.298**
		I	**0.187**	**0.835**	**1.285**	**0.707**	0.259
	RTM	F	0.185	0.871	**1.763**	1.111	**0.180**
		I	**0.189**	**0.883**	1.761	1.111	0.144
	LDA	F	0.188	**0.849**	**1.696**	**1.046**	**0.273**
		I	**0.194**	0.834	1.130	0.597	0.213

Table 6.7: Comparison of topic quality on the three datasets. Letter *F* indicates the best configuration identified by BO, while letter *I* denotes the best initial configuration identified during random sampling.

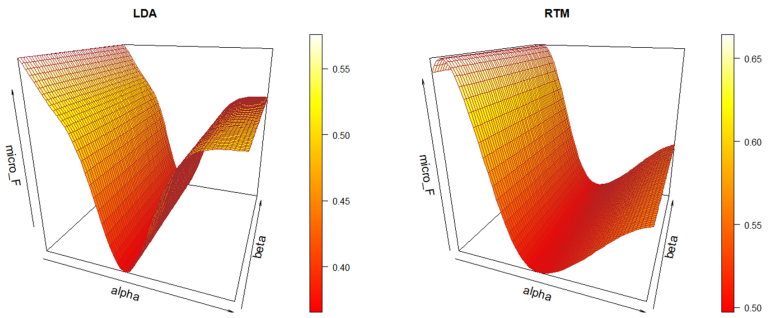


Figure 6.9: Approximated Micro-F1 for LDA (left) and RTM (right) for the M10 dataset.

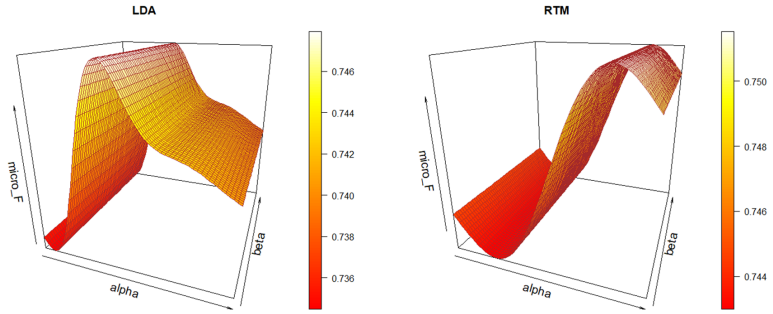


Figure 6.10: Approximated Micro-F1 for LDA (left) and RTM (right) for the WebKB dataset.

initial configuration leads to better performance in terms of these considered metrics. Since the models are optimized with respect to the micro-F1 measure affecting the document-topic distribution as the K-dimensional representation of a document, KL-U, KL-V, PUW and NPMI, which are metrics based on word-topic distribution, show a decreasing topic quality trend. We can therefore conclude that a model, whose hyperparameters are optimized for document classification purposes, is not able to maintain good capabilities in terms of topics interpretability and quality.

We report in Tables 6.8 and 6.9, the output of D-CRTM-N, being the model that obtains the best performance on Cora, and LLDA that obtains a comparable performance in terms of document classification on the same dataset, but cannot guarantee a good trade-off with respect to the other quality metrics (as we can observe in Table 6.7 and we will later observe in section 6.3.2). In particular, we report the topic words of D-CRTM-N and LLDA with the hyperparameter configurations that correspond to the best obtained Micro-F1.

By analyzing Table 6.8, related to the D-CRTM-N model, we can highlight that there is a good association between the class label and

CLASS	TOPIC WORDS	NPMI
Case-Based Reasoning	**learn** system case design **use** reason knowledg casebas **problem** plan	0.137
Genetic Algorithms	genet **algorithm problem use** program search **learn** evolv result optim	0.111
Reinforcement Learning	**learn** reinforc **use problem algorithm** method state control paper system	0.090
Probabilistic Methods	**model algorithm** bayesian **use** network estim method distribut belief **problem**	0.089
Theory	**learn algorithm use model** tree **problem** gener set show concept	0.073
Neural Networks	network **learn use** neural **model algorithm** system function method gener	0.064
Rule Learning	**learn** program **use** induct paper system exampl perform gener schedul	0.058
Average NPMI		0.0887
Average TOPIC DIVERSITY		0.5143

Table 6.8: Output generated by the D-CRTM-N model. The first column denotes the class associated to each topic, the second column shows the topic words and the third column reports the corresponding NPMI coherence. The words reported in **bold** denote the words shared across different topics.

the topic words that allows us to easily understand the content of each topic. Additionally, considering the NPMI coefficients of each topic and the words reported in bold that denote their presence in multiple topics, we can note a general good quality of the topics. Since only 35% of the words are shared between the extracted topics, and with an average NPMI equal to 0.0887, we can conclude that D-CRTM-N extracts coherent and diverse topic descriptors.

LLDA (see Table 6.9) on the other hand, although it achieves an average NPMI equal to 0.0923 (which is greater than D-CRTM-N), suffers of few limitations from the interpretation point of view. In fact, the LLDA output is characterized by two topics that correspond to the same class label and by 75% of the words that are shared between the extracted topics. This suggests us that, even if LLDA has higher performance from the NPMI point of view, it is not able to achieve a good trade-off with respect to other measures such as diversity. In conclusion, D-CRTM-N seems a promising model that provides the most remarkable trade-off between the topic coherence/diversity metrics when optimizing with respect to the F1 measure.

The results achieved so far open new research opportunities. In particular, since relational topic models have shown in most of the cases that prediction capabilities and topic interpretability have an opposite trend, it could be worth exploring which estimation strategy between

CLASS	TOPIC WORDS	NPMI
Case-based Reasoning	**learn system case use reason knowledg** casebas design **problem** plan	0.137
Case-based Reasoning	**learn system design use case reason knowledg casebas problem method**	0.116
Genetic Algorithms	genet **problem algorithm use program** search **learn** result **paper** optim	0.105
Reinforcement Learning	**learn** reinforc **algorithm problem use method** control state **paper function**	0.086
Rule Learning	**learn use program** exampl induct logic **gener paper** schedul **method**	0.070
Probabilistic Methods (Theory)	**algorithm learn model use problem method network** bayesian set **gener**	0.068
Neural network	**network learn use model** neural **algorithm system function method gener**	0.064
Average NPMI		0.092
Average TOPIC DIVERSITY		0.411

Table 6.9: Output generated by the LLDA model. The first column denotes the class associated to each topic, the second column shows the topic words and the third column reports the corresponding NPMI coherence. The words reported in **bold** denote the words shared across different topics.

constrained and multi-objective optimization (Horn and Bischl, 2016; Paria et al., 2019) leads to an optimal trade-off between them.

MICRO-F1 VERSUS QUALITY MEASURES We further analyze the trade-off between Micro-F1 and the other quality measures. It is important to remark that hyperparameter optimization has been performed with respect to a single objective, that is Micro-F1, but it is anyway important to investigate which is the algorithm offering the best trade-off between document classification and interpretability.

Considering all the bi-objective pairs Micro-F1 *vs* one of the quality measures, this results into five different charts for each datasets, comparing the *approximated Pareto frontiers* provided by the seven considered models. It is important to recall that a Pareto frontier $\mathcal{P}$, into a space spanned by two maximization objectives $f_1(x)$ and $f_2(x)$ – as in our case – is given by

$$\mathcal{P} = \left\{ \left(f_1(x), f_2(x)\right) : \nexists (f_1(x'), f_2(x')) : f_1(x') > f_1(x) \vee f_2(x') > f_2(x) \right\}.$$

It is also said that pairs on the Pareto frontier *dominate* all the others. The set of points x associated to pairs laying on the Pareto frontier is called Pareto set. In our case, a given point x is a hyperparameter configuration. Since we have sequentially sampled hyperparameter configurations, the resulting Pareto set and Pareto frontier can be only regarded as approximations of the actual ones.

Figure 6.11 is related to the Cora dataset and shows that there does not exist a clear winner among the considered models. However, some interesting results emerge. We can observe that LDA is always dominated by at least another algorithm. This result is reasonable, since the model does not incorporate the labels or the links. Among the CRTM models, D-CRTM-N always dominates D-CRTM-U, except for the case Micro-F1 *vs* KL-B. As far as KL-B and NPMI are considered as second objectives, the approximated Pareto frontier of D-CRTM-N consists of only one point: this means that optimizing Micro-F1 resulted equivalent to optimize KL-B or NPMI – that is this two objectives resulted correlated with Micro-F1. We can also observe that WSB-RTM and D-CRTM-N seem complimentary with respect to the cases Micro-F1 *vs* NPMI, Micro-F1 *vs* PUW, and (by also including D-CRTM-U) Micro-F1 *vs* KL-B. Thus, the union of their approximated Pareto frontiers leads to the best unique approximated Pareto frontier, dominating all the other algorithms.

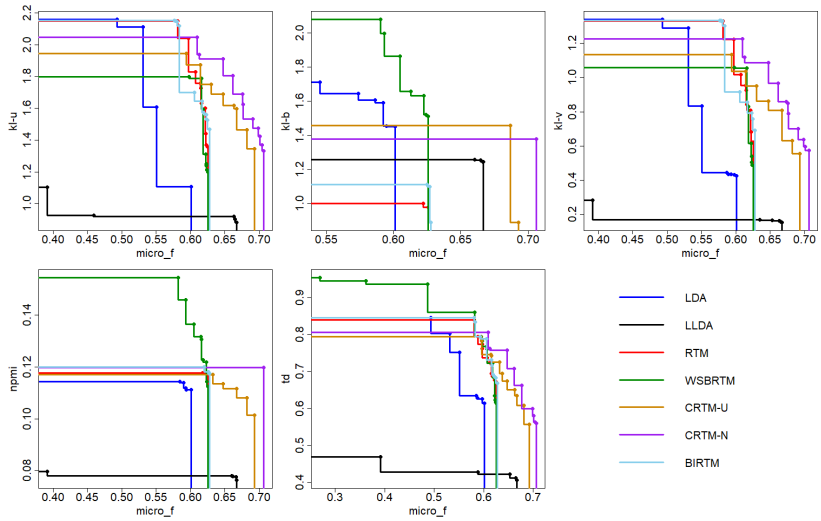


Figure 6.11: Pareto frontiers between accuracy (Micro-F1) and each one of coherence measures on the Cora dataset.

Figure 6.12 is related to the M10 dataset and, also in this case, a clear winner cannot be identified among the considered algorithms. We can notice that D-CRTM-N dominates all the other algorithms on two cases, specifically those having as second objective KL-U and KL-V. Substantially, the same consideration could be made also in the case Micro-F1 *vs* PUW, expect for a non-dominated pair for RTM and one for Bi-RTM. As already observed on the Cora dataset, with respect to KL-B as second objective, the approximated Pareto frontier of D-CRTM-N consists of only one point: this means that optimizing Micro-F1 resulted equivalent to optimize KL-B (i.e., KL-B is correlated with Micro-F1).

Finally, Figure 6.13 is related to the WEBKB dataset. As for the previous two datasets, there is not a clear winner among the considered

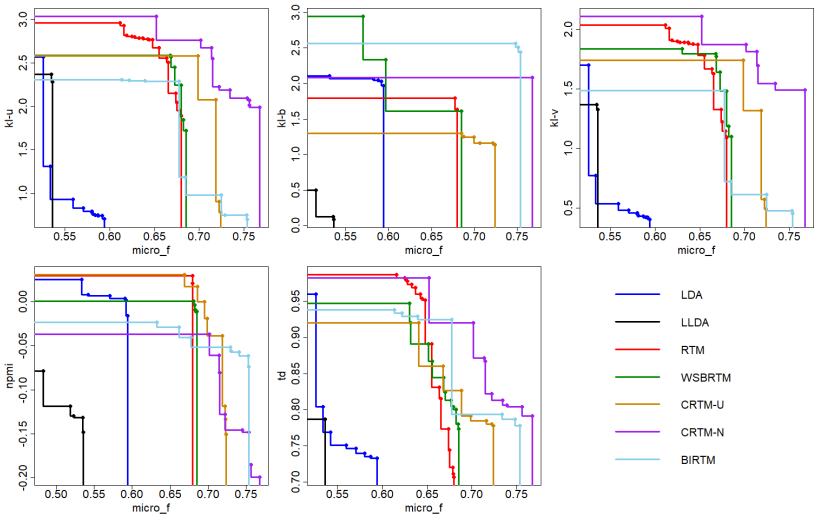


Figure 6.12: Pareto frontiers between accuracy (Micro-F1) and each one of coherence measures on the M10 dataset.

models, even if D-CRTM-N resulted (again) the best option. More specifically, D-CRTM-N dominates all the other algorithms, with the following exceptions:

- WSB-RTM, in all the cases. The union of the two approximated Pareto frontiers – the D-CRTM-N's and the WSB-RTM's ones – results in the best approximated Pareto frontier dominating all the other algorithms (except for one LDA's point in the Micro-F1 *vs* NPMI case);
- LDA, for just one point, in the case having NPMI as second objective;
- with respect to PUW as second objective, the approximated Pareto frontier of D-CRTM-N consists of only one point: this means that optimizing Micro-F1 resulted equivalent to optimize PUW on this specific dataset.

As a final consideration, optimizing the Micro-F1, only, allowed us to identify accurate models, for each one of seven algorithms considered but, according to the a-posteriori Pareto efficiency analysis, we can conclude that D-CRTM-N is the best-performing algorithm, because it provides the most reasonable trade-off with the topic coherence metrics. On the other hand, WSB-RTM provides less accurate results in terms of Micro-F1, but the topics appear to be the most coherent.

This behavior may be due to the fact that WSB-RTM does not take into account the knowledge related to the labels of the documents, which provides a great improvement in the performance in terms of Micro-F1. On the other hand, explicitly modeling blocks of documents, instead of just the relationships between pairs of documents,

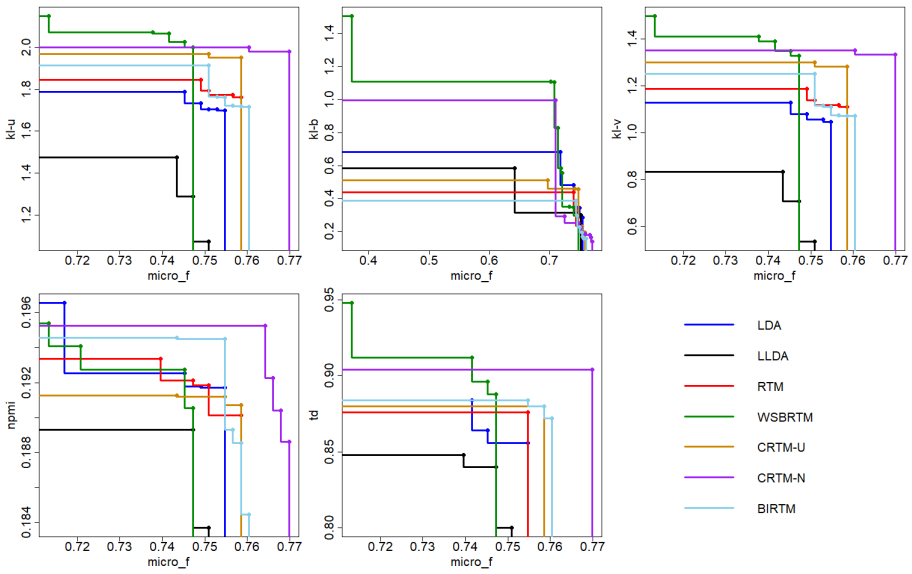


Figure 6.13: Pareto frontiers between accuracy (Micro-F1) and each one of coherence measures on the WebKB dataset.

can help the topic models to account for those documents that are similar but they are not considered as similar because there is not a link between them (we way not have this information). Recognizing that a group of documents belong to the same block, even if not all the documents link to each other, encourages that the documents to share the same topic distributions, thus improving the resulting topic coherence.

6.4 COMPARATIVE ANALYSIS OF NEURAL TOPIC MODELS

We will now demonstrate that we can use Bayesian Optimization as well for the other main class of topic models, i.e. Neural Topic Models. Neural Topic Models. For this category of models, the problem of finding the best hyperparameter configuration has become even more compelling, since topic models based on neural networks are usually controlled by a high number of hyperparameters. We therefore perform an empirical analysis of recent NTMs by optimizing the hyperparameters of the models with respect to different metrics. We aim to investigate if the relationship between hyperparameters, document length and performance measures, to finally understand under which conditions we can exploit at best the potentiality of each model.

To this purpose, we use Bayesian Optimization (BO) (Archetti and Candelieri, 2019), a well-known and efficient strategy for hyperparameter tuning, to determine the optimal hyperparameter settings for four different evaluation metrics of five state-of-the-art NTMs. The hyperparameter optimization allows us to guarantee a fair compari-

son between the models and investigate their behavior with different hyperparameter settings.

6.4.1 *Methodology*

As seen before, we adopt a single-objective Bayesian Optimization approach, using the comparative framework topic modeling OCTIS (Terragni et al., 2021a), to optimize the hyperparameters of five different topic models with respect to four different evaluation metrics. Each metric investigates a different aspect of a model.

Optimizing a model's hyperparameters not only allows us to investigate the robustness of a model over different evaluation metrics, but we can also investigate the performance of the optimized evaluation metric on datasets with different features and the relationship between the optimized evaluation metric and the other metrics.

TOPIC MODELS. In our investigation, we focus on the following state-of-the-art topic models based on a neural variational frameworks. We consider Neural LDA (Srivastava and Sutton, 2017, NeurLDA), Product-of-experts LDA (Srivastava and Sutton, 2017, ProdLDA), the Embedded Topic Models (Dieng et al., 2020, ETM), and finally we use a variant of the family of Contextualized Topic Models (CTM), namely the ZeroShotTM we have presented in Section 5.3.

Concerning ETM, following the original paper, we will refer to the former version of the model as ETM, while the one that uses pretrained word embeddings (PWE) will be referred to as ETM-PWE.

We also consider the well-known Latent Dirichlet Allocation (Blei et al., 2003a, LDA) as a baseline.

EVALUATION METRICS. We consider four evaluation metrics that investigate different aspects of a topic model:

- Micro-F1 (**F1**): We train a linear support vector machine (SVM) that predicts the document's class using the topic distribution θ of each document (given by each topic model) as its feature representation.
- **IRBO** on the 10-top words of the topics (as defined in Section 5.2).
- **NPMI** computed on the 10-top words of the topics. The word co-occurrences are computed on the original dataset.
- KL-Background **(KL-B)**.

6.4.2 *Experimental Setting*

DATASETS AND PREPROCESSING To analyze the impact of the length of the documents with respect to several models and performance measures, we consider two different datasets: 20Newsgroup[10] (20NG), where each document is characterized by a long text, and M10 (Lim and Buntine, 2015), which is composed of titles of scientific papers, and therefore it represents a case study of short texts. We use the datasets as provided in OCTIS.

BAYESIAN OPTIMIZATION AND MODEL SETTINGS. We optimize the models' hyperparameters using BO for each evaluation metric. We trained each model 30 times and considered the median as the evaluation of the function to be optimized. The initial configurations are randomly sampled via Latin Hypercube Sampling and equal to the number of the hyperparameters to optimize plus 2 (to provide enough configurations for the initial surrogate model). The total number of BO iterations is 30 for LDA and 120 for the other models. We use Random Forests as the surrogate model and the Upper Confidence Bound (UCB) as the acquisition function.

We report the models' hyperparameters and their corresponding ranges in Table 6.10.

Model	**Hyperparameter**	**Range**
LDA	α prior	$[10^{-4}, 10]$
	β prior	$[10^{-4}, 10]$
NeurLDA/ ProdLDA/ CTM	Dropout	$[0, 1 - 10^{-6}]$
	Learning rate	$[10^{-6}, 10^{-1}]$
	Momentum	$[0, 1]$
	Activation function	elu, leakyrelu, relu, rrelu, selu, sigmoid, softplus
	Optimizer	adadelta, adagrad, adam, rmsprop, sgd
	# Neurons	100, 200, ..., 1000
	# Layers	1, 2, 3, 4, 5
	Learn priors	true, false
ETM/ ETM-PWE	Dropout	$[0, 1 - 10^{-6}]$
	Learning rate	$[10^{-6}, 10^{-1}]$
	Weight decay	$[10^{-6}, 10^{-1}]$
	Activation function	elu, leakyrelu, relu, rrelu, selu, softplus, tanh
	Optimizer	adadelta, adagrad, adam, asgd, rmsprop
	# Neurons	100, 200, ..., 1000
ETM	Rho size	100, 200, 300

Table 6.10: Hyperparameters and ranges.

10 http://qwone.com/ jason/20Newsgroups/

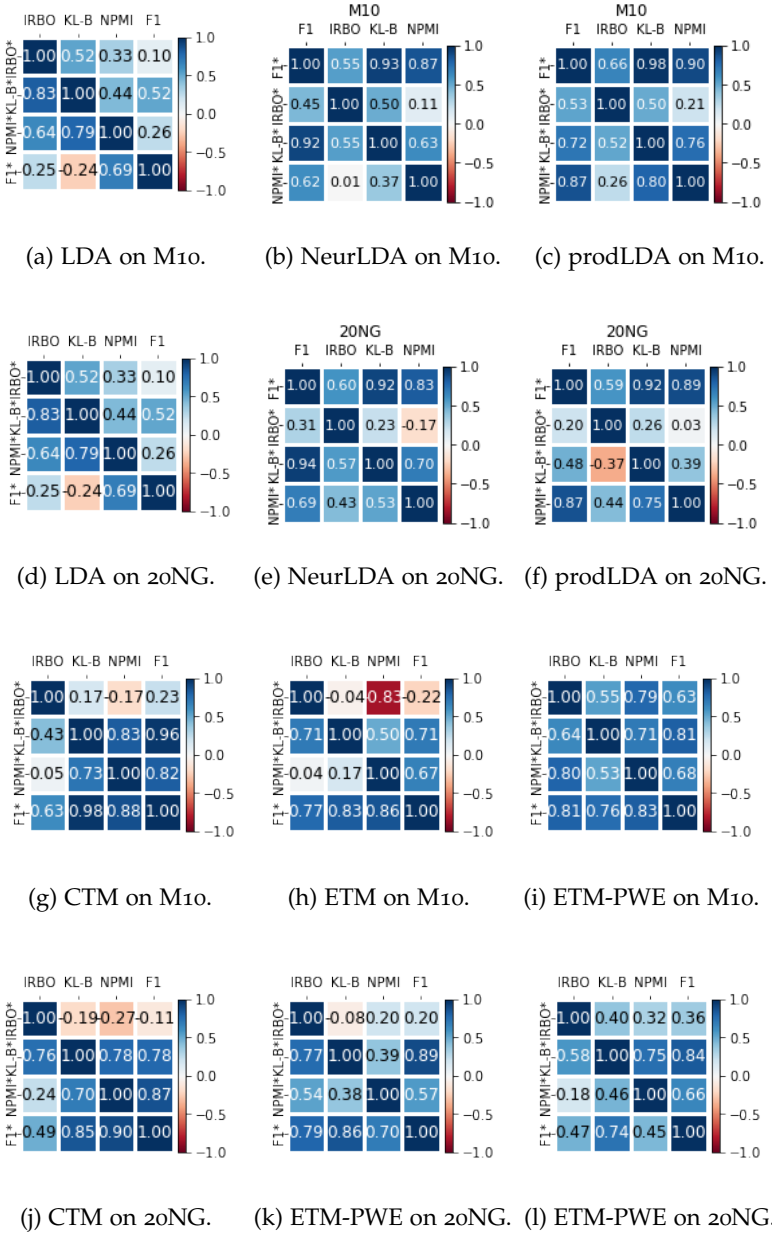
	20NG				M10			
	F1*	**IRBO***	**NPMI***	**KL-B***	**F1***	**IRBO***	**NPMI***	**KL-B***
LDA	0.469	0.963	0.064	**2.299**	0.472	0.944	-0.089	**2.343**
NeurLDA	0.339	**1.000**	0.067	0.907	0.420	**1.000**	-0.131	0.904
ProdLDA	0.373	0.998	0.107	0.992	0.539	**1.000**	0.044	1.652
CTM	0.361	0.998	**0.118**	1.019	0.563	**1.000**	**0.055**	0.937
ETM	0.453	0.996	0.080	0.370	0.534	0.997	-0.028	0.532
ETM-PWE	**0.471**	0.986	0.089	0.424	**0.585**	0.997	-0.070	0.201

Table 6.11: Median of each performance metric (columns) for each single-objective optimization (rows).

Regarding LDA, we optimize the hyperparameters α and β priors that the sparsity of the topics in the documents and sparsity of the words in the topic distributions respectively. These hyperparameters are set to range between 10^{-4} and 10 on a logarithmic scale.

The hyperparameters of the neural models are mainly related to the architecture of the network. For all the neural models, we optimize the *dropout* (ranging between 0 and $1 - 10^{-6}$) and the *momentum* (ranging between 0 and 1). We optimize the *learning rate*, that is set to range between 10^{-4} and 10^{-1}, on a logarithm scale. We also consider different variants of *activation functions* and *optimizers*.

Regarding NeurLDA, ProdLDA, and CTM in particular, we optimize the *number of layers* (ranging from 1 to 5), and the *number of neurons* (ranging from 100 to 1000). For simplicity, each layer has the same number of neurons. Finally, we also consider the hyperparameter *learn priors* that controls if the priors are learnable parameters.



(a) LDA on M10. (b) NeurLDA on M10. (c) prodLDA on M10.

(d) LDA on 20NG. (e) NeurLDA on 20NG. (f) prodLDA on 20NG.

(g) CTM on M10. (h) ETM on M10. (i) ETM-PWE on M10.

(j) CTM on 20NG. (k) ETM-PWE on 20NG. (l) ETM-PWE on 20NG.

Figure 6.14: Metrics-metrics correlations.

We use the contextualized document representations derived from SentenceBERT (Reimers and Gurevych, 2019). We use the pre-trained BERT model fine-tuned on the natural language inference (NLI) task.[11]

Considering ETM and ETM-PWE, in addition to the hyperparameters mentioned above, we only optimize the *number of neurons* (ranging from 100 to 1000). We follow the original implementation, for which the number of hidden layers is set to 1. For ETM-PWE, we use pre-trained Word2vec word embeddings (Mikolov et al., 2013),

11 https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

trained on the Google News corpus (3 million 300-dimension English word vectors).

For the neural models, we set the batch size to 200 and we adopted an early stopping criterion for determining the convergence of each model. We set the remaining model parameters to their default values. We set the number of topics to be discovered equal to the number of classes available in each dataset, i.e. 10 for M10 and 20 for 20NG. For running the experiments, we use OCTIS, which already integrates the implementations of the considered models and metrics.

6.4.3 *Empirical Analysis and Discussion*

ROBUSTNESS OF NEURAL TOPIC MODELS. In table 6.11 we report the median of the four evaluation metrics for each topic model obtained by the best hyperparameter configuration. Rows represent the optimized metric (marked as *metric**), while columns denote the median of the evaluated metric. The overall best values for each metric and dataset are reported in bold. First of all, we can observe that there is not a model that outperforms the others for all the considered metrics. In fact, it seems that each topic model works better for a specific metric.

In particular, LDA is the topic model that obtains the best performance in terms of KL-B*, thus obtaining topics that are significant rather than background topics. While, the topic models based on the neural variational framework defined in (Srivastava and Sutton, 2017), i.e. NeurLDA, ProdLDA, and CTM, are the ones that obtain the highest diversity. Regarding the topic coherence, CTM obtains the best topic coherence for both datasets. In fact, it improves the performance of ProdLDA (second-best model for the topic coherence) through the incorporation of the contextualized pre-trained representations of the documents. Finally, ETM-PWE outperforms the other models in terms of F1*, probably due to the contribution of the pre-trained word embeddings.

Provided that each topic model seems to reach the best performance only in a specific metric, it follows that they cannot simultaneously guarantee optimal performance for the other metrics. We will further investigate the trade-off between different metrics in the following analysis. A complete overview of the best configuration of hyperparameters discovered by BO for all the models and for all the considered evaluation measures is reported in Tables C.1, C.2, C.3, C.4, C.5 and C.6 in the Appendix C. This would allow a user to choose a promising hyperparameter configuration for the evaluation metric of their interest.

IMPACT OF THE DOCUMENT LENGTH. We can derive other insights by analyzing Table 6.11 and comparing the two datasets. In

particular, we highlight that for LDA the document length seems to be an invariant when optimizing on the KL-B* metric. This insight can be grasped by considering the KL-B* of LDA (i.e. 2.343 for M10 and 2.299 for 20NG) that, not only are the best performance when compared to the other models, but they suggest that LDA performs well independently on the document length and therefore it guarantees optimal KL-B* both on short and long documents.

Another important insight is about the F1 measures obtained by LDA (0.472 and 0.469), ETM (0.534 and 0.453), and ETM-PWE (0.585 and 0.471), which seem to be not affected by the length of the documents. On the other hand, the results for the F1 measure for NeurLDA, ProdLDA, and CTM (which are based on the same architecture) are affected by the documents' length, obtaining the best performance for short texts. In these cases, when the models achieve a high F1 on short documents (0.420 by NeurLDA, 0.539 by ProdLDA, and 0.563 by CTM), the performance on short documents is lower (0.339 by NeurLDA, 0.373 by ProdLDA, and 0.361 by CTM).

When optimizing for the IRBO* metric, all the models succeed in obtaining almost completely diverse topics, both for long and short texts. The performance of IRBO* for LDA* is slightly affected when dealing with short texts. Finally, we remark that CTM obtains an excellent topic coherence for both datasets, but, on the other hand, the remaining models seem to be particularly affected when dealing with short texts, assuming NPMI values inferior to 0.

METRICS-METRICS CORRELATIONS. In Figure 6.14, we report the correlations between the evaluation metrics when a single-objective optimization policy is performed. The rows of the correlation matrices denote the optimized metrics (F1*, IRBO*, KL-B*, and NPMI*), while the columns the non-optimized evaluated measures (F1, IRBO, KL-B, and NPMI). According to these results, we can observe if optimizing a model for a specific metric allows us for an increasing or decreasing performance of the other metrics. In Figure 6.14, we report the Spearman correlation coefficients between metrics using all the runs of a given experiment.

Concerning LDA, when the model is optimized for the KL-B*, NPMI*, or F1*, then the IRBO is positively correlated with these metrics. It is then sufficient to optimize one of the other metrics to get also diverse topics. This occurs in particular for the KL-B* and NPMI* on long documents (0.87 and 0.98 respectively). It is also interesting to notice that optimizing for KL-B* does not imply a maximization for the F1 and NPMI on long texts. To achieve better topic coherence and classification, we should consider background topics as well.

Focusing on NeurLDA, ProdLDA, and CTM, we do not observe substantial differences between long and short documents. IRBO* is not strongly correlated with the other metrics, especially for long doc-

uments. This can be grasped by observing the coefficients IRBO* vs F1, KL-B, and NPMI reported in Figure (6.14e), (6.14f) and (6.14j). On the contrary, optimizing NeurLDA, ProdLDA, and CTM for F1*, NPMI* or KL-B* guarantees, in most of the cases, a good performance on all the metrics both for short and long documents (Figure (6.14b), (6.14e), (6.14c) and (6.14f)).

Concerning ETM, the difference between long and short documents is clear: the optimization of a given metric can be detrimental to the majority of the other metrics when dealing with short documents. In fact, the optimization of ETM w.r.t. IRBO* and NPMI* originates correlation values with all the other metrics that are close to zero or negative (Figure 6.14h). On the other hand, F1* and KL-B* seem not to be affected by the difference of the datasets. This suggests that maximizing KL-B* or F1* implies good performance also for other purposes. Focusing on long documents (Figure 6.14k), the optimization of ETM w.r.t. F1*, KL-B*, and NPMI* originates positive correlation values for all the other metrics. On the other hand, we can highlight that optimizing the topic diversity IRBO* does not allow us to simultaneously obtain good performance on topic coherence (NPMI) on long documents. Regarding ETM-PWE, we do not notice a clear difference between the two datasets. The introduction of the pre-trained word embedding into the training process of the model seems to be beneficial for all the metrics.

To summarize, optimizing the neural models according to the IRBO* is not always convenient and may lead to incoherent topics or poor document classification performance. Another important insight concerns the optimization of F1*, which usually guarantees to maximize IRBO, KL-B, and NPMI, for both short and long documents, except for LDA.

6.5 SUMMARY OF THIS CHAPTER

In the following, we give a short summary of this chapter. In the introduction of the chapter, we reported the following research questions:

Q6.1 Can we determine if a topic model can guarantee an optimal trade-off between different performance measures?

Q6.2 Can a performance measure imply a competing or correlated target for other performance measures?

To answer these questions, we provide a solution based on Bayesian Optimization. We propose to optimize the hyperparameters of topic models using single-objective BO. Moreover, to make it accessible to everyone in the NLP and ML community, we also release a python library that integrates topic models, evaluation metrics, pre-processed datasets, and hyperparameter optimization in the same place.

We showed the effectiveness of BO on different sets of topic models in Sections 6.3 and Sections 6.4. Our results show that for both categories of topic models, a single-objective optimization strategy leads to optimize the target function (Q6.1), but this can be detrimental to other evaluation metrics (Q6.2). We also analyzed the impact of the Bayesian optimization by varying models, datasets and evaluation metrics.

The comparative analysis and experiments that we have carried out on different topic models have indeed several implications. In the following Chapter we will explore two fundamental directions: (1) the use of multi-objective hyperparameter optimization for discovering the best trade-off between different metrics and (2) hyperparameter transfer from a dataset to an unseen dataset for a more efficient discovery of the optimal hyperparameter configurations.

7

BEYOND SINGLE-OBJECTIVE HYPERPARAMETER OPTIMIZATION

In Chapter 6 we have seen how to use single-objective hyperparameter optimization for guaranteeing a fairer comparison between the models. Although this approach may be useful to identify the best hyperparameter configuration for an evaluation metric, it disregards possible competing objectives. Indeed, **we may be interested into optimizing more than objective at the same time.** In this context, a multi-objective approach is ideal. This would allow researchers to discover the topic model that guarantees the best trade-off between the different metrics of interest.

Moreover, **although Bayesian Optimization is more efficient than other methods (Snoek et al., 2012), the algorithm requires a fair amount of iterations** to guarantee the convergence to an optimal solution, especially when the number of hyperparameter is large. Similarly to an expert who selects the best hyperparameters given their prior knowledge, we should look forward to adopting an automatic method to exploit the knowledge we have acquired during our past experiments to select a good hyperparameter configuration for a topic model. We refer to this transfer learning mechanism as to *hyperparameter transfer*. Testing if the hyperparameters can be transferred from a dataset to an unseen dataset is by no means a necessary step that would allow us to reduce the computational costs of the Bayesian Optimization.

In this Chapter, we will therefore overcome the limitations of the single-objective optimization approach and address the following research questions:

Q7.1: Can we optimize the hyperparameters of a topic model to guarantee an optimal trade-off between different performance measures using multi-objective optimization?

Q7.2: Can we transfer the best hyperparameter configurations from a dataset to an unseen dataset?

In this Chapter, we will apply a multi-objective optimization approach to different categories of topic models to reveal the trade-off between different evaluation metrics in Section 7.1. We will also investigate an approach to transfer to best hyperparameter configurations of a given metric to an unseen dataset in Section 7.2. This will also allow us to investigate which dataset features may play a role in the transfer learning process.

7.1 MULTI-OBJECTIVE OPTIMIZATION FOR TOPIC MODELS (OCTIS 2.0)

Single-objective BO can be generalized to multiple objective functions (Paria et al., 2019). Instead of having a single objective function to optimize, we can formulate the problem in the following way:

$$\mathbf{x}^* = \arg\max_{x \in \mathcal{X}} (f_1(x), f_2(x), \ldots, f(x_N)) \tag{56}$$

where $N > 1$ is the number of objective functions f_i to optimize and $\mathcal{X}$ is a design space of interest. As before, the space can be composed of hyperparameters of different types: categorical, continuous or also conditional inputs.

It is not always possible to optimize all the metrics jointly, but instead some functions may be competing. For this reason, the final aim of Multi-objective Optimization is to recover the Pareto frontier of the objective functions, i.e. the set of Pareto optimal points. A point is Pareto optimal if it cannot be improved in any of the objectives without degrading some other objective. We also say that a point $x_1 \in \mathcal{X}$ dominates $x_2 \in \mathcal{X}$ then x_1 beats or ties x_2 along every possible objective taken into consideration. More formally, a point x_1 dominates x_2 iff $f_n(x_1) \geqslant f_n(x_2) \forall n \in N$ and $\exists n \in N$ s.t. $f_n(x_1) > f(x_2)$.

Here, we will use a recent multi-objective methodology presented in (Paria et al., 2019). This approach uses scalarization functions, which convert multi-objective values to scalars. We refer the readers to the original paper for additional details on multi-objective Bayesian optimization.

We integrate this approach, with appropriate changes into OCTIS, thus originating OCTIS v2.0.

EXAMPLE OF USAGE OF OCTIS 2.0. We report a simple code snippet that will run a multi-objective optimization experiment. This is very similar to the single-objective approach. The users only needs to provide a dataset, a model, the hyperparameter space (defined in a configuration file) and the metrics to optimize.

```
# loading of a pre-processed dataset
dataset = Dataset()
dataset.fetch_dataset("M10")

#model instantiation
lda = LDA(num_topics=25)

#definition of the metrics to optimize
td = TopicDiversity()
coh = Coherence()
metrics = [td, coh]

#definition of the search space
config_file = "path/to/search/space/file"
```

```
#define and launch optimization
mmm = MOOptimizer(
        dataset=dataset, model=model,
        config_file=config_file,
        metrics=metrics, maximize=True)
mmm.optimize()
```

This code will return the values of the hyperparameters corresponding to the points on the Pareto front for LDA with 25 topics, discovered by maximizing the diversity and the coherence of the topics on the dataset M10.

7.1.1 *Experimental Setting*

OBJECTIVE FUNCTIONS. In our investigation, we consider three well-known objective functions that consider different aspects of a topic model: the quality of the topics (NPMI), the diversity of the topics (IRBO, Section 5.2), and the prediction capability of the model in a classification task (F1). These three aspects are usually investigated in the topic modeling literature (Chang et al., 2009; Dieng et al., 2020). However, this set could be extended or reduced based on the needs of the users' needs.

All the considered functions must be maximized. NPMI and IRBO are computed on the top-10 words of each topic. We use a polynomial SVM and we compute the Micro-F1 measure. We will refer to this metric as F1.

MODELS In our evaluation, we consider three distinct topic models, chosen to be the representatives of different categories of topic models (Stevens et al., 2012; Zhao et al., 2021): classical probabilistic models, matrix factorization methods, and neural topic models. Due to their different formulations, all the considered models are controlled by different types of hyperparameters that we will detail later.

- LDA (as defined in 2.4).
- NMF (Non-Negative Matrix Factorization) (Paatero and Tapper, 1994) [1] is a statistical method that reduces the dimensionality of the input corpus of D documents, viewed as a matrix M of shape D $\times$ $|W|$, where $|W|$ represents the length of the vocabulary. It aims at decomposing M as the product of two matrices V and H, such that the dimension of V is $|W| \times$ K and that of H is D $\times$ K. The decomposed matrices must consist of only non-negative values.
- ZeroShotTM (as defined in 5.3).

1 https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

DATASETS. We consider six datasets: 20 NewsGroups (20NG), AFP[2], BBC News (BBC), M10, StackOverflow (SO)[3], and SearchSnippets (SS)[4] (Phan et al., 2008). These datasets pertain to different domains. In the topic modeling literature, it is well-known that the length of the documents can affect the performance of the topic models (Albalawi et al., 2020). For this reason, we selected 3 datasets composed of long texts (20NG, AFP, BBC) and 3 datasets composed of long texts (M10, SS, SO). Moreover, the AFP dataset is in French, while the others are in English.

Datasets SS and SO are already preprocessed. The dataset source refers to (Phan et al., 2008) for the dataset details; however the pre-processing pipeline is not available. Datasets 20NG, M10, and BBC are the ones available in OCTIS. We refer the reader to Section 6.2 for details on the preprocessing. For APF, we followed the same preprocessing pipeline, except we removed the words that have document frequency lower than 1% and higher than 70% and we removed the documents with less than 5 words.

For the sake of completeness, we report the main statistics of the pre-processed datasets in Table 7.1 and the pre-processing details in the Appendix. The datasets are split in training (70%), testing (15%) and validation set (15%).

Name	# Docs	# Labels	# Unique words	Avg doc length (std)
20NG	16,309	20	1612	48 (130)
AFP	26,599	17	2686	156 (174)
BBC	2,225	5	2949	120 (72)
M10	8,355	10	1696	6 (2)
SS	12,295	8	4705	14 (5)
SO	16,407	20	2257	5 (2)

Table 7.1: Characteristics of the considered datasets.

MULTI-OBJECTIVE HYPERPARAMETER OPTIMIZATION SETTINGS We use the *Dragonfly* library (Kandasamy et al., 2020; Paria et al., 2019) to simultaneously optimize topic quality (NPMI), topic diversity (IRBO) and classification (F1). To obtain robust evaluations of the objective metrics, we train each model 30 times and consider the median of the 30 evaluations as the evaluation of the function to be optimized. A number n of initial configurations is randomly sampled via Latin Hypercube Sampling, with n equal to the number of hyperparameters to optimize plus 2 to provide enough configurations for

2 http://193.55.113.124/topic-model-api/dataset/afp_fr.tsv
3 https://github.com/qiang2100/STTM
4 https://github.com/qiang2100/STTM

the initial surrogate model to fit. The total number of BO iterations for each model is 125. We use Gaussian Process as the probabilistic surrogate model and the Upper Confidence Bound (UCB) as the acquisition function.

Model	Hyperparameter	Values/Range
All	Number of topics	[5, 150]
LDA	α prior	$[10^{-4}, 10]$
	β prior	$[10^{-4}, 10]$
NMF	Regularization factor	[0, 0.5]
	L1-L2 ratio	[0,1]
	Initialization method	random, nndsvd, nndsvda, nndsvdar
	Regularization	V matrix, H matrix, both
CTM	Activation function	softplus, relu, sigmoid, leakyrelu, rrelu, elu, selu
	Dropout	[0, 0.95]
	Learn priors	true (1), false (0)
	Learning rate	$[10^{-4}, 10^{-1}]$
	Momentum	[0, 0.9]
	Number of layers	{1, 2, 3, 4, 5}
	Number of neurons	{100, 200, . . ., 900, 1000}
	Optimizer	adagrad, adam, sgd, adadelta, rmsprop

Table 7.2: Hyperparameters and ranges.

HYPERPARAMETER SETTING We summarize the models' hyperparameters and their corresponding ranges in Table 7.2. For each model, we optimize the number of topics, ranging from 5 to 150 topics. Regarding LDA, we also optimize the hyperparameters α and β priors that the sparsity of the topics in the documents and sparsity of the words in the topic distributions respectively. These hyperparameters are set to range between 10^{-4} and 10 on a logarithmic scale.

The hyperparameters of NMF are mainly related to the regularization that can be applied to the factorized matrices. The *regularization* hyperparameter controls if the regularization is applied only to the matrix V, or to the matrix H, or both of them. The *regularization factor* denotes the constant that multiplies the regularization terms. It is set to range between 0 and 0.5 (where 0 means no regularization). *L1-L2* ratio controls the ratio between L1 and L2-regularization. It ranges between 0 and 1, where 0 corresponds to L2 regularization only, 1 corresponds to L1 regularization only, otherwise it is a combination of the two types. We also optimize the *initialization method* for the two matrices *W* and H.

Since CTM is a neural topic model, its hyperparameters are mainly related to the network architecture. We optimize the *number of layers* (ranging from 1 to 5), and the *number of neurons* (ranging from 100 to 1000, with a step of 100). For simplicity, each layer has the same number of neurons. We also consider different variants of *activation functions* and *optimizers*. We set the *dropout* to range between 0 and 0.95 and the *momentum* between 0 and 0.9. Finally, we optimize the *learning rate*, that is set to range between 10^{-4} and 10^{-1}, on a logarithm scale, and the hyperparameter *learn priors* that controls if the priors are learnable parameters. We fix the batch size to 200 and we adopted an early stopping criterion for determining the convergence of each model. We use the contextualized document representations derived from SentenceBERT (Reimers and Gurevych, 2019). In particular, we use the pre-trained RoBERTa model fine-tuned on STS[5] for the English datasets and the multilingual Universal Sentence Encoder[6] for AFP.

For all the models, we set the remaining parameters to their default values.

7.1.2 *Results*

In the following, we discuss the results of the MOBO experiments and the hyperparameter transfer experiments. We report the best 5 hyperparameter configurations for each model and metric on each dataset in Appendix C.

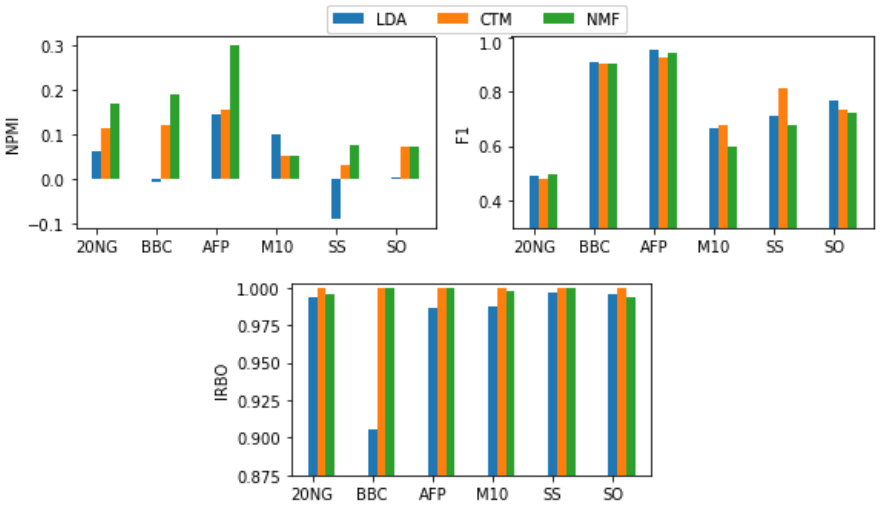


Figure 7.1: Best performance of the topic models for each evaluation metric on the considered datasets.

5 `stsb-roberta-large`
6 `distiluse-base-multilingual-cased-v1`

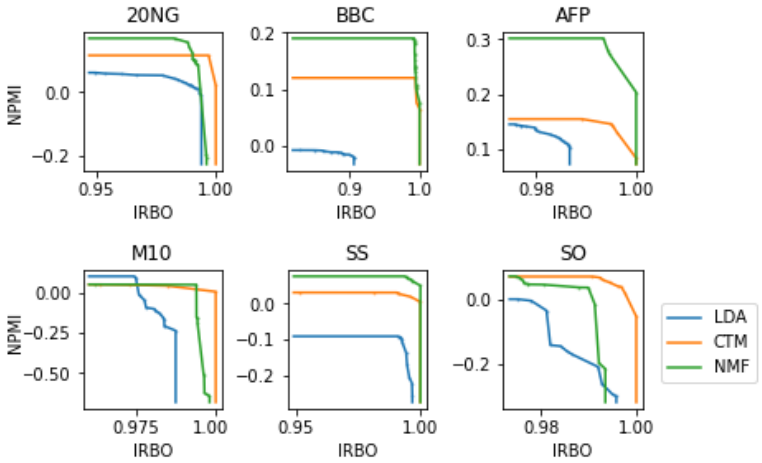


Figure 7.2: Pareto frontier for the metrics NPMI and IRBO for each model on the considered datasets.

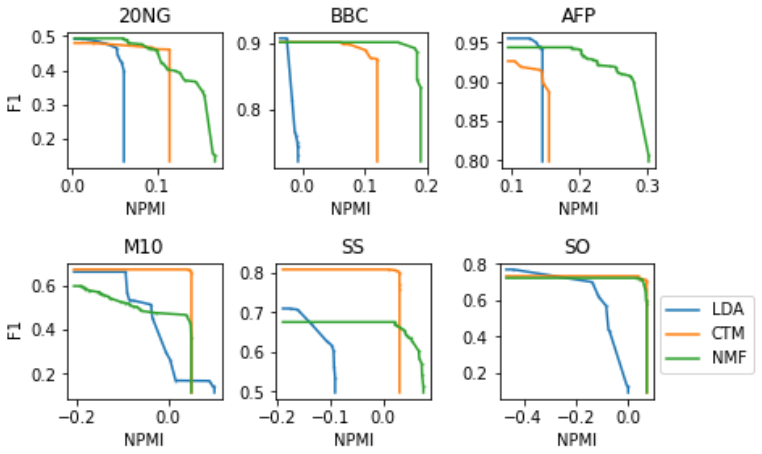


Figure 7.3: Pareto frontier for the metrics F1 and NPMI for each model on the considered datasets.

NO TOPIC MODEL WINS THEM ALL. Figure 7.1 reports the best performance of the models for each metric and dataset obtained by the MOBO experiments. It is important to notice that the hyperparameter configuration that allows a topic model to obtain the best performance for a given metric may differ from the optimal hyperparameter configuration for another evaluation metric.

Regarding the models' performance for the topic coherence (plot on the left), we can observe that NMF outperforms the other models in most cases. The stronger regularization in NMF generally leads to sparse topics and this likely leads to higher coherence scores (Burkhardt and Kramer, 2019). Considering the predicting capabilities of the models (central plot of Figure 7.1), CTM usually outperforms LDA and NMF for short-text documents (M10, SS), while LDA gets the best results in long-text datasets (20NG, BBC, AFP). We note that CTM incorporates contextualized representations originated by a limited

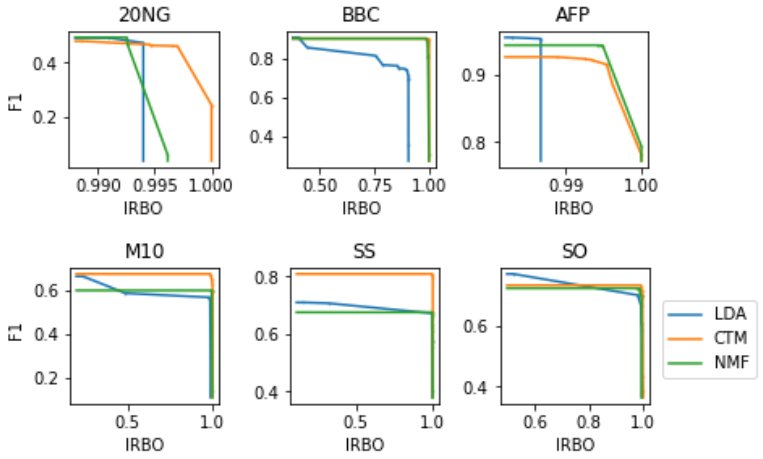


Figure 7.4: Pareto frontier for the metrics F1 and IRBO for each model on the considered datasets.

number of tokens, and not by the entire document. It follows that the representations of CTM may not produce accurate results for long-text documents. Finally, in the right plot of Figure 7.1, we observe that CTM and NMF reach comparable topic diversity performance, often getting topics that are totally different from each other.

As a concluding remark, except for the coherence, we showed that it is difficult to determine an always-winning topic model when we boost the performance of the models using multi-objective optimization. This finding is consistent with other investigations (Korencic et al., 2018; Stevens et al., 2012), despite that previous work did not optimize the models' hyperparameters. This result raises a question on the fairness of the past comparisons between topic models. This contributes to the growing amount of negative results when reviewing previously published work in light of new experiments (Rogers et al., 2020b).

CONFLICTING OBJECTIVES. Although considering the best performance for each topic model can provide an indicator of its capabilities, it is essential not to focus on a single metric, but rather to jointly consider multiple objectives. Hereby, we show the trade-off between a pair of metrics by plotting the Pareto frontier of the considered metrics. Figures 7.2, 7.3 and 7.4 show the frontier of each model for the pairs of metrics (NPMI, IRBO), (F1, NPMI) and (F1, IRBO) respectively.

We can observe that in most cases no model dominates the others, i.e. there is not any Pareto frontier that is better than the others for all the objectives. For example, if we consider the frontiers for NPMI and IRBO on 20NG, the frontier of the models CTM and NMF dominate LDA. However, CTM and NMF do not dominate each other. In other

words, for the dataset 20NG, a user that aims at obtaining coherent and diverse topics has to compromise between the two objectives.

In some cases, a topic model that outperforms the others for a given metric performs very poorly considering other metrics. For example, LDA on M10 obtains the best topic coherence but achieves a very low F1 (<0.2). Specifically, when considering F1 vs NPMI, we observe that to obtain high performance for a given metric we need to degrade the others, and vice versa.

These results enforce the idea of not limiting the experimental campaign of topic models to a single-objective hyperparameter optimization approach. Such methodology may lead to non-optimal results for the metrics that are not optimized. Yet, we should advocate for models that can guarantee the best trade-off among all the metrics of interest.

THE COST OF THE HYPERPARAMETER OPTIMIZATION. Although optimizing the hyperparameters of a topic model guarantees a fair comparison with other models, this approach is computationally expensive. In our work, we used BO because it is more efficient than other methods (Bergstra and Bengio, 2012; Snoek et al., 2012). Yet, the process requires a fair amount of iterations to guarantee the convergence to an optimal solution, especially when the hyperparameter space is large. Moreover, we also run the models with the same hyperparameter configuration for 30 times to guarantee robust results. It follows that replicating these results require time and computational resources. In Table 7.3 we report an average estimation of the time expressed in minutes for an iteration of the hyperparameter optimization for the considered models on the 20NG and M10. The overall running time of the optimization can vary depending on the number of documents, the dimensionality of the vocabulary, on the selected hyperparameters (e.g. the number of topics), and of course on the total iterations of the MOBO. For the details on the used architecture, we refer to Appendix C.

	Datasets	
	20NG	M10
LDA	37.51	16.45
NMF	42.16	21.66
CTM	65.98	39.24

Table 7.3: Estimated minutes to complete one iteration of the MOBO for 20NG and M10 for each model.

In light of these observations, we argue that the knowledge that we have acquired for this extensive experimental campaign needs to be

exploited and transferred. This will lead to results that are more accurate and obtained more efficiently. As previously mentioned, one direction is to use the best hyperparameter configurations on a dataset to initialize the hyperparameter optimization on the unseen target dataset (Feurer et al., 2014). Here, we do not discuss this direction in detail, but we will later show that hyperparameter transfer is effective in some cases and it is therefore a promising solution to minimize the computational cost to achieve optimal results on new datasets.

7.2 HYPERPARAMETER TRANSFER

The knowledge related to optimal hyperparameter configurations, which we acquire during the multi-objective optimization, can be transferred to an unseen dataset. his can be done in a zero-shot fashion, i.e. evaluating the best hyperparameters on a dataset to a new dataset. Another option is to use the the best hyperparameter configurations on a dataset to initialize the Bayesian optimization on the target unseen dataset. Given that the configurations used to initialize the optimization are close to the optimal configurations, the optimization process will reach the optimal results in less time than with the random initialization.

Our hypothesis is that an optimal hyperparameter configuration is strongly dependent on some dataset features. We have also previously seen in Section 7.1 that the length of the documents have a different effect on topic models. To prove this hypothesis, we follow a simple and effective hyperparameter transfer approach, based on the work of Feurer et al. (2014).

Let $f_i(x)$ with $i = \{1, \dots, N\}$ denotes an objective function (here, $N = 3$) and $\gamma_i^1, \dots, \gamma_i^D$ denote the best hyperparameter configurations discovered by MOBO for the previously seen datasets $1, \dots, D$ respectively. Each γ_i^d is composed of the t best hyperparameters configuration for the objective function f_i. Feurer et al. (2014) define some dataset features, also called *metafeatures*, and a similarity measure for each feature, thus allowing to initialize the surrogate model of the BO with the best hyperparameter configurations of the dataset that is the most similar.

Here, we follow the opposite direction to show whether an optimal hyperparameter configuration is consistent across different datasets or not. We train a topic model on an unseen target dataset with the best hyperparameter configurations γ_i^d of previously seen dataset d for the given objective function f_i. If a configuration can be effectively transferred to every dataset (i.e. the best hyperparameter configuration transferred from a dataset to the target one and vice versa achieves performances that are close to the optimal ones), then it follows that the configurations are independent of the datasets' features. Otherwise, if some configurations do not transfer well on a target

dataset, it implies that the hyperparameter configurations are dependent on the metafeatures.

7.2.1 *Experimental Setting*

We divide our experiments in three settings:

S1) First, we want to verify whether it is possible to transfer the best hyperparameter configurations from a dataset to another one. In this setting, we consider the 5 best hyperparameter configurations for each metric, model, and dataset obtained during the multi-objective optimization experiments in Section 7.1. We use the identified evaluations coming from a dataset to train a topic model on a different target dataset. As before, to obtain a robust result, we train the model with the same hyperparameter configuration 30 times and consider the median of the 30 evaluations.

S2) We aim to empirically show what is the effect of using a good set of initial configurations (ideally, the ones obtained from transferring the knowledge from the previous experiments), compared to random initialization of the MOBO algorithm. Therefore, we select a set of best hyperparameter configurations and use these to initialize the MOBO process and compare them with the initial random initialization. We expect that the random initialization will require more iterations to get optimal results.

S3) We will see that our experiments suggest that the language seems to be invariant to the transfer of the hyperparameters. Therefore, we will further explore this direction by comparing the transfer of hyperparameters with a multilingual parallel dataset. Similarly to the first setting of experiments, we consider the 5 best hyperparameter configurations for each model and dataset obtained from the multi-objective optimization experiments performed on the new multilingual dataset (following the same procedure of Section 7.1, but we focus only on NPMI. We use the identified evaluations coming from a dataset to train a topic model on a different target dataset. We train the model with the same hyperparameter configuration 30 times and consider the median of the 30 evaluations.

DATASETS. For the setting S1 and S2, we consider the same datasets as the previous experiments in Section 7.1. While for the final setting (S3), we consider additional parallel datasets, along with the already used ones. In particular, we consider the dataset W1 that we have used in Section 5.3, composed of 20,000 English DBpedia abstracts. We randomly sample 5,000 documents. Since DBpedia abstracts are

not parallel translations, we use Google Translate to obtain parallel the documents in English (DB_EN), Italian (DB_IT), French (DB_FR), German (DB_DE), Romanian (DB_RO), Spanish (DB_ES) and Portuguese (DB_PT).

The additional parallel datasets have been pre-processed according to the following standard procedure: we lowercase the text and remove punctuation; we use language-specific lemmatizers to lemmatize the text and remove the stop-words according to language-specific stop-word lists; finally we remove words with a document frequency lower than 0.1% and higher than 40% and we remove the documents with less than 5 words.

Name	# Docs	# Labels	# Unique words	Avg doc length (std)
20NG	16,309	20	1612	48 (130)
AFP	26,599	17	2686	156 (174)
BBC	2,225	5	2949	120 (72)
M10	8,355	10	1696	6 (2)
SS	12,295	8	4705	14 (5)
SO	16,407	20	2257	5 (2)
DB_DE	4,996	-	3564	15 (4)
DB_EN	4,995	-	3431	20 (4)
DB_ES	4,996	-	3602	19 (4)
DB_FR	4,996	-	3685	20 (4)
DB_IT	4,992	-	3689	19 (4)
DB_PT	4,996	-	3621	19 (4)
DB_RO	4,997	-	3736	20 (4)

Table 7.4: Characteristics of the considered datasets.

7.2.2 *Results*

S1: HYPERPARAMETER CONSISTENCY ACROSS DATASETS. In the following, we report the results related to the hyperparameter transfer to an unseen dataset. This allows us to identify if the best hyperparameters are consistent across all the datasets. Figure 7.5 shows the results for LDA, CTM and NMF for each metric. Each matrix represents the performance of a model when the best 5 hyperparameter configurations are transferred from a dataset (columns) to the target dataset (rows). We compute the average of the 5 runs and we normalize each row. The diagonal of the matrix is then usually 1, since it represents the best configurations identified by the multi-objective optimization. We report the disaggregated results in the Appendix C.

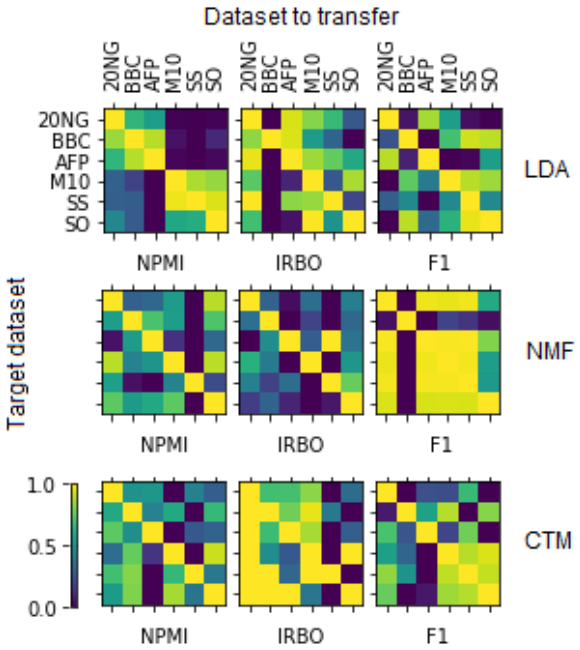


Figure 7.5: Setting S1: Heatmap matrices representing the performance when transferring the 5 best configurations from a dataset to a target dataset. The average of the runs is computed and each row is normalized between 0 and 1.

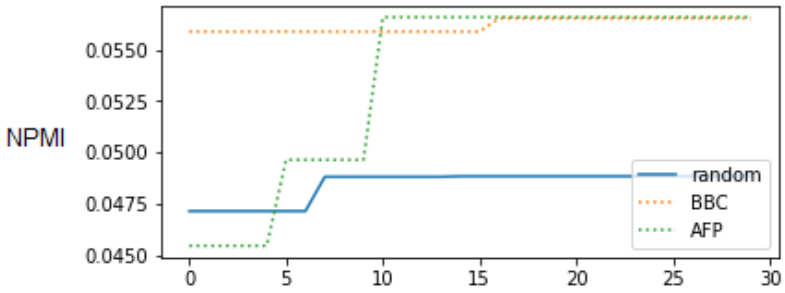
Let us consider the transfer on LDA (first three matrices on top). Regarding NPMI, when we transfer the configurations from/to 20NG, AFP, and BBC, the topic model obtains results that are similar to the ones discovered by the previous MOBO experiments. Similar observations hold for the datasets M10, SS and SO. We can therefore deduce that the document length has an impact on the discovery of the best hyperparameter configuration for topic coherence. We observe a similar behavior for the IRBO performance, with the exception of the BBC dataset. Although the values of the topic diversity are very close to each other, the long-text datasets usually get similar performance when the hyperparameters are transferred from long-text datasets, and the same holds for short-text datasets.

On the other hand, concerning the F1 performance, we observe a different trend: the configurations coming from BBC, M10, SS and SO seem to be transferable to that group of datasets, while a configuration coming from 20NG or AFP does not guarantee high performance on the previous datasets. This suggests that the document length is not the only feature that needs to be taken into consideration when we transfer the hyperparameters.

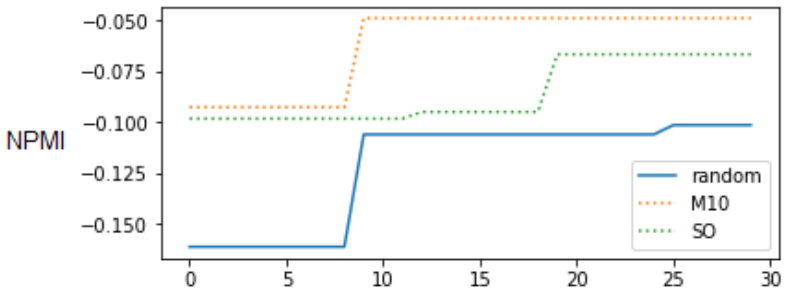
Concerning the models NMF and CTM, we can observe different patterns for each metric. For example, if we consider the topic coherence in CTM, the configurations related to datasets 20NG, AFP, BBC and SS have close performances, but are distant from M10 and SO.

On the other hand, in NMF, the datasets 20NG, M10 and SO appear to be similar to each other, and distant from the others. This might be related to the fact that the topic models are regulated by different types of hyperparameters, which have a different impact on the models' objectives.

In the considered experiments, we transfer the hyperparameters of a French dataset, i.e. AFP, to English datasets (and vice versa). The AFP's configurations transferred to another dataset can yield good results, thus suggesting that the features that make two datasets suitable for a transfer for a given metric are likely to be independent of the dataset language. This result is extremely relevant because it would allow us to transfer the known best hyperparameter configurations in low-resource settings, when the best hyperparameter configuration for a dataset in a given language is not available or is expensive to compute.



(a) 20NG dataset.



(b) SS dataset.

Figure 7.6: LDA performance in terms on NPMI with MOBO initialized with random configurations or with the configurations transferred from another dataset.

S2: RANDOM INITIAL CONFIGURATIONS VS TRANSFERRED INITIAL CONFIGURATIONS. We will now empirically show what is the effect of using a good set of initial configurations (obtained from the transfer of knowledge from previous experiments), compared to the random initialization.

Figures 7.6a and 7.6b show the NPMI performance of LDA for the first 30 MOBO iterations on the datasets 20NG and SS respectively, when the random initialization is performed (random) or when the MOBO is initialized with the best configurations deriving from another dataset. In particular, we transfer the configurations from BBC and AFP for 20NG and the configurations from M10 and SO for SS, since these are the configurations that transferred better for the considered target datasets. Since the first 5 iterations are not ordered chronologically, we just report the maximum of them.

We can observe from both figures that, when using the transferred hyperparameters, the MOBO can achieve better results in just a few iterations, outperforming the 30 iterations of the MOBO initialized with random configurations. Therefore using the transferred configurations as initial ones can be helpful to obtain good results in less iterations.

S3: HYPERPARAMETER TRANSFER AMONG PARALLEL MULTILINGUAL CORPORA. In the following, we report the results related to the hyperparameter transfer to an unseen dataset considering the multilingual parallel corpora. In this case, given the large number of experiments to perform, instead of transfer the best configurations for a single metric, we focus only on LDA and on the best configurations with respect to NPMI with a threshold on the topic diversity (we could not compute F1 on the datasets because labels are not available). Given the previous experiments on S1, we expect that if we transfer a hyperparameter configuration from a dataset to another parallel dataset, then we will maintain similar results.

Figure 7.7 shows the results. As before, the matrix represents the performance of the model when the best 5 hyperparameter configurations for NPMI (with the topic diversity threshold of 0.8) are transferred from a dataset (columns) to the target dataset (rows). We compute the average of the 5 runs and we normalize each row.

As expected, we can observe that the diagonal reaches always the best value per row, meaning that the multi-objective optimization allowed us to get an optimal result. Moreover, we can clearly observe the lighter square in the top left of the matrix. The best hyperparameters transferred from and to the multilingual datasets allow LDA to get optimal results for the NPMI. On the contrary, if we transfer from a DBpedia dataset to another non-DBpedia dataset, we rarely obtain competitive results. These results confirmed our hypothesis that the language might be an invariant for the transfer, but instead what counts is more related to the statistics of the words and documents. It would be worth investigating in detail which are the main important features that allows for a good transfer from a dataset to another unseen dataset.

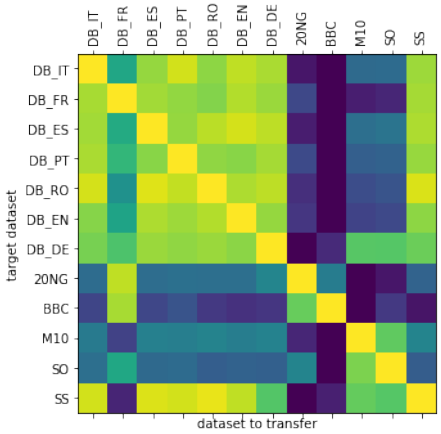


Figure 7.7: Setting S3: Heatmap matrix representing the performance when transferring the 5 best configurations from a dataset to a target dataset. The average of the runs is computed and each row is normalized between 0 and 1.

7.3 SUMMARY OF THIS CHAPTER

In the following, we give a short summary of this chapter. In the introduction of the chapter, we reported the following research questions:

Q7.1: Can we optimize the hyperparameters of a topic model to guarantee an optimal trade-off between different performance measures using multi-objective optimization?

Q7.2: Can we transfer the best hyperparameter configurations from a dataset to an unseen dataset?

To answer Q7.1, we investigated the role of a multi-objective optimization approach in topic models. We saw that multi-objective optimization is effective to discover the best trade-off between different metrics. Moreover, as already confirmed by the experiments of the previous Chapter, our results show that when we boost the models' performance at the best of their capabilities, it is not possible to identify an always-winning topic model for each considered objective, thus raising a question on the fairness of the past evaluations and comparisons between topic models. This result is further enforced when additional objectives are jointly considered.

Regarding Q7.2, we showed that, in some cases, it is possible to effectively transfer the hyperparameters from a dataset to another. This result paves the way to exciting future research directions. In fact, the hyperparameter transfer allows researchers to avoid several and expensive iterations of hyperparameter optimization. In fact, we also empirically show that we can use the transferred configurations to initialize the Bayesian Optimization process, reaching optimal results with fewer iterations.

It is also worth further investigating which dataset features contribute to a configuration transferable from a dataset to another. Our results suggest that the document length plays a role in the transfer, but other features such as the word and class distributions could be important too. We also showed that the dataset features are likely to be independent of the dataset language, leading to the use of hyperparameter transfer even to unseen datasets in different and low-resource languages.

8

CONCLUSIONS

In this thesis, we have defined and compared different methods for incorporating information into topic models. We have started with methods that incorporate document-level and word-level relational information into classical probabilistic topic models. Our results have shown that we can improve the quality of topics and the representation of the documents.

We also investigated the incorporation of context information into neural topic models, to overcome their limitations related to the BoW assumption. We used contextualized document representations that allowed us to both improve the quality of the topics and address cross-lingual tasks.

We also focused our attention on the evaluation of topic models, proposing a comprehensive framework for evaluating and comparing topic models, based on single-objective or multi-objective hyperparameter Bayesian Optimization. This method is generalizable to any type of topic model and hyperparameters, and allowed us to investigate the relationships between evaluation metrics, models, hyperparameters, and datasets. Although this method can handle expensive objective functions, it still requires high computational resources. We, therefore, explored the possibility of transferring the best hyperparameters configurations from a dataset to an unseen dataset.

I would also like to share my perspective on the accessibility and usage of topic models. Latent Dirichlet Allocation with over 40'000 citations is by far the most cited and used topic model. Several models and approaches have been proposed over the years (including the ones proposed in this thesis), much more sophisticated and complex than LDA. Each time a researcher proposes a new topic model, they claim to have surpassed the state of the art. Yet, everyone continues to use LDA, even in contexts where we know it will not perform optimally (Xue et al., 2020). Researchers and practitioners who are not involved in the topic modeling field might not be aware of the progress of this active and evolving field. State-of-the-art topic modeling implementations are not always publicly released and, even if they are, they are not necessarily accessible to everyone (or to most of the audience of the practitioners). Unfortunately, this is a relevant issue not only in the topic modeling field but also in the NLP community (Bianchi and Hovy, 2021).

While keeping this issue in mind, I tried to provide and release topic models and libraries that are in fact accessible. I have just started to see the effects of this decision. The libraries I contributed to have

just reached over 1025 GitHub stars and over 120'000 downloads and this assures me that we have been following the right direction. Hopefully, this will also lead the research in this field and related fields to progress more quickly and efficiently.

FUTURE RESEARCH DIRECTIONS This thesis opens up to different research directions, for example,

- as already mentioned, the approaches based on the modeling of potential functions are modular and easy to implement. They could be applied to a wide variety of classical topics models, with slight modifications. Our experiments were limited to specific sources of information, but incorporating and modeling other sources of information, both domain-specific or general, could be investigated as well.
- We have shown that topic models can benefit from the use of contextualized representations. Research about language models and document representations is rapidly evolving and growing. We are confident that our contextualized topic models can further benefit from the novel and better contextualized embeddings that will be proposed in the future.
- Another exciting direction involves the transfer learning capabilities deriving from the use of contextualized language models. We have seen that we can use multilingual representations to train a topic model on a language and predict the topics on unseen languages. These results can be further explored both considering low-resource languages and other modalities (e.g. images).
- Our experiments of comparison between different models raise a question on the fairness of the past evaluations. This contributes to the growing amount of negative results when reviewing previously published work in light of new experiments. We agree with Zhao et al. (2021) about the critical need of a benchmarks for topic modeling.
- Finally, the choice of the right hyperparameters of a topic model is still an open challenge. Bayesian Optimization can be still expensive and hyperparameter transfer seems to be a promising direction. However, investigating which dataset features contribute to a good transfer from a dataset to another is essential.

BIBLIOGRAPHY

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Albalawi, R., Yeap, T. H., and Benyoucef, M. (2020). Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence*, 3:42.

Aletras, N. and Stevenson, M. (2013). Evaluating Topic Coherence Using Distributional Semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS)*, pages 13–22.

Aletras, N. and Stevenson, M. (2014). Measuring the Similarity between Automatically Generated Topics. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 22–27. The Association for Computer Linguistics.

Ali, M. (2020). *PyCaret: An open source, low-code machine learning library in Python*. PyCaret version 2.3.

Allahyari, M. and Kochut, K. (2016). Discovering coherent topics with entity topic models. In *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016*, pages 26–33.

AlSumait, L., Barbará, D., Gentle, J., and Domeniconi, C. (2009). Topic significance ranking of LDA generative models. In *Joint European conference on machine learning and knowledge discovery in databases*, volume 5781 of *Lecture Notes in Computer Science*, pages 67–82. Springer, Springer.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, pages 25–32.

Andrzejewski, D., Zhu, X., Craven, M., and Recht, B. (2011). A Framework for Incorporating General Domain Knowledge into Latent Dirichlet Allocation Using First-Order Logic. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1171–1177.

Archetti, F. and Candelieri, A. (2019). *Bayesian Optimization and Data Science*. Springer International Publishing.

Artetxe, M., Ruder, S., and Yogatama, D. (2020). On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 4623–4637. Association for Computational Linguistics.

Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34, Montreal, QC, Canada.

Bai, H., Chen, Z., Lyu, M. R., King, I., and Xu, Z. (2018). Neural models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, pages 27–36.

Bao, S., Xu, S., Zhang, L., Yan, R., Su, Z., Han, D., and Yu, Y. (2009). Joint emotion-topic modeling for social affective text mining. In *2009 Ninth IEEE International Conference on Data Mining*, pages 699–704. IEEE.

Barnes, J., Klinger, R., and im Walde, S. S. (2018). Bilingual sentiment embeddings: Joint projection of sentiment across languages. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018s*, pages 2483–2493. Association for Computational Linguistics.

Basu, S., Bilenko, M., and Mooney, R. J. (2004). A Probabilistic Framework for Semi-Supervised Clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68.

Batmanghelich, K., Saeedi, A., Narasimhan, K., and Gershman, S. (2016). Nonparametric Spherical Topic Modeling with Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, page 537.

Belford, M. and Greene, D. (2019). Comparison of embedding techniques for topic modeling coherence measures. In *Proceedings of the Poster Session of the 2nd Conference on Language,Data and Knowledge (LDK 2019)*, volume 2402 of *CEUR Workshop Proceedings*, pages 1–5. CEUR-WS.org.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). *Neural Probabilistic Language Models*, pages 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2):281–305.

Bhatia, S., Lau, J. H., and Baldwin, T. (2016). Automatic labelling of topics with neural embeddings. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pages 953–963. ACL.

Bianchi, F., Attanasio, G., Pisoni, R., Terragni, S., Sarti, G., and Lakshmi, S. (2021a). Contrastive language-image pre-training for the italian language. *arXiv preprint arXiv:2108.08688*.

Bianchi, F. and Hovy, D. (2021). On the gap between adoption and understanding in nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3895–3901.

Bianchi, F., Terragni, S., and Hovy, D. (2021b). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pages 759–766. Association for Computational Linguistics.

Bianchi, F., Terragni, S., Hovy, D., Nozza, D., and Fersini, E. (2021c). Cross-lingual Contextualized Topic Models with Zero-shot Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134. ACM.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006)*, volume 148 of *ACM International Conference Proceeding Series*, pages 113–120. ACM.

Blei, D. M. and Lafferty, J. D. (2009). Visualizing topics with multi-word expressions. *arXiv preprint arXiv:0907.1013*.

Blei, D. M. and McAuliffe, J. D. (2007). Supervised Topic Models. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 121–128.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003a). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003b). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Boyd-Graber, J. L. and Blei, D. M. (2008). Syntactic Topic Models. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 185–192.

Boyd-Graber, J. L. and Blei, D. M. (2009). Multilingual topic models for unaligned text. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, pages 75–82. AUAI Press.

Boyd-Graber, J. L., Hu, Y., and Mimno, D. M. (2017). Applications of topic models. *Found. Trends Inf. Retr.*, 11(2-3):143–296.

Boyd-Graber, J. L., Mimno, D., and Newman, D. (2014). *Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements*. CRC Handbooks of Modern Statistical Methods. CRC Press.

Buntine, W. L. (2009). Estimating Likelihoods for Topic Models. In *Proceedings of the 1st Asian Conference on Machine Learning, (ACML)*, pages 51–64.

Burkhardt, S. and Kramer, S. (2019). Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model. *Journal of Machine Learning Research*, 20(131):1–27.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Candelieri, A. and Archetti, F. (2019). Global optimization in machine learning: the design of a predictive analytics application. *Soft Computing*, 23(9):2969–2977.

Carrow, S. (2018). PyTorchAVITM: Open Source AVITM Implementation in PyTorch. Github.

Casella, G. and Robert, C. P. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Association for Computational Linguistics.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations*, pages 169–174. Association for Computational Linguistics.

Chaney, A. J. and Blei, D. M. (2012). Visualizing Topic Models. In *Proceedings of the 6th International Conference on Weblogs and Social Media*. The AAAI Press.

Chang, J. and Blei, D. (2009). Relational topic models for document networks. In *Artificial intelligence and statistics*, pages 81–88. PMLR.

Chang, J., Boyd-Graber, J. L., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 288–296.

Chauhan, U. and Shah, A. (2021). Topic modeling using latent dirichlet allocation: A survey. *ACM Computing Surveys (CSUR)*, 54(7):1–35.

Chen, N., Zhu, J., Xia, F., and Zhang, B. (2013a). Generalized relational topic models with data augmentation. In *Twenty-Third International Joint Conference on Artificial Intelligence*, pages 1273–1279.

Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., and Ghosh, R. (2013b). Discovering coherent topics using general knowledge. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13*, pages 209–218. ACM.

Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., and Ghosh, R. (2013c). Leveraging Multi-Domain Prior Knowledge in Topic Models. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2071–2077.

Chuang, J., Manning, C. D., and Heer, J. (2012). Termite: visualization techniques for assessing textual topic models. In Tortora, G., Levialdi, S., and Tucci, M., editors, *International Working Conference on Advanced Visual Interfaces, AVI 2012, Capri Island, Naples, Italy, May 22-25, 2012, Proceedings*, pages 74–77. ACM.

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

Das, R., Zaheer, M., and Dyer, C. (2015). 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 of the Asian Federation of Natural Language Processing, ACL 2015, pages 795–804. The Association for Computational Linguistics.

Daud, A., Shaikh, A. M. A. R., and Rajpar, A. H. (2009). Scientific reference mining using semantic information through topic modeling. *Research Journal of Engineering & Technology*, 28(2):253–262.

Deng, F., Siersdorfer, S., and Zerr, S. (2012). Efficient jaccard-based diversity analysis of large document collections. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1402–1411.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dieng, A. B., Ruiz, F. J. R., and Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Ding, R., Nallapati, R., and Xiang, B. (2018). Coherence-Aware Neural Topic Modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 830–836, Brussels, Belgium. Association for Computational Linguistics.

Doan, T. and Hoang, T. (2021). Benchmarking neural topic models: An empirical study. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 4363–4368. Association for Computational Linguistics.

Doogan, C. and Buntine, W. L. (2021). Topic model or topic twaddle? re-evaluating semantic interpretability measures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 3824–3848. Association for Computational Linguistics.

Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proc. of the 23rd International Conference on Computational Linguistics*, pages 277–285.

Erosheva, E., Fienberg, S., and Lafferty, J. (2004). Mixed-membership Models of Scientific Publications. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5220–5227.

Fei, G., Chen, Z., and Liu, B. (2014). Review Topic Discovery with Phrases using the Pólya Urn Model. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 667–676.

Ferrone, L. and Zanzotto, F. M. (2017). Symbolic, distributed and distributional representations for natural language processing in the era of deep learning: a survey. *arXiv preprint arXiv:1702.00764*.

Fersini, E., Messina, E., Felici, G., and Roth, D. (2014). Soft-constrained inference for Named Entity Recognition. *Information Processing & Management*, 50(5):807–819.

Feurer, M., Springenberg, J., and Hutter, F. (2015). Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1128–1135.

Feurer, M., Springenberg, J. T., and Hutter, F. (2014). Using Meta-Learning to Initialize Bayesian Optimization of Hyperparameters. In *Proceedings of the International Workshop on Meta-learning and Algorithm Selection, co-located with 21st European Conference on Artificial Intelligence (MetaSel@ECAI 2014)*, volume 1201 of *CEUR Workshop Proceedings*, pages 3–10, Prague, Czech Republic. CEUR-WS.org.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *CoRR*, abs/1807.02811.

George, C. P., Doss, H., et al. (2017). Principled selection of hyperparameters in the latent dirichlet allocation model. *J. Mach. Learn. Res.*, 18(1):5937–5974.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Greene, D. and Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Greene, D., O'Callaghan, D., and Cunningham, P. (2014). How many topics? stability analysis for topic models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Proceedings, Part I*, volume 8724 of *Lecture Notes in Computer Science*, pages 498–513. Springer.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Griffiths, T. L., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. (2004). Integrating Topics and Syntax. In *Advances in Neural Information Processing Systems 18 (NIPS 2004)*, pages 537–544.

Gruber, A., Weiss, Y., and Rosen-Zvi, M. (2007). Hidden Topic Markov Models. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 163–170.

Guo, W., Wu, S., Wang, L., and Tan, T. (2015). Social- model for social networks. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, pages 1731–1734.

Gupta, P., Chaudhary, Y., Buettner, F., and Schütze, H. (2019). Document informed neural autoregressive topic models with distributional prior. In *AAAI2019*, pages 6505–6512. AAAI Press.

Gupta, P., Chaudhary, Y., Runkler, T., and Schuetze, H. (2020). Neural topic modeling with continual lifelong learning. In *International Conference on Machine Learning*, pages 3907–3917. PMLR.

Gutiérrez, E., Shutova, E., Lichtenstein, P., de Melo, G., and Gilardi, L. (2016). Detecting cross-cultural differences using a multilingual topic model. *Transactions of the Association for Computational Linguistics*, 4:47–60.

Gwet, K. L. (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.

Hao, S. and Paul, M. J. (2018). Learning multilingual topics from incomparable corpora. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 2595–2609. Association for Computational Linguistics.

Harrando, I., Lisena, P., and Troncy, R. (2021). Apples to apples: A systematic evaluation of topic models. RANLP.

He, Y., Wang, C., and Jiang, C. (2017). Modeling document networks with tree-averaged copula regularization. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 691–699.

Head, T., MechCoder, G. L., Shcherbatyi, I., et al. (2018). scikit-optimize/scikit-optimize: v0. 5.2.

Hefny, A., Gordon, G., and Sycara, K. (2013). Random Walk Features for Network-aware Topic Models. In *NIPS 2013 Workshop on Frontiers of Network Analysis*, volume 6.

Heyman, G., Vulic, I., and Moens, M. (2016). C-BiLDA extracting cross-lingual topics from non-parallel texts by distinguishing shared from unshared content. *Data Mining and Knowledge Discovery*, 30(5):1299–1323.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Distributed Representations, pages 77–109. MIT Press.

Hoffman, M. D., Blei, D. M., and Bach, F. R. (2010). Online Learning for Latent Dirichlet Allocation. In *Proceedings in the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 856–864.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM.

Horn, D. and Bischl, B. (2016). Multi-objective parameter configuration of machine learning algorithms using model-based optimization. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Hovy, D., Bianchi, F., and Fornaciari, T. (2020). "You sound just like your father" Commercial machine translation systems include stylistic biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1686–1690, Online. Association for Computational Linguistics.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Hu, Y., Boyd-Graber, J. L., Satinoff, B., and Smith, A. (2014). Interactive topic modeling. *Machine Learning*, 95(3):423–469.

Jagarlamudi, J. and Daumé, H. (2010). Extracting multilingual topics from unaligned comparable corpora. In *Advances in Information Retrieval*, pages 444–456, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jelodar, H., Wang, Y., Yuan, C., and Feng, X. (2018). Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78:15169–15211.

Jomaa, H. S., Schmidt-Thieme, L., and Grabocka, J. (2020). Zero-shot transfer learning for gray-box hyper-parameter optimization.

Kandasamy, K., Vysyaraju, K. R., Neiswanger, W., Paria, B., Collins, C. R., Schneider, J., Póczos, B., and Xing, E. P. (2020). Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *Journal of Machine Learning Research*, 21:81:1–81:27.

Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., and Rudzicz, F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics: X*, 4:100057.

Kim, H., Sun, Y., Hockenmaier, J., and Han, J. (2012). ETM: Entity Topic Models for Mining Documents Associated with Entities. In *Proceedings of the 12th IEEE International Conference on Data Mining, ICDM 2012*, pages 349–358.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Korencic, D., Ristov, S., and Snajder, J. (2018). Document-based topic coherence measures for news media text. *Expert Systems with Applications*, 114:357–373.

Kou, W., Li, F., and Baldwin, T. (2015). Automatic labelling of topic models using word vectors and letter trigram vectors. In *Information Retrieval Technology - 11th Asia Information Retrieval Societies Conference, AIRS 2015*, volume 9460 of *Lecture Notes in Computer Science*, pages 253–264. Springer.

Krstovski, K., Smith, D. A., and Kurtz, M. J. (2016). Online multilingual topic models with multi-level hyperpriors. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016*, pages 454–459. Association for Computational Linguistics.

Lacoste-Julien, S., Sha, F., and Jordan, M. I. (2008). Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 897–904. Curran Associates, Inc.

Larochelle, H. and Lauly, S. (2012). A neural autoregressive topic model. In *Advances in Neural Information Processing Systems 25:*

26th Annual Conference on Neural Information Processing Systems 2012, pages 2717–2725.

Lau, J. H., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic labelling of topic models. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1536–1545. The Association for Computer Linguistics.

Lau, J. H., Newman, D., and Baldwin, T. (2014a). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 530–539.

Lau, J. H., Newman, D., and Baldwin, T. (2014b). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, pages 556–562. MIT Press.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Li, C., Duan, Y., Wang, H., Zhang, Z., Sun, A., and Ma, Z. (2017). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems (TOIS)*, 36(2):1–30.

Li, J., Sun, A., Han, J., and Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Li, X., Ouyang, J., and Zhou, X. (2015). Supervised topic models for multi-label classification. *Neurocomputing*, 149:811–819.

Lim, K. W. and Buntine, W. (2015). Bibliographic analysis with the citation network topic model. In *Asian conference on machine learning*, pages 142–158. PMLR.

Lim, K. W. and Buntine, W. L. (2014). Bibliographic Analysis with the Citation Network Topic Model. In *Proceedings of the Sixth Asian Conference on Machine Learning (ACML)*.

Lindsey, R. V., Headden, W., and Stipicevic, M. (2012). A Phrase-Discovering Topic Model Using Hierarchical Pitman-Yor Processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 214–222.

Lisena, P., Harrando, I., Kandakji, O., and Troncy, R. (2020). TomodAPI: A Topic Modeling API to Train, Use and Compare Topic Models. In *2nd International Workshop for Natural Language Processing Open Source Software (NLP-OSS)*.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 1073–1094. Association for Computational Linguistics.

Liu, X., Duh, K., and Matsumoto, Y. (2015). Multilingual topic models for bilingual dictionary extraction. *ACM Transactions on Asian Language Information Processing*, 14(3):11:1–11:22.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ma, T. and Nasukawa, T. (2017). Inverted bilingual topic models for lexicon extraction from non-parallel data. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 4075–4081. IJCAI.org.

Manning, C. D. and Schütze, H. (2001). *Foundations of statistical natural language processing*. MIT Press.

McCallum, A., Corrada-Emmanuel, A., and Wang, X. (2005). Topic and role discovery in social networks. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05*, pages 786–791.

McCallum, A., Wang, X., and Corrada-Emmanuel, A. (2007). Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. *J. Artif. Intell. Res.*, 30:249–272.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*.

Mei, Q., Cai, D., Zhang, D., and Zhai, C. (2008). Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, pages 101–110.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems, I-SEMANTICS 2011*, ACM International Conference Proceeding Series, pages 1–8. ACM.

Miao, Y., Grefenstette, E., and Blunsom, P. (2017). Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2410–2419. PMLR.

Miao, Y., Yu, L., and Blunsom, P. (2016). Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1727–1736. JMLR.org.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mimno, D. M. and McCallum, A. (2007). Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 500–509. ACM.

Mimno, D. M. and McCallum, A. (2008). Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, volume 24, pages 411–418. Citeseer.

Mimno, D. M., Wallach, H. M., Naradowsky, J., Smith, D. A., and McCallum, A. (2009). Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, pages 880–889. Association for Computational Linguistics.

Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1791–1799. JMLR.org.

Mukherjee, A. and Liu, B. (2012). Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 339–348.

Murdock, J. and Allen, C. (2015). Visualization techniques for topic model checking. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic Evaluation of Topic Coherence. In *Proceedings of the Conference of the*

North American Chapter of the Association of Computational Linguistics (NAACL),, pages 100–108.

Nguyen, D. Q., Billingsley, R., Du, L., and Johnson, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Nozza, D., Bianchi, F., and Hovy, D. (2020). What the [mask]? making sense of language-specific bert models. *arXiv preprint arXiv:2003.02912*.

Nozza, D., Sas, C., Fersini, E., and Messina, E. (2019). Word embeddings for unsupervised named entity linking. In *International Conference on Knowledge Science, Engineering and Management*, pages 115–132. Springer.

O'Callaghan, D., Greene, D., Carthy, J., and Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Syst. Appl.*, 42(13):5645–5657.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

Paisley, J., Wang, C., Blei, D. M., and Jordan, M. I. (2014). Nested hierarchical dirichlet processes. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):256–270.

Paria, B., Kandasamy, K., and Póczos, B. (2019). A Flexible Framework for Multi-Objective Bayesian Optimization using Random Scalarizations. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 115 of *Proceedings of Machine Learning Research*, pages 766–776. AUAI Press.

Parisi, G. (1988). *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley.

Pavlinek, M. and Podgorelec, V. (2017). Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, 80:83–93.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543. ACL.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019*, pages 7–14. Association for Computational Linguistics.

Phan, X. H., Nguyen, M. L., and Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, pages 91–100. ACM.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Popa, C. and Rebedea, T. (2021). BART-TL: weakly-supervised topic label generation. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*, pages 1418–1425. Association for Computational Linguistics.

Qiang, J., Li, Y., Yuan, Y., Liu, W., and Wu, X. (2018). Sttm: A tool for short text topic modeling. *arXiv preprint arXiv:1808.02215*.

Qiang, J., Zhenyu, Q., Li, Y., Yuan, Y., and Wu, X. (2019). Short text topic modeling techniques, applications, and performance: A survey. *arXiv preprint arXiv:1904.07695*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rahimi, A., Li, Y., and Cohn, T. (2019). Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 151–164. Association for Computational Linguistics.

Ramage, D., Hall, D. L. W., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 248–256.

Ramage, D., Manning, C. D., and Dumais, S. T. (2011). Partially labeled topic models for interpretable text mining. In *Proceedings of*

the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 457–465.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Rezaeinia, S. M., Rahmani, R., Ghodsi, A., and Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139–147.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining, WSDM 2015*, pages 399–408. ACM.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020a). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rogers, A., Sedoc, J., and Rumshisky, A., editors (2020b). *Proceedings of the First Workshop on Insights from Negative Results in NLP*, Online. Association for Computational Linguistics.

Rosen-Zvi, M., Griffiths, T. L., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press.

Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway.

Salakhutdinov, R. and Hinton, G. E. (2009). Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 1607–1614. Curran Associates, Inc.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Schuster, T., Ram, O., Barzilay, R., and Globerson, A. (2019). Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 1599–1613. Association for Computational Linguistics.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Sievert, C. and Shirley, K. (2014). Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, volume 25, pages 2951–2959.

Søgaard, A., Vulić, I., Ruder, S., and Faruqui, M. (2019). Cross-lingual word embeddings. *Synthesis Lectures on Human Language Technologies*, 12(2):1–132.

Srivastava, A. and Sutton, C. (2017). Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017*.

Stevens, K., Kegelmeyer, W. P., Andrzejewski, D., and Buttler, D. (2012). Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012*, pages 952–961. ACL.

Sun, S. (2013). A Review of Deterministic Approximate Inference Techniques for Bayesian Machine Learning. *Neural Computing and Applications*, 23(7-8):2039–2050.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2004). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems, 17*, pages 1385–1392.

Terragni, S., Fersini, E., Galuzzi, B. G., Tropeano, P., and Candelieri, A. (2021a). OCTIS: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270, Online. Association for Computational Linguistics.

Terragni, S., Fersini, E., and Messina, E. (2020). Constrained relational topic models. *Information Sciences*, 512:581–594.

Terragni, S., Fersini, E., and Messina, E. (2021b). Word embedding-based topic similarity measures. In *Natural Language Processing and Information Systems - 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021*. Springer.

Tran, N. K., Zerr, S., Bischoff, K., Niederée, C., and Krestel, R. (2013). Topic cropping: Leveraging latent topics for the analysis of small corpora. In *Research and Advanced Technology for Digital Libraries - International Conference on Theory and Practice of Digital Libraries, TPDL 2013*, volume 8092, pages 297–308. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008.

Vayansky, I. and Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, 94:101582.

Wahabzada, M., Xu, Z., and Kersting, K. (2010). Topic models conditioned on relations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 402–417. Springer.

Wallach, H. M. (2006). Topic Modeling: Beyond Bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 977–984.

Wallach, H. M. (2008). *Structured topic models for language*. PhD thesis, University of Cambridge.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. M. (2009). Evaluation Methods for Topic Models. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1105–1112.

Wang, Q., Song, D., and Li, X. (2017). Incorporating entity correlation knowledge into topic modeling. In *Proceedings of the IEEE International Conference on Big Knowledge, ICBK 2017*, pages 254–258.

Wang, X. and McCallum, A. (2006). Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433. ACM.

Wang, X., McCallum, A., and Wei, X. (2007). Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 697–702.

Webber, W., Moffat, A., and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Wu, S. and Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844.

Wu, X., Li, C., Zhu, Y., and Miao, Y. (2020). Learning multilingual topics with neural variational inference. In *Natural Language Processing and Chinese Computing - 9th CCF International Conference, NLPCC 2020*, volume 12430 of *Lecture Notes in Computer Science*, pages 840–851. Springer.

Xue, J., Chen, J., Chen, C., Zheng, C., Li, S., and Zhu, T. (2020). Public discourse and sentiment during the covid 19 pandemic: Using latent dirichlet allocation for topic modeling on twitter. *PloS one*, 15(9):e0239441.

Yamada, I., Asai, A., Shindo, H., Takeda, H., and Takefuji, Y. (2018). Wikipedia2vec: an optimized tool for learning embeddings of words and entities from wikipedia. *arXiv preprint arXiv:1812.06280*.

Yang, W., Boyd-Graber, J., and Resnik, P. (2016a). A discriminative topic model using document network structure. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 686–696.

Yang, W., Boyd-Graber, J. L., and Resnik, P. (2015a). Birds of a feather linked together: A discriminative topic model using link-based priors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 261–266.

Yang, W., Boyd-Graber, J. L., and Resnik, P. (2016b). A discriminative topic model using document network structure. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Ábrego, G. H., Yuan, S., Tar, C., Sung, Y., Strope, B., and Kurzweil, R. (2020). Multilingual universal sentence encoder for semantic retrieval. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, pages 87–94. Association for Computational Linguistics.

Yang, Y., Downey, D., and Boyd-Graber, J. (2015b). Efficient methods for incorporating knowledge into topic models. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 308–317.

Yang, Y., Downey, D., and Boyd-Graber, J. L. (2015c). Efficient methods for incorporating knowledge into topic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 308–317. The Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 5754–5764.

Zhai, C. (2017). Probabilistic Topic Models for Text Data Retrieval and Analysis. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1399–1401.

Zhai, K., Boyd-Graber, J., Asadi, N., and Alkhouja, M. L. (2012). Mr. lda: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the 21st international conference on World Wide Web*, pages 879–888.

Zhang, A., Zhu, J., and Zhang, B. (2013). Sparse relational topic models for document networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 670–685. Springer.

Zhao, B. and Xing, E. P. (2006). Bitam: Bilingual topic admixture models for word alignment. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. The Association for Computer Linguistics.

Zhao, H., Du, L., Buntine, W., and Liu, G. (2017). Metalda: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 635–644. IEEE.

Zhao, H., Phung, D. Q., Huynh, V., Jin, Y., Du, L., and Buntine, W. L. (2021). Topic modelling meets deep neural networks: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 4713–4720. ijcai.org.

Zhu, J., Ahmed, A., and Xing, E. P. (2012). Medlda: maximum margin supervised topic models. *J. Mach. Learn. Res.*, 13:2237–2278.

Zhu, Y., Yan, X., Getoor, L., and Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013*, pages 473–481.

Part IV

APPENDIX

A

PROBABILISTIC GRAPHICAL MODELS

In this Chapter, we will review some of the main notions related to probabilistic graphical modeling. For a detailed analysis of probabilistic graphical models, we refer the reader to (Koller and Friedman, 2009).

Probabilistic graphical models (PGM) are a rich framework for encoding probability distributions. A PGM approach to learning from data is to imagine and mimic the true phenomenon that generated the data. We will refer to this process as to "generative process". Usually the phenomenon that generated the data is unknown and we want to learn more about it. A PGM assumes the existence of a set of latent (or unobserved) variables that represent the hidden structure underlying the observed data. In the case of topic modeling, the underlying topics are the latent variable to infer.

A.1 PLATE NOTATION

Probabilistic graphical models use a graph-based representation as the basis for encoding a complex distribution. In this graphical representation, the nodes (circles) correspond to the random variables (or the priors over the random variables), and the edges correspond to probabilistic interactions between them, i.e. conditional dependencies. A variable may be replicated for multiple times. To express this idea, the variable may be inserted into a plate which indicates how many times the variable may be replicated. Moreover, the circles may be shaded, denoting an observed random variable, or not shaded, denoting an unobserved random variable.

We will illustrate this notation in Figure A.1 through the example of Latent Dirichlet Allocation (Blei et al., 2003a, LDA), which will be extensively used throughout the thesis. In this context, knowing the meaning of each variable is not relevant. We can just notice that w_{nd} is an observed random variable, while the others are unobserved. Moreover, the variables w_{nd} and z_{nd} are replicated for N_d times. Analogously, θ is replicated for D times and ϕ is replicated for K times. α and β are the priors of θ and ϕ respectively.

A.2 JOINT DISTRIBUTION

Given the graphical representation of a model, we can deduce the joint distribution of the probabilistic model. We can do this because the model clearly represents the conditional dependency relation-

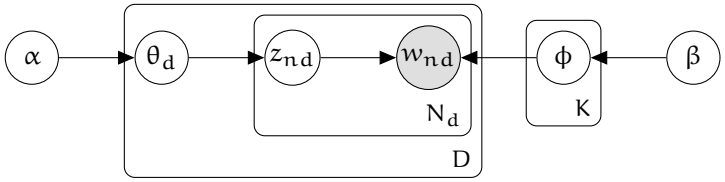


Figure A.1: LDA in plate notation.

ships between the random variables. Therefore, the joint probability distribution will be as follows:

$$p(\theta, \mathbf{z}, \boldsymbol{\phi}, \mathbf{w}|\alpha, \beta) = p(\boldsymbol{\phi}|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|z, \phi) \tag{57}$$

which can be rewritten as follows, by replicating the variables:

$$p(\theta, \mathbf{z}, \boldsymbol{\phi}, \mathbf{w}|\alpha, \beta) = \tag{58}$$

$$\prod_{k=1}^{K} p(\phi_k|\beta) \cdot \prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_{nd}|\theta_d)p(w_{nd}|z_{nd}, \phi_{z_{nd}}) \tag{59}$$

A.3 POSTERIOR INFERENCE

The latent random variables, which have ideally generated the data, are then inferred by reversing the generative process. This process is called "inference".

Inference in PGM computes the posterior distribution, which is the distribution of the latent variables after taking into account the observed data. This is determined by Bayes' rule:

$$P(h|X, \alpha) = \frac{P(X|h, \alpha) \cdot (h|\alpha)}{P(X|\alpha)} \tag{60}$$

where h represents the latent variables to infer, X the data (or evidence) and α represent the priors over the random variables (or hyperparameters). In topic models, the posterior is usually intractable because not obtained in a closed form distribution, so we need to use approximate methods. In Section 2.4, we report two of the most common ones.

A.4 CONJUGACY OF PROBABILITY DISTRIBUTIONS

It is often convenient to assume that the prior distribution of a random variable comes from a family of distributions called conjugate priors. The usefulness of a conjugate prior is that the corresponding posterior distribution will be in the same family, and the calculation may be expressed in closed form.

For example, the Dirichlet distribution is the conjugate prior distribution of the categorical distribution (a generic discrete probability

distribution with a given number of possible outcomes) and multinomial distribution (the distribution over observed counts of each possible category in a set of categorically distributed observations). For a detailed explanation of the probability distributions, we refer the reader to (Murphy, 2012).

B

TOPIC SIMILARITY MEASURES

When a topic model automatically generates a set of topics underlying a given corpus, few of them could be similar while others could be different. For instance, a topic about technology, characterized by the words “card video monitor cable vga", is more similar to the topic “gif image format jpeg color" than one about animals (“cat animal dog cats tiger"). Methods for automatically determining the similarity between topics have several potential applications, such as the validation of the quality of the topic modeling output for determining potential overlaps between pairs of topics (AlSumait et al., 2009) and document retrieval based on topic proximity (Boyd-Graber et al., 2017).

To estimate the similarity between topics, several metrics have been introduced in the state of the art. Most of them are based on word tokens and usually adopt a list of top-N terms to estimate if two topics are related. On the other hand, few approaches exploit the probability distribution of the words denoting the topics to compute the similarity between themes. These distribution-based measures suffer from the high dimensionality of the vocabulary, generating solutions that do not strongly correlate with human judgment (Aletras and Stevenson, 2014). On the contrary, approaches that focus only on the word tokens of a topic (Bianchi et al., 2021b; Tran et al., 2013) ignore that two words could be lexicographically different but denoting a similar meaning. For instance, the words *cat* and *kitten* should not be considered totally dissimilar. A preliminary investigation that partially addressed the above problems has been introduced in (Aletras and Stevenson, 2014). They represent the words of a topic as vectors in a semantic space constructed from an external source or from the corpus using Pointwise Mutual Information (PMI). However, this approach is computationally expensive, requiring to compute the probability of the co-occurrence for each pair of words in the corpus, and does not take into account the more recent advances in Word Embeddings (Grave et al., 2018; Mikolov et al., 2013; Pennington et al., 2014), that have already proved their benefits in several NLP applications and topic modeling (Batmanghelich et al., 2016; Nguyen et al., 2015). Moreover, this approach does not take into account that the topics extracted are actually ranked lists of words, where the rank provides useful insight. In particular, if two topics contain the same words but at different ranking positions, this aspect should be considered when evaluating the similarity of the generated solution.

In this Chapter, we will therefore propose new topic similarity metrics that exploit the nature of word embeddings and take into consideration topics as ranked lists of words. We demonstrate in the experimental evaluation that these metrics can discover semantically similar topics, also outperforming the state-of-the-art topic similarity metrics.

B.1 Topic Similarity / Distance Measures: State of the Art

Recall that each topic in probabilistic topic models is represented as a multinomial distribution over the vocabulary, usually referred to as *word-topic distribution*. Researchers usually consider the top-N most probable words (from the word-topic distribution) to represent a topic. This top-N ranked list of words can be called *topic descriptor* (Belford and Greene, 2019). The word-topic distribution and topic descriptors are the two key elements that can be exploited to estimate the similarity between two themes.

The topic descriptor of a topic i will be referred to as t_i, represented by its top-N most likely words, i.e. $t_i = \{v_0, v_1, \ldots, v_{N-1}\}$, where v_k is a word of the vocabulary V. We will refer to the word distribution of a topic i as ϕ_i, which is a multinomial distribution over the vocabulary V. In particular, ϕ_{iv} represents the probability of the word v in the topic i.

In Section 3.4 we have already mentioned the existing metrics for estimating the similarity or diversity of the topics. We can roughly divide them into metrics that are based on the counts of the shared word tokens, i.e. Average Jaccard Similarity (JS), Ranked-Biased Overlap (RBO), Average Pairwise Pointwise Mutual Information (PMI), and metrics that are based on the probability distributions, i.e. Average Log Odds Ratio (LOR), Kullback-Leibler Divergence (KL-DIV).

In the following, we will propose several metrics that are based on pre-trained word embeddings to overcome the limitations of the existing metrics.

B.2 Word Embedding-Based Similarity

To overcome the absence of semantics in the traditional similarity measures available in the state of the art, one can resort to the use of word embeddings to capture conceptual relationships between words. In the word embedding spaces, the vector representations of the words appearing in similar contexts tend to be close to each other (Mikolov et al., 2013). We can therefore exploit the nature of word embeddings and define new metrics to estimate how much two topic descriptors are similar.

B.2.1 *Word Embedding-based Centroid Similarity (WECS)*

The simplest strategy consists of computing the centroids of two topics t_i and t_j and then estimating their similarity. Let be $\overrightarrow{t_i}$ the vector centroid of the topic descriptor t_i computed as the average of word embeddings considering all the words belonging to the topic t_i.

The Word Embedding-based Centroid Similarity between two topics is estimated as $WECS(t_i, t_j) = sim(\overrightarrow{t_i}, \overrightarrow{t_j})$, where sim is a measure of similarity between vectors, i.e. cosine similarity.

B.2.2 *Word Embedding-based Pairwise Similarity (WEPS)*

An alternative to WECS consists of averaging the pairwise similarity between the embedding vectors of the words composing the topic descriptors. We define the similarity between two topics t_i and t_j as follows:

$$WEPS(t_i, t_j) = \frac{1}{N^2} \sum_{v \in t_i} \sum_{u \in t_j} sim(w_v, w_u) \tag{61}$$

where N represents the number of words of each topic, and w_v and w_u denote the word embeddings associated with words v and u respectively.

B.2.3 *Word embedding-based Weighted Sum Similarity (WESS)*

A simple way to combine the probability distributions and the word embeddings is to compute the sum of the word embeddings of the words in the vocabulary, where the sum is weighted by the probability of each term in the topic. Then, we compute the similarity between the resulting word embeddings.

More formally, let be $b_i = \sum_{v \in V} \phi_{iv} \cdot w_v$ the weighted sum of the word embeddings of the vocabulary for the topic i. Therefore, the WESS for the topic i and j is defined as $sim(b_i, b_j)$.

B.2.4 *Word Embedding-based Ranked-Biased Overlap (WERBO)*

We can extend the Ranked-Biased Overlap metric (Webber et al., 2010) and define a new metric of similarity that is top-weighted and makes use of word embeddings. Given the lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$, the words *cat* and *kitten* are similar, even though they are lexicographically different. It follows that their overlap at depth 2 should be higher than 1. We therefore generalize the concept of overlap to handle word embeddings instead of simple word tokens.

Algorithm 1 shows how to compute the generalized overlap between two topic descriptors t_i and t_j. First of all, we compute the

Algorithm 1 Calculate generalized overlap at depth h

Input: t_i, t_j topic descriptors composed of n words; h depth of the list, where $h \leqslant n$

```
1: for u := 1, ..., h do
2:     for v := 1, ..., h do
3:         sim[w_u^i, w_v^j] := similarity(w_u^i, w_v^j)
4:     end for
5: end for
6: overlap := 0
7: while sim is not empty do
8:     max_value := max(sim)
9:     w_u^i, w_v^j := get_indices(max_value)
10:    remove all entries of w_u^i and w_v^j from sim
11:    overlap := overlap + max_value
12: end whilereturn overlap
```

similarity between all the pairs of word embedding vectors w_u^i and w_v^j belonging to the two topics i and j (line 1-5). The associative array sim (line 3) is indexed by the tuple (w_u^i, w_v^j) and contains all the computed similarities. Subsequently (line 7-12), we process the associative array sim to get the words that are the most similar, to then update the overlap variable. In particular, the algorithm searches for the tuple (w_u^i, w_v^j) that has the highest similarity in sim (line 8), removes from sim all the entries containing w_u^i or w_v^j (line 9-10) and finally updates the overlap by adding the highest similarity value corresponding to the tuple (w_u^i, w_v^j) (line 12). For example, let us compute the generalized overlap at depth 3 of the word lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$. The result will be $sim(animal, animal) + sim(cat, kitten) + sim(animals, dog)$, because $(animal, animal)$ are identical vectors and should be summed first, then $(cat, kitten)$ are the second most similar vectors, and finally $(animals, dog)$ are the remaining vectors and should be summed at last.

In the proposed algorithm, $similarity(w_u^i, w_v^j)$ is the angular similarity between the vectors associated with the word embeddings related to the words u and v respectively[1]. Notice that this approach is based on a greedy strategy that estimates the overlapping by considering first the most similar embeddings of the words available in the top-h list. We will then refer to this approach as **WERBO-M**. Instead of computing the similarity between each word embedding, an alternative metric can compute the centroid of the embeddings at depth h. In this way, the overlap at depth h is just defined as

1 We use the angular similarity instead of the cosine because we require the overlap to range from 0 to 1.

$similarity(\overrightarrow{t_i}, \overrightarrow{t_j}) \cdot h$, where $\overrightarrow{t_i}$ and $\overrightarrow{t_j}$ are the centroids of the topics t_i and t_j respectively. We will refer to this metric as **WERBO-C**.

B.2.5 *Weighted Graph Modularity (WGM)*

We can rethink two topic descriptors in the form of a graph. Each word represents a node in the graph, while the edges denote the similarity between the words. Considering two topics composed of their own words (nodes), the intra-topic similarity connections should be higher than the extra-topic similarity connections with any other topic. We can express this idea by using the measure of modularity, which estimates the strength of division of a graph into modules (in our case, topics).

Let $G = (U, E)$ be a fully connected graph, where U is the words related to t_i and t_j and E are weighted edges denoting the similarity between pairs of word embeddings. In particular, an edge weight is defined as $A_{uv} = sim(w_v, w_u)$, where $(u, v) \in E$, $v, u \in U$ and $sim(\cdot, \cdot)$ is the angular similarity between two word embeddings. Given the graph G, originating from two topic descriptors t_i and t_j, the Weighted Graph Modularity (WGM) can be estimated as:

$$WGM(t_i, t_j) = \frac{1}{2m} \sum_{v,u \in U(G)} [A_{vu} - \frac{k_v k_u}{2m}] \mathbb{1}_{vu} \tag{62}$$

where k_v and k_u denote the degrees of the nodes v and u respectively, m is the sum of all of the edge weights in the graph, and $\mathbb{1}_{vu}$ is an indicator function defined as 1 if v and u are words belonging to the same topic, 0 otherwise. Modularity ranges from -1/2 (non-modular topics) to 1 (fully separated topics). Therefore, it should be considered as a dissimilarity score.

B.3 Experimental Investigation

B.3.1 *Experimental Setting*

Compared measures. Before proceeding with the description of the validation strategy and the performance measures adopted for a comparative evaluation, we summarize the investigated measures. In particular, in Table B.1 we provide details about all the metrics, reporting their main features:

- TD, which denotes if the metric considers the top-N words of the topic descriptors;
- PD, that reports if the metric considers the topic probability distribution;

- WE, which indicates if the metric overcomes the limitation of the discrete representation of words by using Word Embeddings;

- TW, that identify if the metric is top-weighted, i.e. the words at the top of the ranked list are more important than the words in the tail.

The measures' implementations are integrated into the topic modeling framework OCTIS, available at https://github.com/mind-lab/octis.

Similarity/Distance Measure	TD	PD	WE	TW
Jaccard Similarity (JS) (Tran et al., 2013)	✓			
Rank-biased Overlap (RBO) (Webber et al., 2010)	✓			✓
Pointwise Mutual Information (PMI) (Aletras and Stevenson, 2014)	✓			
Average Log Odds Ratio (LOR) (Chaney and Blei, 2012)		✓		
Kullback-Leibler Divergence (KL-DIV) (Sievert and Shirley, 2014)		✓		
Word embedding-based Centroid Similarity (WECS)	✓		✓	
Word Embedding Pairwise Similarity (WEPS)	✓		✓	
Word Embedding-based Weighted Sum Similarity (WESS)		✓	✓	
Word Embedding-based RBO - Match (WERBO-M)	✓		✓	✓
Word Embedding-based RBO - Centroid (WERBO-C)	✓		✓	✓
Weighted Graph Modularity (WGM)	✓		✓	

Table B.1: Summary of the characteristics of the metrics. The newly proposed metrics are reported in bold.

VALIDATION STRATEGY. To validate the proposed similarity measures, and compare them with the state-of-the-art ones, we selected the most widely adopted topic model to produce a set of topics to be evaluated. In particular, we trained Latent Dirichlet Allocation (LDA) (Blei et al., 2003a) on two benchmark datasets, i.e. BBC news (Greene and Cunningham, 2006) and 20 NewsGroups.[2], originating 50 different topics per dataset.[3] For the pre-processing, we removed the punctuation and the English stop-words[4], and we filtered out the less frequent words, obtaining a final vocabulary of 2000 terms.

Given the topics extracted by LDA, we disregarded those with a low value of topic coherence, measured by using Normalized Pointwise Mutual Information (NPMI) (Lau et al., 2014a) on the dataset itself as a reference corpus. Then we randomly sampled 100 pairs of topics (for each dataset) that have been evaluated by three annotators, by considering the top-10 words. In particular, the annotators have rated if two topics were related to each other or not, using a value

2 http://people.csail.mit.edu/jrennie/20Newsgroups/

3 We trained LDA with the default hyperparameters of the Gensim library.

4 We used the English stop-words list provided by MALLET: http://mallet.cs.umass.edu/

of 0 (not related topics) and 1 (similar topics). The final annotation of each pair of topics has been determined according to a majority voting strategy on the rates given by the three annotators.

For the metrics that are based on the topic descriptors, we considered the top-10 words of each topic. Regarding the metrics that are based on word embeddings, we used Gensim's[5] Word2Vec model to compute the embedding space on the corpus with the default hyperparameters. The co-occurrence probabilities for the estimation of PMI have been computed on the training dataset. For the metrics that represent dissimilarity scores, such as KL-DIV, the LOR and WGM metrics, we considered their inverse.

PERFORMANCE MEASURES. We evaluated the capabilities of all the topic similarity metrics, both the ones available in the state of the art and the proposed ones, by measuring Precision@k, Recall@k and F1-Measure@k.

In particular, Precision@k (P@k) is defined as the fraction of the number of retrieved topics among the top-k retrieved topics that are relevant and the number of retrieved topics among the top-k retrieved topics. Recall@k (R@k) is defined as the fraction of the number of retrieved topics among the top-k retrieved topics that are relevant and the total number of relevant topics. F1-Measure@k (F1@k) is defined the harmonic mean between P@k and R@k, i.e.

$$F1@k = 2(P@k \cdot R@k)/(P@k + R@k). \tag{63}$$

B.3.2 *Experimental Results*

Table B.2 shows the results for the BBC News dataset in terms of P@k, R@k and F1@k by varying k for 1 to 5. As a first remark, we can see that the metrics that are based on the shared word tokens only, i.e. the Jaccard Distance (JD) and Rank-biased Overlap (RBO), achieve the lowest performance. KL-DIV and LOR, which are based only on the topic-word probability distributions, outperform the baselines JD and RBO, but they are not able to outperform the proposed measures that consider the word embeddings similarities. The most competitive metric with respect to the proposed ones is the PMI, which obtains comparative results to the word-embedding metrics for $k = 2$. These results suggest that considering a richer representation of topical words helps in retrieving semantically similar topics to a given target topic. In particular, WERBO-M and WERBO-C reach the highest scores in most of the cases. This means that not only the meaning of the words are important when evaluating the similarity of two topics, but also the position of each word in the topic matters. In fact,

5 https://radimrehurek.com/gensim/

		State-of-the-art metrics					Proposed metrics					
	k	JD	RBO	PMI	LOR	KL-DIV	WESS	WEPS	WECS	WERBO-M	WERBO-C	WGM
P@K	1	0.818	0.864	0.955	0.846	0.909	0.909	0.955	**1.000**	**1.000**	**1.000**	0.818
	2	0.727	0.705	**0.864**	0.769	0.750	0.795	0.841	0.841	**0.864**	**0.864**	0.795
	3	0.652	0.667	0.803	0.667	0.652	0.742	0.788	0.773	**0.818**	0.788	0.773
	4	0.557	0.557	0.705	0.596	0.602	0.682	0.705	0.693	**0.716**	**0.716**	0.693
	5	0.482	0.491	0.573	0.492	0.536	0.573	**0.582**	**0.582**	**0.582**	**0.582**	0.573
	avg	0.647	0.657	0.706	0.674	0.690	0.740	0.774	0.778	**0.796**	0.790	0.730
R@K	1	0.348	0.364	0.417	0.423	0.402	0.409	0.417	**0.439**	**0.439**	**0.439**	0.379
	2	0.545	0.534	**0.663**	0.641	0.587	0.614	0.648	0.648	0.659	**0.663**	0.621
	3	0.697	0.712	0.871	0.776	0.716	0.803	0.856	0.833	**0.879**	0.845	0.833
	4	0.784	0.784	0.977	0.885	0.848	0.951	0.977	0.966	**0.989**	**0.989**	0.966
	5	0.867	0.879	0.989	0.910	0.932	0.985	**1.000**	**1.000**	**1.000**	**1.000**	0.989
	avg	0.648	0.655	0.783	0.727	0.697	0.752	0.780	0.777	**0.793**	0.787	0.758
F1@K	1	0.456	0.479	0.539	0.521	0.517	0.524	0.539	**0.570**	**0.570**	**0.570**	0.480
	2	0.589	0.574	**0.708**	0.651	0.617	0.650	0.689	0.689	0.705	**0.708**	0.656
	3	0.644	0.660	0.798	0.675	0.645	0.734	0.783	0.765	**0.809**	0.777	0.765
	4	0.627	0.627	0.786	0.677	0.673	0.762	0.786	0.775	**0.797**	**0.797**	0.775
	5	0.595	0.605	0.698	0.610	0.654	0.697	**0.709**	**0.709**	**0.709**	**0.709**	0.698
	avg	0.582	0.589	0.706	0.627	0.621	0.673	0.701	0.701	**0.718**	0.712	0.675

Table B.2: Precision@K, Recall@K and F1-Measure@k on the BBC News dataset.

WERBO-M and WERBO-C outperform the metrics WEPS and WECD that do not take into consideration the rank of the words.

Table B.3 reports the results on the 20NewsGroups dataset. Here, the obtained results are similar to the previous dataset. All the word embedding-based metrics outperform the state-of-the-art ones. In particular, WERBO-C outperforms the other metrics or obtain comparable results in most the cases. Even if WESS is the similarity metric that obtains the best performance on average, the results obtained by WERBO-C and WERBO-M are definitely comparable. Also on this dataset PMI seems to be the most competitive metric, however the word-embedding metrics metrics outperform it in most of the cases.

We report in Table B.4 two examples of topics evaluated by the considered similarity/distance measures. The first example reports two topics, that clearly represent two distinct themes, likely *religion* and *technology*. In this case, all the proposed metrics can capture the diversity of the two topics as well as the measure of the state of the art. On the other hand, the second example reports two related topics about *technology*. We can easily notice that while all the measures of the state of the art suggest that the two topics are completely different because of their low values (e.g. JS = 0.053 and KL-DIV = -4.415), the proposed metrics can capture their actual similarity.

		State-of-the-art metrics					Proposed metrics					
	k	JD	RBO	PMI	LOR	KL-DIV	WESS	WEPS	WECS	WERBO-M	WERBO-C	WGM
P@K	1	0.833	0.833	**1.000**	0.833	0.833	**1.000**	**1.000**	0.958	0.958	0.917	0.958
	2	0.646	0.667	0.813	0.792	0.792	**0.833**	0.813	**0.833**	0.813	**0.833**	**0.833**
	3	0.569	0.569	0.681	0.653	0.667	0.694	0.694	0.694	**0.708**	**0.708**	0.694
	4	0.458	0.458	0.583	0.563	0.583	0.583	0.583	0.583	**0.604**	**0.604**	0.583
	5	0.408	0.408	0.492	0.492	0.492	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**
	avg	0.583	0.587	0.714	0.666	0.673	**0.722**	0.718	0.714	0.717	0.713	0.714
R@K	1	0.424	0.424	**0.542**	0.375	0.396	**0.542**	**0.542**	0.500	0.500	0.459	0.500
	2	0.581	0.591	0.758	0.667	0.737	**0.779**	0.758	**0.779**	0.758	0.772	**0.779**
	3	0.705	0.701	0.869	0.793	0.848	0.890	0.890	0.890	**0.904**	**0.904**	0.890
	4	0.734	0.734	0.950	0.866	0.950	0.950	0.950	0.950	**0.974**	**0.974**	0.950
	5	0.807	0.807	0.974	0.946	0.974	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**
	avg	0.650	0.651	0.819	0.730	0.781	**0.830**	0.825	0.821	0.825	0.819	0.821
F1@K	1	0.522	0.522	**0.653**	0.487	0.501	**0.653**	**0.653**	0.612	0.612	0.570	0.612
	2	0.566	0.580	0.727	0.681	0.706	**0.748**	0.727	**0.748**	0.727	0.744	**0.748**
	3	0.587	0.585	0.709	0.670	0.692	0.725	0.725	0.725	**0.739**	**0.739**	0.725
	4	0.527	0.527	0.674	0.640	0.674	0.674	0.674	0.674	**0.696**	**0.696**	0.674
	5	0.510	0.510	0.610	0.607	0.610	**0.621**	**0.621**	**0.621**	**0.621**	**0.621**	**0.621**
	avg	0.542	0.545	0.675	0.617	0.637	**0.684**	0.680	0.676	0.679	0.674	0.676

Table B.3: Precision@K, Recall@K and F1-Measure@k on 20 NewsGroups.

Topic 1	Topic 2	Metrics	Topic 1	Topic 2	Metrics
god	ftp	JS=0	tiff	window	JS=0.053
christian	fax	RBO=0	gif	application	RBO=0.057
christianity	pub	PMI=-0.042	image	manager	PMI=0.327
religion	graphics	LOR=-3.204	format	display	LOR=-2.110
faith	computer	KL-DIV=-4.36416	jpeg	color	KL-DIV=-4.415
christ	software	WESS=-0.145	formats	widget	WESS=0.787
sin	version	WEPS=-0.0941	color	mouse	WEPS=0.402
people	mail	WECS=-0.183	images	screen	WECS=0.565
view	gov	WERBO-M=0.472	complex	button	WERBO-M=0.651
paul	mit	WERBO-C=0.120	resolution	user	WERBO-C=0.170
		WGM=-0.102			WGM=-0.015
Ground Truth = unrelated topics			Ground Truth = similar topics		

Table B.4: Qualitative comparison of the considered measures. Since KL-DIV, LOR and WGM represent dissimilarity scores, they are reported as their inverse.

C

ADDITIONAL RESULTS

C.1 COMPARATIVE ANALYSIS BETWEEN NEURAL TOPIC MODELS

In this Section we report additional details on the comparative analysis between neural topic models of Section 6.4.

C.1.1 *Best Hyperparameter Configurations*

We report all the best configurations of hyperparameters discovered by BO for all the models and for all the considered evaluation measures in Tables C.1, C.2, C.3, C.4, C.5 and C.6. This would allow a user to choose a promising hyperparameter configuration for the evaluation metric of their interest.

C.2 HYPERPARAMETER TRANSFER

In this Section we report additional details on the hyperparameter transfer experiments of Section 7.2.

C.2.1 *Disaggregated Results*

Figures C.1, C.2 and C.3 show reports the obtained value of the considered metric for the 5 best hyperparameter configurations that we transferred from a dataset (x-axis) to the target dataset ($\rightarrow$ dataset name).

		α **prior**	β **prior**	**Median**
20NG	F1*	1.332	1.146	0.472
	IRBO*	0.325	0.004	0.954
	KL-B*	0.006	3.054	2.299
	NPMI*	0.658	0.520	0.066
M10	F1*	0.627	1.870	0.469
	IRBO*	0.349	9.403	0.939
	KL-B*	$2 \cdot 10^{-4}$	9.614	2.343
	NPMI*	0.005	1.531	-0.083

Table C.1: Best configuration of hyperparameters discovered by BO for LDA for each evaluation measure.

		Activation	Dropout	Learn Priors	Learning Rate
20NG	F1*	sigmoid	0.0839	1	0.0097
	IRBO*	sigmoid	0.0839	1	0.0097
	KL-B*	sigmoid	0.9481	1	0.0039
	NPMI*	selu	0.0381	0	0.0208
M10	F1*	elu	0.0025	1	0.0611
	IRBO*	sigmoid	0.0839	1	0.0097
	KL-B*	rrelu	0.0198	1	0.0089
	NPMI*	softplus	0.1664	0	0.0006

		Momentum	Num Layers	Num Neurons	Optimizer
20NG	F1*	0.789	1	800	adam
	IRBO*	0.789	1	800	adam
	KL-B*	0.984	1	1000	sgd
	NPMI*	0.949	3	600	adam
M10	F1*	0.742	5	1000	adam
	IRBO*	0.789	1	800	adam
	KL-B*	0.512	5	100	adam
	NPMI*	0.374	1	400	sgd

Table C.2: Best configuration of hyperparameters discovered by BO for ProdLDA for each evaluation measure.

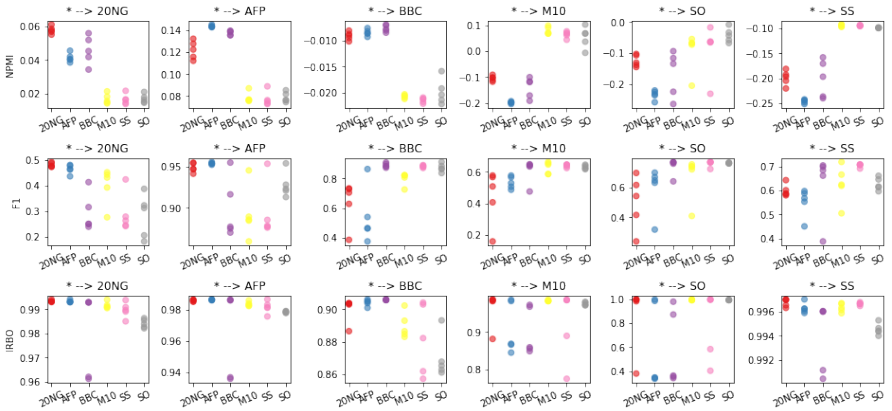


Figure C.1: Training for LDA: * → d denotes that we transfer hyperparameters from the dataset * to train LDA on the dataset d. The x-axis reports the different datasets from which a configuration is transferred.

		Activation	Dropout	Learn Priors	Learning Rate
20NG	F1*	sigmoid	0.084	0	0.0314
	IRBO*	sigmoid	0.062	1	0.0273
	KL-B*	elu	0.0003	0	0.0008
	NPMI*	sigmoid	0.130	0	0.0075
M10	F1*	sigmoid	0.061	0	0.0129
	IRBO*	leakyrelu	0.125	0	0.0019
	KL-B*	selu	0.0003	0	0.0186
	NPMI*	selu	0.087	1	0.0002

		Momentum	Num Layers	Num Neurons	Optimizer
20NG	F1* 0.575	1	1000	adam	0.339
	IRBO*	0.667	1	400	adam
	KL-B*	0.891	3	700	adam
	NPMI*	0.797	1	800	rmsprop
M10	F1*	0.756	1	800	rmsprop
	IRBO*	0.859	2	200	sgd
	KL-B*	0.269	2	600	adam
	NPMI*	0.754	1	100	sgd

Table C.3: Best configuration of hyperparameters discovered by BO for NeurLDA for each evaluation measure.

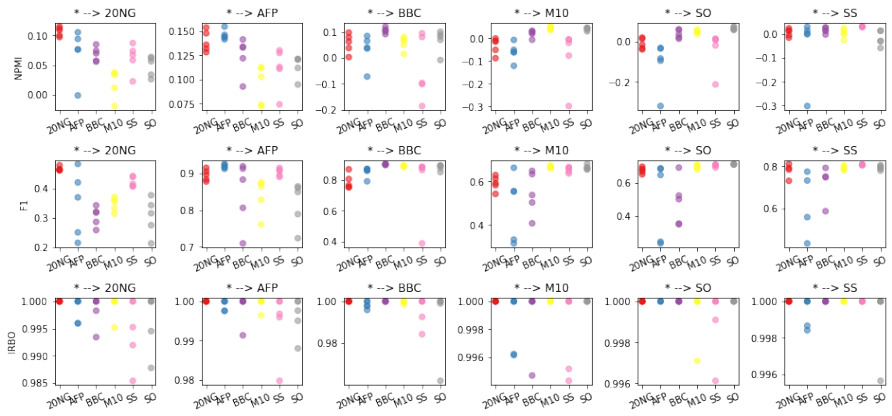


Figure C.2: Training for CTM: * → d denotes that we transfer hyperparameters from the dataset * to train CTM on the dataset d. The x-axis reports the different datasets from which a configuration is transferred.

		Activation	Dropout	Learn Priors	Learning Rate	Momentum
20NG	F1*	sigmoid	0.046	1	0.0018	0.751
	IRBO*	leakyrelu	0.145	0	0.0922	0.336
	KL-B*	elu	0.013	0	0.0950	0.725
	NPMI*	selu	0.064	0	0.0065	0.945
M10	F1*	sigmoid	0.190	1	0.0087	0.091
	IRBO*	sigmoid	0.084	1	0.0097	0.789
	KL-B*	selu	0.088	1	0.0135	0.964
	NPMI*	sigmoid	0.617	0	0.0010	

		Momentum	Num Layers	Num Neurons	Optimizer
20NG	F1*	0.751	1	700	adam
	IRBO*	0.336	1	800	adam
	KL-B*	0.725	5	300	rmsprop
	NPMI*	0.945	1	1000	rmsprop
M10	F1*	0.091	2	800	adam
	IRBO*	0.789	1	800	adam
	KL-B*	0.964	5	800	adam
	NPMI*	0.308	1	800	sgd

Table C.4: Best configuration of hyperparameters discovered by BO for CTM for each evaluation measure.

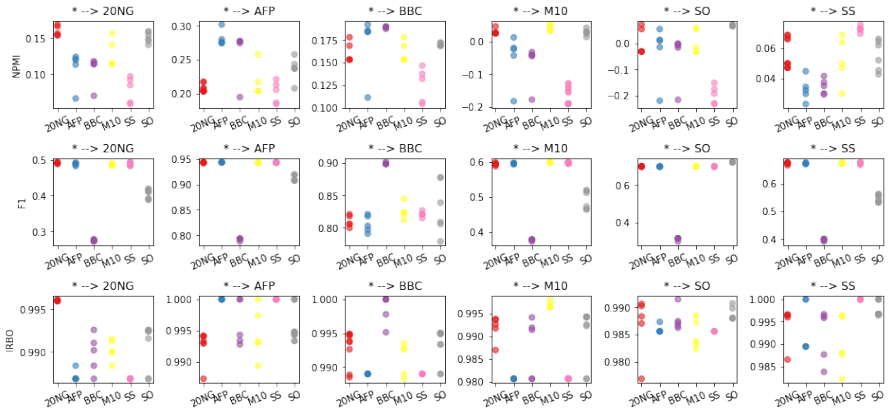


Figure C.3: Training for NMF: * → d denotes that we transfer hyperparameters from the dataset * to train NMF on the dataset d. The x-axis reports the different datasets from which a configuration is transferred.

		Activation	BOW norm	Dropout	Learning Rate
20NG	F1*	leakyrelu	1	0.315	0.006393
	IRBO*	sigmoid	0	0.919	0.000176
	KL-B*	leakyrelu	1	0.044	0.027539
	NPMI*	leakyrelu	1	0.009	0.004234
M10	F1*	rrelu	1	0.058	0.006062
	IRBO*	sigmoid	0	0.206	0.000003
	KL-B*	selu	0	0.602	0.003294
	NPMI*	relu	1	0.500	0.005000

		Optimizer	Rho size	Hidden size	Weight decay
20NG	F1*	adam	200	800	0.000005
	IRBO*	sgd	200	300	0.000004
	KL-B*	adagrad	300	300	0.000005
	NPMI*	adam	200	200	0.000005
M10	F1*	adam	100	600	0.000001
	IRBO*	adagrad	200	100	0.007168
	KL-B*	adam	300	1000	0.000155
	NPMI*	adam	300	300	0.000001

Table C.5: Best configuration of hyperparameters discovered by BO for ETM for each evaluation measure.

		Activation	BOW norm	Dropout	Learning Rate
20NG	F1*	elu	1	0.814	0.000008
	IRBO*	relu	0	0.918	0.000002
	KL-B*	selu	1	0.157	0.004597
	NPMI*	elu	0	0.121	0.000331
M10	F1*	softplus	0	0.182	0.000042
	IRBO*	selu	1	0.406	0.008958
	KL-B*	leakyrelu	1	0.051	0.013990
	NPMI*	relu	1	0.500	0.005000

		Optimizer	Hidden size	Weight decay
20NG	F1*	adam	700	0.000190
	IRBO*	adam	600	0.001485
	KL-B*	adam	1000	0.000076
	NPMI*	rmsprop	1000	0.000004
M10	F1*	adam	800	0.000001
	IRBO*	adam	1000	0.002974
	KL-B*	adam	300	0.000002
	NPMI*	adam	300	0.000001

Table C.6: Best configuration of hyperparameters discovered by BO for ETM-PWE for each evaluation measure.

C.2.2 *Best hyperparameters configurations*

Tables C.7, C.8 and C.9 report the 5 best hyperparameter configurations for LDA for F1, NPMI, and IRBO respectively. Analogous details are provided in Tables C.10, C.11 and C.12, C.13, C.14 and C.15 for CTM and NMF respectively.

C.2.3 *Computing Infrastructure*

We ran the experiments on a machine equipped with 4 T1390 GPU, CUDA v11.1, 512GB RAM, Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz.

Dataset	α prior	β prior	# Topics	IRBO	NPMI	F1
20NG	0.0001	10.0000	20	0.95	0.061	0.397
20NG	0.0008	6.3062	26	0.953	0.058	0.424
20NG	0.0121	0.0001	21	0.95	0.057	0.408
20NG	0.0023	1.2349	27	0.961	0.057	0.428
20NG	0.0002	0.0002	19	0.946	0.055	0.403
BBC	0.1007	10.0000	32	0.827	-0.007	0.73
BBC	0.0863	6.7863	48	0.85	-0.007	0.742
BBC	0.1892	1.0099	66	0.861	-0.008	0.746
BBC	0.0080	3.0815	38	0.836	-0.008	0.751
BBC	0.0002	0.0001	46	0.849	-0.009	0.627
AFP	0.0015	0.0001	46	0.976	0.145	0.942
AFP	0.0001	0.0001	49	0.976	0.144	0.94
AFP	0.0054	0.0008	44	0.976	0.143	0.941
AFP	0.0004	0.0010	50	0.977	0.143	0.942
AFP	0.1969	0.0002	46	0.977	0.143	0.944
SS	0.0001	4.8535	147	0.992	-0.093	0.603
SS	0.0003	9.1500	149	0.991	-0.094	0.538
SS	0.0119	10.0000	150	0.991	-0.094	0.567
SS	0.0001	10.0000	150	0.991	-0.095	0.518
SS	0.0001	10.0000	150	0.991	-0.096	0.506
M10	0.0030	10.0000	150	0.974	0.101	0.142
M10	0.0013	10.0000	150	0.972	0.098	0.106
M10	0.0024	10.0000	150	0.973	0.098	0.144
M10	0.0011	10.0000	150	0.974	0.097	0.141
M10	0.0014	10.0000	150	0.972	0.097	0.141
SO	0.0007	10.0000	133	0.97	-0.008	0.093
SO	0.0019	10.0000	138	0.955	-0.012	0.094
SO	0.0004	10.0000	138	0.955	-0.024	0.093
SO	0.0004	10.0000	142	0.94	-0.024	0.093
SO	0.0002	10.0000	142	0.941	-0.025	0.093

Table C.7: Best 5 hyperparameter configurations for LDA on each dataset for NPMI.

Dataset	α prior	β prior	# Topics	IRBO	NPMI	F1
20NG	0.0001	10.0000	20	0.95	0.061	0.397
20NG	0.0008	6.3062	26	0.953	0.058	0.424
20NG	0.0121	0.0001	21	0.95	0.057	0.408
20NG	0.0023	1.2349	27	0.961	0.057	0.428
20NG	0.0002	0.0002	19	0.946	0.055	0.403
BBC	0.1007	10.0000	32	0.827	-0.007	0.73
BBC	0.0863	6.7863	48	0.85	-0.007	0.742
BBC	0.1892	1.0099	66	0.861	-0.008	0.746
BBC	0.0080	3.0815	38	0.836	-0.008	0.751
BBC	0.0002	0.0001	46	0.849	-0.009	0.627
AFP	0.0015	0.0001	46	0.976	0.145	0.942
AFP	0.0001	0.0001	49	0.976	0.144	0.94
AFP	0.0054	0.0008	44	0.976	0.143	0.941
AFP	0.0004	0.0010	50	0.977	0.143	0.942
AFP	0.1969	0.0002	46	0.977	0.143	0.944
SS	0.0001	4.8535	147	0.992	-0.093	0.603
SS	0.0003	9.1500	149	0.991	-0.094	0.538
SS	0.0119	10.0000	150	0.991	-0.094	0.567
SS	0.0001	10.0000	150	0.991	-0.095	0.518
SS	0.0001	10.0000	150	0.991	-0.096	0.506
M10	0.0030	10.0000	150	0.974	0.101	0.142
M10	0.0013	10.0000	150	0.972	0.098	0.106
M10	0.0024	10.0000	150	0.973	0.098	0.144
M10	0.0011	10.0000	150	0.974	0.097	0.141
M10	0.0014	10.0000	150	0.972	0.097	0.141
SO	0.0007	10.0000	133	0.97	-0.008	0.093
SO	0.0019	10.0000	138	0.955	-0.012	0.094
SO	0.0004	10.0000	138	0.955	-0.024	0.093
SO	0.0004	10.0000	142	0.94	-0.024	0.093
SO	0.0002	10.0000	142	0.941	-0.025	0.093

Table C.8: Best 5 hyperparameter configurations for LDA on each dataset for F1.

Dataset	α prior	β prior	# Topics	IRBO	NPMI	F1
20NG	0.1483	0.0083	150	0.994	-0.010	0.473
20NG	0.0446	0.0001	150	0.993	-0.005	0.455
20NG	0.0947	0.0002	150	0.993	0.005	0.399
20NG	0.0048	0.2006	150	0.993	-0.003	0.434
20NG	0.1961	0.0001	111	0.993	0.010	0.365
BBC	0.0001	0.0001	150	0.906	-0.021	0.357
BBC	0.0447	10.0000	150	0.906	-0.020	0.691
BBC	0.0001	0.0001	150	0.906	-0.021	0.348
BBC	0.0004	10.0000	150	0.906	-0.020	0.697
BBC	0.0003	0.0071	150	0.906	-0.020	0.325
AFP	0.1668	0.0002	150	0.987	0.103	0.953
AFP	0.0452	0.0001	148	0.987	0.107	0.948
AFP	0.0017	0.0001	149	0.986	0.107	0.943
AFP	0.0006	0.0015	150	0.986	0.107	0.943
AFP	0.0001	0.0001	150	0.986	0.107	0.943
SS	0.0990	0.4641	98	0.997	-0.256	0.557
SS	0.1242	0.0014	150	0.997	-0.273	0.577
SS	0.2860	0.0085	69	0.997	-0.265	0.579
SS	0.3039	0.9328	83	0.997	-0.220	0.611
SS	0.0052	0.3995	149	0.996	-0.236	0.553
M10	0.1483	0.0001	150	0.987	-0.236	0.291
M10	0.1496	0.0187	103	0.985	-0.252	0.556
M10	0.0524	0.2615	102	0.985	-0.239	0.429
M10	0.0130	0.0011	105	0.984	-0.209	0.500
M10	0.0249	0.0001	117	0.984	-0.207	0.510
SO	0.1657	0.0001	64	0.996	-0.302	0.585
SO	0.0863	0.5342	81	0.995	-0.297	0.605
SO	0.0808	0.0037	92	0.993	-0.304	0.607
SO	0.0135	0.4842	68	0.993	-0.264	0.503
SO	0.0223	0.5298	62	0.993	-0.262	0.504

Table C.9: Best 5 hyperparameter configurations for LDA on each dataset for IRBO.

Dataset	Activation	Dropout	Learn Priors	Learning Rate	Momentum	# Layers	# Topics	# Neurons	Optimizer	IRBO	NPMI	F1
20NG	rrelu	0.000	0	0.0001	0.114	1	143	1000	rmsprop	0.989	0.024	0.480
20NG	elu	0.113	1	0.0020	0.489	2	53	600	adam	0.994	0.094	0.467
20NG	elu	0.061	1	0.0001	0.387	2	118	100	rmsprop	0.985	0.044	0.464
20NG	leakyrelu	0.326	1	0.1000	0.808	1	63	400	rmsprop	0.995	0.082	0.464
20NG	rrelu	0.022	0	0.1000	0.200	1	64	200	adam	0.996	0.097	0.462
AFP	leakyrelu	0.274	1	0.0626	0.116	4	148	1000	rmsprop	0.989	0.104	0.926
AFP	rrelu	0.322	1	0.0015	0.900	1	150	1000	rmsprop	0.993	0.101	0.923
AFP	rrelu	0.354	0	0.0440	0.900	3	126	600	rmsprop	0.988	0.113	0.919
AFP	rrelu	0.115	1	0.0207	0.018	4	138	300	adam	0.991	0.112	0.917
AFP	softplus	0.000	0	0.1000	0.755	2	61	900	adam	0.995	0.145	0.913
BBC	rrelu	0.312	0	0.0149	0.420	3	139	900	adam	0.988	0.000	0.901
BBC	selu	0.327	0	0.0025	0.720	3	150	600	adam	0.978	0.062	0.901
BBC	selu	0.000	1	0.0055	0.061	1	9	800	rmsprop	1.000	0.066	0.899
BBC	elu	0.606	0	0.0001	0.482	2	78	300	sgd	0.948	0.075	0.899
BBC	rrelu	0.220	1	0.0013	0.900	1	6	900	rmsprop	1.000	0.004	0.894
M10	elu	0.438	0	0.0012	0.012	1	16	1000	sgd	0.985	0.042	0.674
M10	selu	0.645	0	0.0006	0.748	1	15	800	sgd	0.973	0.048	0.670
M10	elu	0.549	0	0.0025	0.012	1	24	300	rmsprop	0.980	0.026	0.665
M10	elu	0.708	1	0.0066	0.363	1	37	300	adam	0.971	0.023	0.664
M10	elu	0.640	1	0.0003	0.304	1	37	100	sgd	0.964	0.051	0.661
SO	elu	0.367	1	0.0005	0.558	3	22	600	adam	0.990	0.042	0.732
SO	selu	0.126	1	0.0004	0.377	1	44	700	adam	0.987	-0.019	0.721
SO	elu	0.577	1	0.1000	0.134	3	26	400	adadelta	0.972	0.034	0.717
SO	sigmoid	0.727	1	0.0009	0.716	1	23	600	adam	0.974	0.050	0.715
SO	elu	0.000	0	0.0001	0.840	2	19	600	rmsprop	0.996	0.043	0.714
SS	elu	0.452	1	0.0027	0.285	2	44	300	adam	0.991	0.011	0.809
SS	selu	0.777	1	0.0005	0.590	1	127	400	adam	0.977	0.023	0.807
SS	leakyrelu	0.034	1	0.0721	0.054	2	26	800	rmsprop	1.000	-0.009	0.806
SS	selu	0.489	1	0.0057	0.084	2	59	800	rmsprop	0.992	-0.024	0.805
SS	rrelu	0.562	1	0.0014	0.794	1	71	300	rmsprop	0.990	-0.012	0.804

Table C.10: Best 5 hyperparameter configurations for CTM on each dataset for F1.

Dataset	Activation	Dropout	Learn Priors	Learning Rate	Momentum	# Layers	# Topics	# Neurons	Optimizer	IRBO	NPMI	F1
20NG	elu	0.000	0	0.1000	0.859	1	27	1000	rmsprop	0.997	0.115	0.461
20NG	elu	0.127	0	0.0016	0.009	1	30	900	rmsprop	0.996	0.112	0.458
20NG	leakyrelu	0.000	1	0.0052	0.650	2	44	600	rmsprop	0.995	0.109	0.456
20NG	elu	0.000	1	0.0712	0.097	3	49	1000	adam	0.995	0.100	0.424
20NG	leakyrelu	0.356	0	0.0028	0.744	1	22	1000	rmsprop	0.995	0.097	0.409
BBC	softplus	0.000	1	0.0018	0.306	1	18	400	adam	0.994	0.120	0.876
BBC	softplus	0.272	0	0.0002	0.066	1	16	200	sgd	0.977	0.109	0.878
BBC	relu	0.003	1	0.0002	0.067	1	37	700	sgd	0.977	0.106	0.797
BBC	elu	0.000	0	0.0001	0.500	3	10	300	sgd	0.984	0.101	0.890
BBC	elu	0.130	0	0.0001	0.453	4	26	200	sgd	0.972	0.091	0.890
AFP	rrelu	0.000	0	0.1000	0.105	4	129	900	adagrad	0.989	0.155	0.887
AFP	rrelu	0.000	0	0.0495	0.089	3	41	100	rmsprop	0.995	0.146	0.898
AFP	softplus	0.000	0	0.1000	0.755	2	61	900	adam	0.995	0.145	0.913
AFP	relu	0.000	0	0.0075	0.701	1	97	1000	rmsprop	0.994	0.144	0.896
AFP	relu	0.000	1	0.0875	0.856	4	58	600	adam	0.995	0.142	0.911
SS	selu	0.802	1	0.0001	0.781	3	38	100	rmsprop	0.952	0.031	0.757
SS	relu	0.296	0	0.0001	0.640	1	42	100	rmsprop	0.981	0.031	0.773
SS	selu	0.360	0	0.0002	0.688	1	42	700	rmsprop	0.990	0.031	0.801
SS	selu	0.359	0	0.0001	0.643	1	34	800	rmsprop	0.991	0.030	0.802
SS	selu	0.824	1	0.0002	0.469	1	105	400	rmsprop	0.969	0.029	0.802
M10	elu	0.640	1	0.0003	0.304	1	37	100	sgd	0.964	0.051	0.661
M10	selu	0.645	0	0.0006	0.748	1	15	800	sgd	0.973	0.048	0.670
M10	elu	0.438	0	0.0012	0.012	1	16	1000	sgd	0.985	0.042	0.674
M10	leakyrelu	0.393	0	0.0005	0.111	2	22	700	sgd	0.972	0.038	0.648
M10	softplus	0.300	1	0.0006	0.694	2	30	800	sgd	0.974	0.036	0.657
SO	sigmoid	0.013	1	0.0016	0.442	1	18	100	sgd	0.991	0.073	0.701
SO	selu	0.000	1	0.0023	0.130	2	18	300	sgd	0.992	0.070	0.712
SO	relu	0.000	1	0.0005	0.870	2	16	1000	sgd	0.993	0.062	0.679
SO	rrelu	0.041	0	0.0050	0.305	2	19	200	sgd	0.990	0.060	0.690
SO	leakyrelu	0.482	1	0.0004	0.154	1	17	800	sgd	0.987	0.058	0.698

Table C.11: Best 5 hyperparameter configurations for CTM on each dataset for NPMI.

Dataset	Activation	Dropout	Learn Priors	Learning Rate	Momentum	# Layers	# Topics	# Neurons	Optimizer	IRBO	NPMI	F1
20NG	rrelu	0.0238	0	0.0078	0.741	3	5	400	rmsprop	1.000	0.021	0.215
20NG	softplus	0.4102	1	0.0580	0.270	2	5	600	rmsprop	1.000	0.013	0.203
20NG	rrelu	0.0000	1	0.0874	0.389	1	5	400	adagrad	1.000	0.004	0.235
20NG	rrelu	0.0402	0	0.1000	0.552	4	5	100	adagrad	1.000	-0.006	0.197
20NG	leakyrelu	0.0000	1	0.0633	0.385	1	5	500	adam	1.000	0.016	0.242
BBC	rrelu	0.0225	1	0.0676	0.864	4	5	900	adam	1.000	-0.039	0.843
BBC	selu	0.0000	1	0.0055	0.061	1	9	800	rmsprop	1.000	0.066	0.899
BBC	rrelu	0.2199	1	0.0013	0.900	1	6	900	rmsprop	1.000	0.004	0.894
BBC	relu	0.0000	0	0.0005	0.038	1	5	700	adadelta	1.000	-0.433	0.342
BBC	rrelu	0.0177	1	0.0792	0.819	1	5	900	adagrad	1.000	-0.032	0.894
AFP	selu	0.6558	1	0.0060	0.540	1	5	600	rmsprop	1.000	0.051	0.657
AFP	leakyrelu	0.0000	1	0.0712	0.439	3	5	800	adagrad	1.000	0.033	0.667
AFP	rrelu	0.0000	1	0.1000	0.763	4	5	200	adagrad	1.000	0.025	0.665
AFP	leakyrelu	0.0000	0	0.0009	0.532	1	5	400	rmsprop	1.000	0.064	0.666
AFP	rrelu	0.0000	1	0.0712	0.807	2	8	500	adagrad	1.000	0.083	0.781
SS	rrelu	0.0000	1	0.0578	0.455	4	17	700	rmsprop	1.000	-0.030	0.794
SS	selu	0.0000	1	0.0013	0.900	5	5	800	adam	1.000	-0.105	0.549
SS	selu	0.0000	1	0.0006	0.729	1	7	200	adam	1.000	0.005	0.694
SS	relu	0.1055	1	0.0088	0.836	4	5	300	adam	1.000	-0.155	0.530
SS	relu	0.8341	0	0.0010	0.900	2	5	1000	adam	1.000	-0.153	0.523
M10	rrelu	0.0000	0	0.0086	0.791	1	5	700	adam	1.000	-0.083	0.496
M10	leakyrelu	0.0329	0	0.0059	0.214	2	5	700	rmsprop	1.000	-0.101	0.478
M10	selu	0.0000	1	0.0196	0.088	1	5	300	adam	1.000	-0.116	0.490
M10	relu	0.0000	1	0.0489	0.157	1	5	400	adam	1.000	-0.134	0.492
M10	elu	0.6812	1	0.1000	0.013	1	6	300	adam	1.000	-0.073	0.520
SO	softplus	0.3098	1	0.0001	0.527	3	5	600	rmsprop	1.000	-0.135	0.304
SO	rrelu	0.4490	1	0.0167	0.602	1	5	300	rmsprop	1.000	-0.162	0.305
SO	selu	0.5117	0	0.0233	0.752	1	5	400	rmsprop	1.000	-0.147	0.310
SO	elu	0.1316	1	0.0006	0.405	2	5	900	sgd	1.000	-0.052	0.290
SO	softplus	0.2041	1	0.0002	0.312	3	5	1000	rmsprop	1.000	-0.116	0.308

Table C.12: Best 5 hyperparameter configurations for CTM on each dataset for IRBO.

Dataset	Reg. factor	L1/L2	Initialization	Regularization	# Topics	IRBO	NPMI	F1
20NG	0.000	1.000	random	H matrix	150	0.993	0.060	0.494
20NG	0.109	0.110	random	H matrix	150	0.992	0.059	0.492
20NG	0.000	1.000	nndsvda	H matrix	150	0.992	0.061	0.491
20NG	0.500	0.000	nndsvda	V matrix	150	0.992	0.061	0.489
20NG	0.001	0.537	random	H matrix	140	0.992	0.064	0.489
BBC	0.336	0.000	nndsvdar	both	5	0.989	0.153	0.901
BBC	0.068	0.156	nndsvd	V matrix	5	0.989	0.153	0.899
BBC	0.000	0.738	nndsvda	both	5	0.989	0.153	0.899
BBC	0.000	0.000	nndsvda	both	5	0.989	0.153	0.899
BBC	0.487	0.000	random	V matrix	5	0.989	0.153	0.899
AFP	0.131	0.000	random	both	150	0.994	0.188	0.944
AFP	0.139	0.280	random	H matrix	147	0.994	0.186	0.944
AFP	0.050	0.991	random	H matrix	150	0.994	0.184	0.943
AFP	0.314	0.000	nndsvdar	H matrix	150	0.995	0.186	0.943
AFP	0.445	0.066	nndsvdar	H matrix	148	0.995	0.185	0.943
SS	0.500	0.507	nndsvda	V matrix	150	0.997	0.022	0.676
SS	0.000	0.174	nndsvd	both	146	0.997	0.019	0.674
SS	0.000	0.021	nndsvdar	H matrix	150	0.997	0.017	0.674
SS	0.015	0.000	nndsvdar	both	150	0.997	0.018	0.674
SS	0.219	0.000	nndsvda	V matrix	150	0.997	0.018	0.673
M10	0.275	0.000	nndsvda	H matrix	150	0.994	-0.191	0.599
M10	0.000	0.408	nndsvdar	both	150	0.994	-0.192	0.598
M10	0.477	0.000	nndsvdar	V matrix	150	0.994	-0.191	0.596
M10	0.025	0.534	nndsvd	V matrix	150	0.994	-0.192	0.596
M10	0.000	0.696	nndsvdar	V matrix	146	0.994	-0.189	0.596
SO	0.390	0.570	nndsvda	V matrix	21	0.977	0.034	0.722
SO	0.428	0.563	nndsvdar	H matrix	46	0.986	-0.075	0.721
SO	0.130	0.739	nndsvd	H matrix	43	0.986	-0.067	0.721
SO	0.140	0.087	random	V matrix	44	0.986	-0.070	0.721
SO	0.395	0.730	nndsvdar	V matrix	25	0.981	-0.004	0.720

Table C.13: Best 5 hyperparameter configurations for NMF on each dataset for F1.

Dataset	Reg. factor	L1/L2	Initialization	Regularization	# Topics	IRBO	NPMI	F1
20NG	0.500	0.510	nndsvdar	both	5	0.983	0.169	0.150
20NG	0.423	0.555	nndsvdar	both	5	0.984	0.166	0.156
20NG	0.409	0.303	nndsvda	V matrix	9	0.988	0.156	0.325
20NG	0.312	0.350	nndsvda	both	5	0.980	0.154	0.208
20NG	0.026	0.395	nndsvda	H matrix	10	0.987	0.154	0.339
BBC	0.500	0.788	nndsvda	H matrix	28	0.992	0.190	0.833
BBC	0.429	0.344	nndsvd	both	125	0.959	0.190	0.487
BBC	0.234	0.264	nndsvd	V matrix	26	0.992	0.189	0.818
BBC	0.486	0.000	nndsvd	H matrix	26	0.992	0.189	0.830
BBC	0.465	0.433	nndsvda	H matrix	27	0.992	0.188	0.821
AFP	0.415	0.788	nndsvdar	both	133	0.993	0.302	0.806
AFP	0.426	0.804	nndsvdar	H matrix	20	0.994	0.280	0.900
AFP	0.114	0.078	random	V matrix	24	0.994	0.276	0.906
AFP	0.000	0.705	nndsvdar	both	28	0.994	0.275	0.904
AFP	0.000	0.000	random	V matrix	24	0.994	0.274	0.907
SS	0.379	0.283	nndsvd	both	146	0.994	0.076	0.512
SS	0.179	0.708	nndsvdar	both	99	0.993	0.073	0.543
SS	0.293	0.255	nndsvdar	both	87	0.995	0.073	0.570
SS	0.205	0.477	nndsvd	both	150	0.994	0.072	0.548
SS	0.248	0.319	nndsvdar	both	75	0.995	0.070	0.563
M10	0.000	0.025	random	V matrix	5	0.962	0.051	0.362
M10	0.037	1.000	nndsvdar	V matrix	9	0.994	0.049	0.430
M10	0.120	0.854	nndsvd	H matrix	15	0.989	0.039	0.468
M10	0.031	0.605	nndsvd	V matrix	5	0.981	0.032	0.368
M10	0.080	1.000	nndsvdar	H matrix	5	0.981	0.032	0.369
SO	0.443	0.741	nndsvdar	V matrix	12	0.974	0.071	0.578
SO	0.090	0.673	nndsvdar	both	10	0.975	0.070	0.484
SO	0.182	0.720	nndsvd	V matrix	14	0.975	0.068	0.604
SO	0.440	0.612	nndsvdar	H matrix	13	0.974	0.066	0.587
SO	0.018	0.378	nndsvdar	V matrix	15	0.973	0.064	0.628

Table C.14: Best 5 hyperparameter configurations for NMF on each dataset for NPMI.

Dataset	Reg. factor	L1/L2	Initialization	Regularization	# Topics	IRBO	NPMI	F1
20NG	0.250	1.000	random	V matrix	11	0.996	-0.207	0.056
20NG	0.443	0.980	random	V matrix	87	0.996	-0.195	0.056
20NG	0.479	0.607	random	V matrix	124	0.996	-0.193	0.056
20NG	0.204	0.795	random	V matrix	131	0.996	-0.195	0.056
20NG	0.500	0.919	random	V matrix	67	0.996	-0.196	0.056
BBC	0.390	0.786	nndsvd	both	68	1.000	0.077	0.301
BBC	0.281	0.998	random	V matrix	5	1.000	-0.431	0.227
BBC	0.500	1.000	random	both	5	1.000	-0.431	0.227
BBC	0.500	0.478	random	V matrix	57	0.998	-0.416	0.227
BBC	0.142	0.447	random	H matrix	150	0.995	0.101	0.803
AFP	0.335	0.000	random	H matrix	5	1.000	0.203	0.793
AFP	0.372	1.000	nndsvd	V matrix	5	1.000	0.203	0.788
AFP	0.366	0.883	nndsvda	V matrix	5	1.000	0.203	0.788
AFP	0.282	1.000	random	H matrix	5	1.000	0.197	0.791
AFP	0.011	0.000	random	H matrix	5	1.000	0.203	0.793
SS	0.072	0.018	nndsvda	V matrix	5	1.000	0.050	0.400
SS	0.000	0.742	nndsvdar	V matrix	5	1.000	0.050	0.401
SS	0.000	0.849	nndsvda	H matrix	5	1.000	0.050	0.395
SS	0.199	1.000	nndsvdar	V matrix	5	1.000	0.047	0.396
SS	0.323	1.000	nndsvd	V matrix	5	1.000	0.047	0.392
M10	0.500	0.182	random	V matrix	5	0.998	-0.643	0.135
M10	0.215	0.804	random	V matrix	6	0.997	-0.633	0.135
M10	0.430	0.736	random	V matrix	7	0.996	-0.624	0.135
M10	0.092	0.264	random	V matrix	86	0.996	-0.514	0.135
M10	0.140	0.789	random	V matrix	70	0.996	-0.516	0.135
SO	0.278	0.116	nndsvdar	V matrix	150	0.992	-0.231	0.696
SO	0.442	0.000	nndsvd	H matrix	150	0.992	-0.231	0.699
SO	0.500	0.000	random	V matrix	150	0.992	-0.233	0.697
SO	0.429	1.000	nndsvda	H matrix	5	0.991	-0.013	0.312
SO	0.183	0.000	nndsvda	both	150	0.991	-0.227	0.698

Table C.15: Best 5 hyperparameter configurations for NMF on each dataset for IRBO.