Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Georgios BALIKAS

Thèse dirigée par **Massih-Reza AMINI**, Professeur, Université Grenoble Alpes

préparée au sein du **Laboratoire d'Informatique de Grenoble** dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Explorer et apprendre à partir de collections de textes multilingues à l'aide des modèles probabilistes latents et des réseaux profonds

Mining and Learning from Multilingual Text Collections using Topic Models and Word Embeddings

Thèse soutenue publiquement le **20 octobre 2017**,
devant le jury composé de :

Monsieur MASSIH-REZA AMINI
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Directeur de these
Monsieur CYRIL GOUTTE
CHERCHEUR, CONSEIL NATIONAL DE RECHERCHES CANADA, Rapporteur
Monsieur GAËL DIAS
PROFESSEUR, UNIVERSITE DE CAEN NORMANDIE, Rapporteur
Monsieur LAURENT BESACIER
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Examinateur, Président du jury
Monsieur GUILLAUME VERNAT
DIRECTEUR GENERAL, SOCIETE COFFREO, Examinateur
Monsieur PATRICK GALLINARI
PROFESSEUR, UNIVERSITE PIERRE ET MARIE CURIE, Examinateur

Abstract

Text is one of the most pervasive and persistent sources of information. Content analysis of text in its broad sense refers to methods for studying and retrieving information from documents. Nowadays, with the ever increasing amount of text becoming available online in several languages and writing styles, content analysis of text is of tremendous importance as it enables a variety of applications. To this end, unsupervised representation learning methods like topic models and word embeddings constitute prominent tools. The goal of this thesis is to study and address challenging problems in this area, focusing on both the design of novel text mining algorithms and tools, as well as on studying how these tools can be applied to text collections written in a single or several languages.

In the first part of the thesis we focus on topic models and more precisely on how to incorporate prior information of text structure to them. Topic models are built on the premise of bag-of-words, and therefore words are exchangeable. While this assumption benefits the calculations of the conditional probabilities it results in loss of information. To overcome this limitation we propose two mechanisms that extend topic models by integrating knowledge of text structure to them. To this end, we begin by assuming that the documents are partitioned in thematically coherent text segments. Then, the first mechanism assigns the same topic to the words of a segment. The second, capitalizes on the properties of copulas, a tool mainly used in the fields of economics and risk management that is used to model the joint probability distributions of random variables while having access only to their marginals. Through the use of copulas we propose flexible topic models that can model different degrees of dependence between the topics of a segment.

The second part of the thesis explores bilingual topic models for comparable corpora with explicit document alignments. Typically, a document collection for such models is in the form of comparable document

pairs. The documents of a pair are written in different languages and are thematically similar. Unless translations, the documents of a pair are similar to some extent only. Meanwhile, representative topic models assume that the documents have identical topic distributions, which is a strong and limiting assumption. To overcome the limitations of this assumption we propose novel bilingual topic models that incorporate the notion of cross-lingual similarity of the documents that constitute the pairs in their generative and inference processes. Calculating this cross-lingual document similarity is a task on itself, which we propose to address using cross-lingual word embeddings.

The last part of the thesis concerns the use of word embeddings and neural networks for three text mining applications. First, we discuss polylingual document classification where we argue that translations of a document can be used to enrich its representation. Using an auto-encoder to obtain these robust document representations we demonstrate improvements in the task of multi-class document classification. Second, we explore multi-task sentiment classification of tweets arguing that jointly training classification systems on correlated tasks can improve the obtained performance. To this end we show how one can achieve state-of-the-art performance on a sentiment classification task using recurrent neural networks. The third application we explore is cross-lingual information retrieval. Given a document written in one language, the task consists in retrieving the most similar documents from a pool of documents written in another language. In this line of research, we demonstrate how adapting the transportation problem for estimating document distances one can achieve important improvements.

List of Publications

The following publications are included in parts or in an extended version in this thesis:

- Georgios Balikas, Ioannis Partalas, Eric Gaussier, Rohit Babbar, and Massih-Reza Amini. Efficient model selection for regularized classification by exploiting unlabeled data. *In International Symposium on Intelligent Data Analysis*, IDA 2015, pages 25–36. Springer, 2015.

- Georgios Balikas and Massih-Reza Amini. Multi-label, multi-class classification using polylingual embeddings. *In Advances in Information Retrieval 137- 38th European Conference on IR Research*, ECIR 2016, Padua, Italy, March 20-23, 2016. pages 723–728. Springer, 2016.

- Georgios Balikas and Massih-Reza Amini. Twise at semeval-2016 task 4: Twitter sentiment classification. *In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016*, San Diego, CA, USA, June 16-17, 2016, pages 85–91, 2016.

- Georgios Balikas, Massih-Reza Amini, and Marianne Clausel. On a topic model for sentences. *In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, SIGIR 2016, Pisa, Italy, July 17-21, 2016, pages 921–924, 2016.

- Georgios Balikas, Hesam Amoualian, Marianne Clausel, Éric Gaussier, and Massih-Reza Amini. Modeling topic dependencies in semantically coher- ent text spans with copulas. *In Proceedings of the 26th International Conference on Computational Linguistics*, COLING 2016, December 11-16, 2016, Osaka, Japan, pages 1767–1776, 2016.

- Georgios Balikas, Simon Moura, and Massih-Reza Amini. Multitask learning for fine-grained twitter sentiment analysis. *In Proceedings of the 40th*

International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2017, Tokyo, Japan, August 7-11, 2017, 2017.

In addition to the topics studied in this manuscript which are mentioned above, during my thesis I worked on several other problems leading to the following publications:

- Hesam Amoualian, Wei Lu, Eric Gaussier, Georgios Balikas, Massih R Amini, Marianne Clausel: Topical Coherence in LDA-based Models through Induced Segmentation, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017.
- Georgios Balikas: TwiSe at SemEval-2017 Task 4: Five-point Twitter Sentiment Classification and Quantification , In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, 2017.
- Georgios Balikas and Massih-Reza Amini: An empirical study on large scale text classification with skip-gram embeddings, Neu-IR @ SIGIR 2016, Pisa, Italy, 2016.
- Anil Goyal, Georgios Balikas, Prantapa Goswami, Massih-Reza Amini and Eric Gaussier: Transfer Learning for IR Model Parameter Tuning, Conférence Francophone sur l'Apprentissage Automatique (CAp), Lille, 2015.
- Georgios Balikas and Massih-Reza Amini: Learning language-independent sentence representations for multi-lingual, multi-document summarization, Conférence Francophone sur l'Apprentissage Automatique (CAp), Lille, 2015.

Acknowledgements

This thesis is the result of efforts of a ten-year university student period. During this endeavor is was supported by several people, that I would like to deeply thank.

First, I would like to express my sincere gratitude to my advisor Prof. Massih-Reza Amini. I am grateful for his support and advice. Our discussions and brainstorming during all these years have deeply influenced me. I believe I have become not only a better researcher but also a better person due to our collaboration. Without his support and continuous encouragement several of the results presented in this thesis would not have been obtained.

I would also like to express my gratitude to the members of my Ph.D. thesis committee, Prof. Gael Dias (University of Caen), Dr. Cyril Goutte (National Research Council Canada), Prof. Patrick Gallinari (University of Pierre & Marie Curie/Paris 6), Prof. Laurent Besacier (University of Grenoble-Alps) and Dr. Guillaume Vernat (Coffreo). I was honored to have them in my defense committee. In particular, I would like to thank my thesis reviewers, Prof. Gael Dias and Dr. Cyril Goutte for the time they invested, their valuable feedback on my work and their detailed comments in their reports.

For my graduate studies, I had the honor to be funded by Coffreo. First, I would like to thank the two founders of the company, Emmanuel Cudry and Dr. Guillaume Vernat, not only for trusting me with the role but also for the discussions we had and the valuable lessons I learned for leading a company. I would also like to thank everybody at Coffreo for making my visits at Clermont Ferrand interesting! I would particularly like to thank Dr. Pierre-Antoine Papon, Karim Sayabou and Dr. Guenaelle De Julis, for the long discussions and our collaboration during these years.

I am especially grateful to Prof. Eric Gaussier for the various interactions we had over the last three years. I would like to thank him for being a great research collaborator and for suggesting me interesting directions for my work.

I have been truly lucky to interact with many brilliant people at University of Grenoble Alps and Grenoble in general. Among them, a special thanks go to my friend Dr. Ioannis Partalas for his encouragement and advice during my thesis. I would also like to thank all the members of the AMA team, current and past, as I learned a lot of things from them and I shared many beautiful moments.

I would also like to deeply thank Anthie. Living with me, caring for and supporting my choices for almost eight years now made me a better person. Thank you for being patient and understanding all these years tolerating my idiosyncrasies and habits.

Last but not least, I would like to thank from the depths of my heart my family back in Greece: My parents Andreas and Olga and my brother Stathis. Your continuous encouragement, patience, care and love are giving me courage and confidence. Thank you for everything you did and continue doing for me.

Georgios Balikas
Grenoble, Fall 2017

Contents

1 Introduction **1**

1.1 Thesis statement and overview of contributions 3

1.1.1 Extending Probabilistic Topic Models 5

1.1.2 Word Embeddings and Text Mining Applications 8

1.1.3 Quantification-based algorithmic tools 9

1.2 Outline of the thesis 9

2 Preliminaries for Probabilistic Topic Models **11**

2.1 The Multinomial and Dirichlet Distributions 13

2.1.1 The Multinomial Distribution 14

2.1.2 The Dirichlet Distribution 14

2.1.3 Conjugacy between the Multinomial and Dirichlet distributions 15

2.2 From the term-document co-occurrence matrix to probabilistic topic models 16

2.2.1 Latent Semantic Analysis 16

2.2.2 Probabilistic Latent Semantic Allocation 17

2.2.3 Latent Dirichlet Allocation 19

2.2.4 Distributional Semantics 22

2.3 Multilingual Topic Models 23

2.3.1 Parallel and comparable corpora 23

2.3.2 Bilingual Latent Dirichlet Allocation 24

2.4 Summary 25

3 Preliminaries for Neural Networks **27**

3.1 Word Embeddings with shallow Neural Networks 28

3.1.1 The Skipgram Model 29

3.1.2 The Continuous Bag-of-Words Model 32

3.2 Text Representations using deep neural networks . . . 33
3.3 Cross-lingual Word Embeddings . . . 35
3.3.1 Bilbowa . . . 37
3.3.2 Concept Net . . . 38
3.4 Summary . . . 39

4 Incorporating Prior Knowledge of Text Structure to Topic Models 41
4.1 An overview of the relevant literature . . . 43
4.2 The coherent text segments . . . 44
4.3 Incorporating text structure to topic models . . . 46
4.3.1 segmentLDA: Integrating segment boundaries to LDA . . . 46
4.3.2 Copulas and random variables (intermezzo) . . . 49
4.3.3 copulaLDA: Integrating segment boundaries to LDA using copulas . . . 52
4.4 The Experimental Evaluation . . . 55
4.4.1 Intrinsic Evaluation . . . 59
4.4.2 Extrinsic Evaluation . . . 66
4.5 Summary . . . 69

5 Extending Bilingual Topic Models 73
5.1 An overview of the relevant literature . . . 76
5.2 Framework . . . 79
5.2.1 The bilingual LDA . . . 80
5.2.2 Incorporating text structure into bilingual topic models . . . 81
5.2.3 Extracting multilingual topics from comparable corpora . . . 84
5.2.4 Combining the two models . . . 86
5.3 Experimental Framework . . . 87
5.3.1 Intrinsic Evaluation . . . 90
5.3.2 Extrinsic Evaluation . . . 93
5.4 Summary . . . 96

6 Applications of word embeddings to text mining 101
6.1 Polylingual text classification . . . 102
6.1.1 The model for learning the polylingual embeddings . . . 103
6.1.2 The Experimental Evaluation . . . 105
6.1.3 Summary . . . 109
6.2 Multitask Learning with Neural Networks . . . 109

6.2.1 Multitask Learning for Sentiment Classification 111
6.2.2 The Experimental Framework 112
6.2.3 Experimental results 119
6.2.4 Summary 121
6.3 Cross-lingual text retrieval 123
6.3.1 A Wasserstein-alike distance for Cross-lingual Document Retrieval 124
6.3.2 The Experimental Framework 126
6.3.3 Summary 130
6.4 Chapter Summary 131

7 Concluding Remarks **133**
7.1 Summary of Contributions 133
7.2 Future Directions 134

Bibliography **137**

A Efficient Model Selection for Regularized Classification by Exploiting Unlabeled Data **157**
A.1 Related Work 158
A.2 Accuracy and Macro-F1 Quantification Bounds 159
A.3 Experimental Framework 164
A.4 Summary 168

List of Figures

1.1 An overview of the contributions of the dissertation. 6

2.1 Applying the Vector Space Model to a document collection. 12
2.2 The graphical representation of LDA. 20
2.3 The graphical model of bilingual LDA. 24

3.1 Word Embeddings as a Neural Network Layer. 29
3.2 The skipgram model. 31
3.3 The continuous bag-of-words model. 31
3.4 An unrolled Recurrent Neural Network. 33
3.5 The Long Short-Term Memory Network. 34

4.1 The topic identified by LDA on a Wikipedia document. 42
4.2 Shallow parsing of a sentence with the Stanford Parser. 45
4.3 The graphical model of *segLDA*. 46
4.4 The probabilistic integral transform 50
4.5 Samples from a Frank copula while varying λ. 51
4.6 The copulaLDA generative model. 53
4.7 The effect of rejection sampling in the performance of copulaLDA . 59
4.8 The perplexity achieved by several topic models on English datasets. 62

5.1 Comparable Wikipedia documents in English and Portuguese. . . . 74
5.2 Graphical models of bilingual topics models. 75
5.3 The perplexity achieved by several topic models on bilingual datasets. 94

6.1 The generation of polylingual embeddings with an autoencoder. . . 105
6.2 Evaluating the polylingual embeddings on document classification. 108
6.3 A bidirectional LSTM architecture for multitask learning. 113
6.4 F_1 scores using the nbow+ representations. 121

A.1 Model selection process for SVM for macro-averaged F-measure. . . 167

Chapter 1

Introduction

We live in an interconnected world where new information technologies enable a fast flow of data and information. As a result of the ongoing growth of the World Wide Web, whose omnipresence in our lives persistently changes the way we make decisions and even behave, data are constantly produced at massive volumes. In this era of big data, the access to large amounts of data together with the ever-faster computing machines and data storage facilities have created huge research opportunities. It is therefore, nowadays, possible to study problems and extract valuable information at a scale and granularity that are truly unprecedented.

Among the different types of the data being made available online, text is arguably one of the most pervasive and persistent sources of information. Content analysis of text, that in its broad sense may refer to methods for studying and retrieving information from documents, has been traditionally achieved by close reading and manually coding the retrieved information. However, the voluminous amounts of data available today make it impossible for a single person or groups of people to manually examine text resources of tremendous size. On the other hand, being able to analyze and understand what is discussed online is a critical task and successfully accomplishing it has a huge potential for several real-world applications.

At the same time, the properties of text that is published online in the World Wide Web continuously change. Recent statistics reported, for instance, that the non-English Web content represents more than half of the information that is available on the Internet.[1] This entails that (i) in order to discover and exchange

[1] http://www.internetlivestats.com/internet-users/#byregion

knowledge at world-wide scale one needs algorithms and models capable of modeling and mining text beyond English, (ii) English may not be anymore the *lingua franca* of the Web as the increasing multilingual content pushes users towards consuming content written in their native language(s).

But multilinguality is not the only challenge of today's Web content. The media that people use to express their opinions or publish content also evolve: other than formal documents such as news articles, micro-blogging platforms like Twitter[2] are extremely popular and have become ubiquitous today. Their omnipresence, however, poses major algorithmic challenges due to the specificities of the content published there. For instance, tweets, which are short messages of up to 140 characters published on Twitter, pose several problems due to their style as symbols, abbreviations, slung and creative language are heavily used. As such media have largely democratized online content publishing and sharing, analyzing their data is also of great importance.

The above observations motivate a set of requirements when developing modern systems in order to be able to cope with the vast amounts of available data. These requirements, grouped by the particular characteristics of the data, can be summarized as follows:

Rq. 1) Considering the data type *i.e.*, text, as well as the fact that humans produce text in an ordered way following particular morpho-syntactic rules, we need to be able to take advantage of the structure that is inherent in text data. Such structure may be, for instance, the grouping of words in thematically coherent text spans such as sentences or noun-phrases. Moreover, given the variability in the style of text, we need to be able to represent it efficiently.

Rq. 2) Considering the large number of languages online, we need to be able to represent text written in different languages using language independent representations so that different tasks like prediction can be accomplished across languages. Further, these representations should be able to take advantage of the large amounts of multilingual data, especially those that are unlabeled and cheap to obtain.

Rq. 3) Considering the large amount of unlabeled data, one needs to be able to use them in order to develop both more expressive and semantically rich representations and better performing systems.

[2] `http://www.twitter.com`

Text Mining It follows from the above that *text mining*, the interdisciplinary area that the research of this dissertation belongs to, spans and borrows techniques from the domains of machine learning, natural language processing, information retrieval as well as data mining. The added requirement, imposed by the unprecedented availability of multilingual content, for efficient models that can digest and even benefit from text written in multiple languages is central for the results of the dissertation. Notably, any findings in this domain of (multilingual) text mining have a great potential for improving the performance of applications that affect our everyday lives significantly. Motivated by the above points, our driving force is the importance of text mining techniques towards understanding large amounts of text, possibly written in more that one languages, and providing efficient and effective solutions to problems that are met when dealing with such data. To this end, we propose models as well as algorithmic tools that address various challenges that arise in this fascinating research area.

1.1 Thesis statement and overview of contributions

This dissertation is organized in two parts and contributes models, tools and observations to problems that arise in the area of text mining with a focus on *multilingual* text mining. We build upon effective models and algorithms that address the three main requirements stated above (Rq. 1 - Rq. 3). In particular, the contributions of the dissertation aim at:

i.) Proposing probabilistic topic models that handle one or more input languages. The models incorporate prior knowledge of text structure in the form of boundaries of thematically coherent text spans.

ii.) Exploiting the rich semantic properties of word embeddings, that are vector representations of words that capture their semantic and syntactic properties. They can be used as another source of prior knowledge for (multilingual) topic models or as compact text representations in order to improve the performance achieved for various natural language processing tasks.

Distributional Hypothesis The shared idea that links the models of the dissertation stems from an insight which was perhaps first formulated by Harris [77] who suggested for linguistic items that "If A and B have almost identical environments ... we say that they are synonyms." Another, perhaps more famous, statement

of this principle was formulated from Firth [58]: "You shall know a word by the company it keeps". These statements describe the *distributional hypothesis*, that suggests that linguistic items with similar distributions have similar meanings. Distributional methods build on the distributional hypothesis and propose ways to compute the meaning of a linguistic item using the distribution of words around it. There is a plethora of computational models implementing distributional methods. Both the topic models used for (i.) and the word embedding models used for (ii.) belong in this family as they use word co-occurrence statistics to learn efficient word and document representations. Their difference lies on how they model co-occurrence as well as on the computational means they use to induce the representations. Topic models use a probabilistic framework and aim at modeling an underlying generative story which dictates a set of conditional independences between random variables which enables Markov Chain Monte Carlo inference algorithms for inference. Meanwhile, popular models for learning embeddings rely on a supervised prediction task. We will elaborate more on the similarities and differences of those models and we will evaluate the performance of their extensions at different tasks either when the documents of a collection are written in a single or more languages.

The dissertation contributions regarding point (i.) constitute the first part of the thesis. We utilize linguistic tools like shallow parsers and statistical tools like copulas to extend probabilistic topics models by incorporating parts of text structure. In particular, for the former point we show how to identify thematically coherent groups of words using either linguistically motivated tools or statistically motivated approaches. This results in a hierarchical document representation where a document is decomposed as a set of coherent segments; further each segment is a set of words. In both cases, by taking advantage of observations concerning which words often co-occur, we manage to better model and uncover the topics discussed in a collection. Such knowledge can be then used within a plethora of applications including text classification, document retrieval as well as collection exploration and visualization.

The dissertation contributions regarding point (ii.) constitute the second part of the thesis. We propose models and algorithms that utilize the expressiveness of word embeddings for the benefit of information retrieval and natural language processing applications. We argue that several applications can benefit from the ability of embeddings to capture the semantics of words and we demonstrate how

this can be accomplished for the tasks of text classification of short and long documents, polylingual text classification where one has access to translations of a document and cross-lingual document retrieval.

A third, shorter part of the work performed during the thesis is put in Appendix A. It details algorithmic tools for fast model selection for regularized classification by exploiting unlabeled data. While the main thesis is devoted to models built on the distributional hypothesis, that concerns the word co-occurrence in the document level, this last part of work models properties in the collection level. In particular, we utilize the distribution of document categories within a collection and unlabeled data that are usually cheap to obtain to accelerate model selection and hyperparameter tuning for classification models that use regularization like Support Vector Machines. To this end, we observe that the assumption of identically and independently distributed documents (i.i.d.) between the training and the test parts of a collection, which is common in several document classification settings, may be used to accelerate model selection. Being able to estimate the distribution of categories in unseen documents, motivates learning theory bounds that, in turn, accelerate the process of hyperparameter tuning. Therefore, the work explores how the distribution of categories on unlabeled data can be approximated and evaluates the proposed bounds for model selection.

Next, we provide an overview of the contributions of the dissertation following the above points. Meanwhile, Figure 1.1 provides an overview of the thesis.

1.1.1 Extending Probabilistic Topic Models

Incorporating text structure to probabilistic topics models. *Given a document segmentation denoting the partition of a document to coherent text spans, how can probabilistic topic models be extended to incorporate that knowledge?*

The exchangeability assumption in topic models like Latent Dirichlet Allocation (LDA) often results in inferring inconsistent topics for the words of text spans like noun-phrases, which are generally expected to be topically coherent. We propose *segmentLDA* and *copulaLDA*, two novel topic models that extend LDA by integrating part of the text structure and relax the conditional independence assumption between the word-specific latent topics given the per-document topic distributions. The novel models assume that the words of text spans like noun-phrases are topically bound. The former model (*segmentLDA*) forces all words within a segment to be assigned to the same topic and the binding between topics within

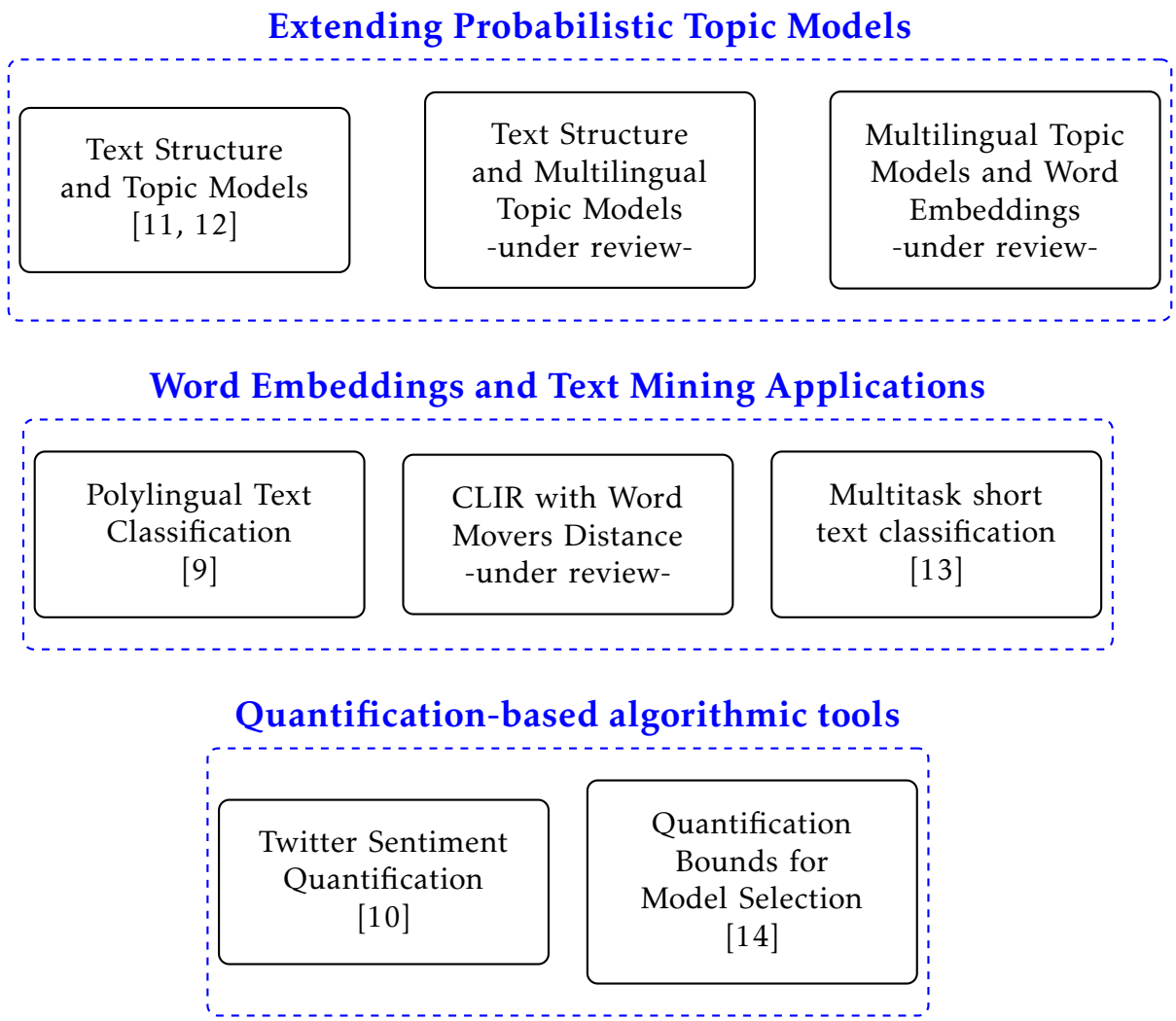


Figure 1.1: An overview of the contributions of the dissertation.

segments is maximal as all word specific topics are equal. The latter (*copulaLDA*) uses copulas when assigning topics to the words of sentences and is therefore more flexible as the strength of the bound between the topics is controlled by the free parameters of the copulas. To demonstrate the efficiency of the novel topic models we conduct experiments on several text datasets consisting of documents in English. We assess the quality of the produced topics using the normalized point-wise mutual information scores, the generalization performance of the models measured by perplexity and the learned document representations as inputs to a document classification task. For this purpose we compare for each topic model a variety of segments that can be considered to be coherent. Our analysis reveals the benefits of integrating prior knowledge of text structure in topic models as well as the advantages of having flexible models and segments of various sizes to accomplish that. (Chapter 4).

Bilingual topics models for comparable corpora. *Given pairs of documents that discuss the same themes to some extend, how can we extend bilingual topic models to better adapt them for such inputs?*

Probabilistic topic models like Latent Dirichlet Allocation (LDA) have been previously extended to the bilingual setting. A fundamental modeling assumption in several of these extensions requires the input documents to be exact translations between them. However, this assumption is strong for comparable corpora, which are, in turn, the most commonly available or easy to obtain. In this chapter we relax this assumption by proposing a binding mechanism between the distributions of the paired documents. The strength of the bound depends on each pair's semantic similarity, that we propose to estimate using bilingual word embeddings learned with shallow Neural Networks. We evaluate the proposed method by extending two topic models: a bilingual adaptation of LDA that assumes bag-of-words inputs and a model that naturally extends those proposed in Chapter 4 in order to incorporate part of the text structure in the form of boundaries of semantically coherent segments. To demonstrate the efficiency of the novel, bilingual topic models we conduct experiments on four bilingual, comparable corpora of English documents with French, German, Italian and Portuguese documents. The obtained results demonstrate the efficiency of our approach in terms of topical coherence measured by the normalized point-wise mutual information, generalization performance measured by perplexity and accuracy in a cross-lingual document retrieval task for each of the language pairs (Chapter 5).

1.1.2 Word Embeddings and Text Mining Applications

Polylingual Text Classification *How can translations of a document be used to improve the performance in the task of document classification?*

We propose a polylingual text embedding strategy, that learns a language independent representation of texts using Neural Networks. We study the effects of bilingual representation learning for text classification and we empirically show that the learned representations achieve better classification performance compared to traditional bag-of-words and other monolingual distributed representations. Our results also demonstrate that the performance gains are more significant in the interesting case where only few labeled examples are available for training the classifiers (Chapter 6.1).

Multitask Learning for Fine-Grained Twitter Sentiment Analysis *How can we improve the performance of short text sentiment classification using information from correlated tasks? How deep neural networks perform in the task?*

Traditional sentiment analysis approaches tackle problems like ternary (3-category) and fine-grained (5-category) sentiment classification by learning the tasks separately. We argue that such classification tasks are correlated and we propose a multitask approach based on a recurrent neural network that benefits by jointly learning them. Our study demonstrates the potential of multitask models on this type of problems and improves the state-of-the-art results in the fine-grained sentiment classification problem (Chapter 6.2).

Cross-lingual Document Retrieval Using Word Embeddings *How can we use the tools developed for the transportation problem to calculate the distance of documents written in different language and perform cross-lingual document retrieval?*

We extend Word Mover's Distance, a recently proposed distance function between documents for the task of cross-lingual document retrieval (CLDR). We show that the metric can naturally incorporate various term weighting schemes and that it benefits from high quality multilingual word embeddings. Using word embeddings that incorporate information from a Knowledge Base we show that our method outperforms state-of-the-art baselines on six CLDR problems by a large margin in terms of evaluation measures like Mean Reciprocal Rank and P@1 (Chapter 6.3).

1.1.3 Quantification-based algorithmic tools

Quantification-Based Bounds for Supervised Model Selection *Assuming access to identically and independently distributed data, how can we accelerate the model selection process of regularized classification systems using quantification?*

Hyper-parameter tuning is a resource-intensive task when optimizing classification models. The commonly used k-fold cross validation can become intractable in the large scale settings as a classifier should learn billions of parameters. At the same time, in real-world, one often encounters multiclass classification scenarios where only a few labeled examples are available; model selection approaches often offer little improvement in such cases and the default values of learners are used. We propose bounds for classification on accuracy and macro measures (precision, recall, f_1 measure) that motivate efficient schemes for model selection and can benefit from the existence of unlabeled data. We demonstrate the advantages of those schemes by comparing them with k-fold cross validation and hold-out estimation in the setting of large scale classification (Appendix A).

1.2 Outline of the thesis

The rest of the dissertation is organized as follows: The next two chapters present the basic concepts and background material that the author believes are essential for the topics discussed in the thesis. In particular,

- Chapter 2 is a concise introduction to probabilistic topics models, and
- Chapter 3 presents basic concepts used for representation learning with neural networks.

Then, the next two chapters are devoted to our work and contributions concerning topic models:

- Chapter 4 presents the contributions when integrating parts of text structure to monolingual topic models, while
- Chapter 5 presents an adaptation of bilingual topic models for comparable corpora.

The remaining of the contributions concern the use of word embeddings for text mining tasks:

- Chapter 6 presents work concerning the improvements one may achieve using word embeddings for polylingual classification, multitask classification and cross-lingual document retrieval.

Having presented the main thesis contributions, in Chapter 7, we offer our concluding remarks about the topics that are discussed in this thesis and we describe promising future research directions.

Lastly, Appendix A which is self-contained, introduces the task of quantification as well as an algorithmic tool where quantification-based classification bounds are derived whose application accelerates model selection.

Chapter 2

Preliminaries for Probabilistic Topic Models

Document collections are on the basis of every text mining application. A collection C is a set of documents: $C = \{d_1, \ldots, d_M\}$ where each document is a set of ordered words $d_i = (w_1, \ldots, w_N)$. As the amount of data available today is unprecedented, a fundamental challenge is its analysis, especially when those data are in the form of unstructured text like documents. The challenge therefore decouples in being able to organize large amounts of text data without requiring manual labor, as the later is expensive and scales poorly for such problems. To this end, efficient algorithms that can describe the themes discussed in documents without any type of annotations are important.

Given a document collection, the first problem that naturally arises is how one can describe it in a vectorized format that would be suitable for further calculations. The vector space model [156] (VSM), is an algebraic model for representing text as a set of vectors of identifiers. The identifiers aim at modeling the occurrence (or absence) of terms in a given document. According to the VSM, these identifiers are the indexes of the terms of the vocabulary of the collection and their values are each term's frequency.

To better describe the application of VSM to documents and collections, Figure 2.1 shows an example. The text excerpt is vectorized such that the non-zero elements express the amount of information conveyed by the frequency of the vocabulary words within the excerpt. The first element of the vector, for example, models the word "information" and is populated with 3 as "information" occurs thrice in the excerpt. Moving from the document level to the collection level (bottom part of Fig. 2.1) the vectorization process is applied to each of the N documents of the collection. Therefore, instead of a single vector, the output of the collection

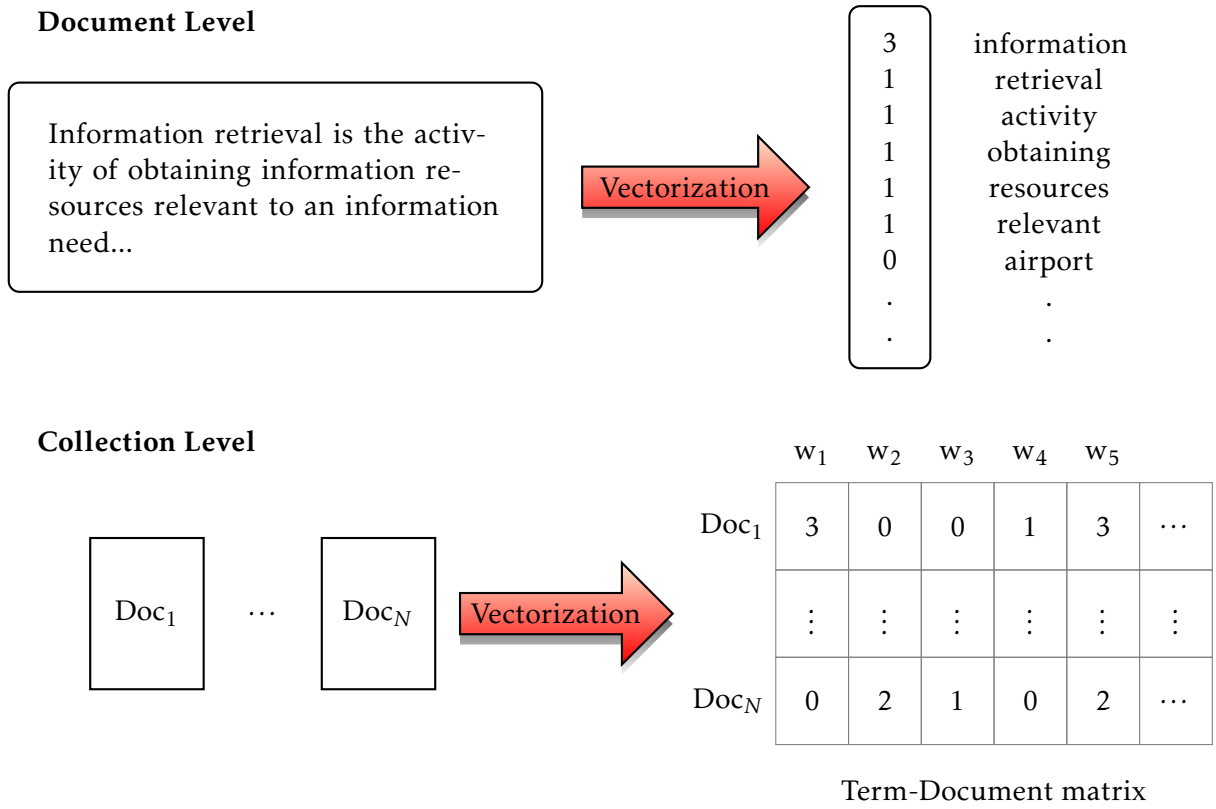


Figure 2.1: The Vector Space Model for a document (above) and a collection of documents (below). A document is transformed to a vector with non-zero indexes corresponding to the frequency of the words associated to these indexes. Repeating the process for each document of a collection creates the term-document matrix.

vectorization process is a matrix, namely the *term-document co-occurrence matrix* that comprises N vectors.

Notice how the bag-of-words assumption is inherent to the vectorization process: the order of the words within documents is ignored. In the example of Figure 2.1, independently of the word order the process would result in vectors whose non-zero elements would always be the same. Also, the produced vectors and the term-document co-occurrence matrix are very sparse. The vector dimensionality equals the size of the vocabulary (V) of the collection so that the occurrences can be modeled. The sparsity stems from the fact that only few unique words compared to V appear is each document.

The vector space model received a lot of attention partially due to the performance improvements on a number of applications produced on its basis. Despite

its success there are several limitations. Two of them, that motivate the contributions of this manuscript, are:

i. The inability of the model to capture the semantics of the words. As each word is represented by an index in the resulting vector, the semantic relatedness of the words is not captured. Furthermore, the properties of synonymous or polysemous words can not be modeled.

ii. The exchangeability assumption that results in the bag-of-words representation. As shown above the order of words as well as the way they are grouped in phrases, sentences etc. is lost during the vectorization step and the resulting vectors are of big dimensionality.

As shown above, the resulting matrix that models the occurrence of words in the documents of a collection can be big, noisy and sparse. Given this term-document matrix, one may feel that there should be some structure or pattern in the way that words occur in documents or co-occur with other words. The models presented in this chapter aim at uncovering these patterns. The outputs of the approaches to be presented can be used to estimate semantic similarities of text spans ranging from words to larger text passages.

In the rest, we briefly review basic concepts of topic modeling which will serve as a fundamental theoretical background for the remainder of this thesis. It illustrates how topic models can be seen as computational models that implement the distributional hypothesis. The concepts to be presented are not covered in their entirety as the purpose of the demonstration is to be used as a concise overview which is accompanied by references for further reading. Furthermore, the chapter focuses only on introductory material that the author considers necessary for a deeper understanding of the rest of the text.

2.1 The Multinomial and Dirichlet Distributions

We begin with a short overview of probability distributions that will be used when developing the topic models. Along with the definitions of these distributions we introduce the notation that will be used in the rest of the manuscript.

2.1.1 The Multinomial Distribution

One of the important distributions that is extensively used for topic modeling is the multinomial distribution [8]. The multinomial distribution models the outcome of a k-sided dice when rolled n times. It gives the probability of any particular combination of numbers of successes for the various categories, when n independent trials are performed. The outcome of a trial is a success for exactly one of the k categories. If p_i is the probability of the outcome i, it is required for each trial that $\sum_{i=1}^{k} p_i = 1$.

The multinomial distribution generalizes other distributions in various ways. In particular:

- If for the number of trials one has $n = 1$ and for the number of outcomes has $k = 2$, the multinomial distribution is the Bernoulli distribution.
- If $n = 1$ and $k > 2$ it is the categorical distribution, which is equivalent to the result of rolling a k-sided dice once.
- If $n > 1$ and $k = 2$ it is equivalent to the binomial distribution.

In terms of notation, assuming k outcomes and n trials, $Mult(n, \boldsymbol{p})$, where $\boldsymbol{p} = (p_1, \ldots, p_k)$ denotes the outcome of the multinomial experiment with probabilities $\boldsymbol{p}$. From the above, letting $Cat(\boldsymbol{p})$ to be the result of a draw from a categorical distribution, it follows that:

$$Mult(1, \boldsymbol{p}) = Cat(\boldsymbol{p}).$$

The probability density function of a multinomial distribution parametrized by $\boldsymbol{p}$ that estimates the probability of observing the i-th event x_i times, when we have n events in total is:

$$p(x_1, x_2, \ldots, x_k) = \frac{n!}{x_1 \cdot x_2 \cdots x_k} p_1^{x_1} \cdots p_k^{x_k}, \text{ with } \sum_{i=1}^{k} x_i = n.$$

2.1.2 The Dirichlet Distribution

The Dirichlet distribution [8], denoted $Dir(\boldsymbol{\alpha})$ is a family of multivariate continuous probability distributions. It is a probability distribution over the space of multinomial distributions, *i.e.*, to generate data X from a Dirichlet distribution with parameters $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_k$ you first draw a $\boldsymbol{p} \sim Dir(\boldsymbol{\alpha})$, and then draw the

data $X \sim Mult(n, \boldsymbol{p})$. Therefore, the Dirichlet distribution is a distribution over distributions controlled by $\boldsymbol{\alpha}$. Compared to the standard multinomial draws, the Dirichlet distribution introduces an extra layer with parameters $\boldsymbol{\alpha}$ that controls the probabilities $\boldsymbol{p}$ according to which the data X are generated. The probability density function of the Dirichlet distribution is

$$p(x_1, \ldots, x_k | \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{a_i - 1}, \text{ where } x_1, \ldots, x_k > 1, \sum_{i=1}^{k} x_i = 1,$$

where $B(\boldsymbol{\alpha})$ is the multivariate Beta function that is equivalent to:

$$B(\boldsymbol{\alpha}) = \frac{n!}{\prod_{i=1}^{k} \alpha_i!}, \text{ where } n = \sum_{i=1}^{k} \alpha_i.$$

2.1.3 Conjugacy between the Multinomial and Dirichlet distributions

The Dirichlet distribution is the conjugate prior of the multinomial distribution. This means that if the prior distribution of the parameters $\boldsymbol{p}$ of a multinomial follow is Dirichlet, then the posterior distribution is also a Dirichlet. This has the benefit of making the posterior distribution easy to calculate, and the Multinomial-Dirichlet conjugates are commonly used for topic modeling. To highlight this, let $(p_1, \ldots, p_k)$ be the multinomial parameters and assume that they are sampled from:

$$(p_1, \ldots, p_k) \sim Dir(\alpha_1, \ldots, \alpha_k),$$

prior to having access to data observations. After observing data $X = (x_1, \ldots, x_k)$, for these $(p_1, \ldots, p_k)$ where x_i denotes how many times event i occurred, the parameters $\boldsymbol{p}$ for our beliefs may be updated as:

$$(p_1, \ldots, p_k) | X \sim Dir(\alpha_1 + x_1, \ldots, \alpha_k + x_k).$$

This signifies that the data observations update our beliefs and this is modeled having the data to be pseudo counts added to the Dirichlet parameters $\boldsymbol{\alpha}$.

Another important comment concerns the role of the $\boldsymbol{\alpha}$ parameters in the samples of the Dirichlet. The smaller the values of $\boldsymbol{\alpha}$, the sparser the obtained sample is. Therefore, in the case of the Multinomial-Dirichlet example demonstrated above, if most of the elements of $\boldsymbol{\alpha}$ are $\alpha_i \ll 1$, then the sampled values $(p_1, \ldots, p_k)$ will be sparse, which means that only a few events will have high probability p_i and the rest of them very low. This property is important for topic models, as this is how sparsity is imposed.

2.2 From the term-document co-occurrence matrix to probabilistic topic models

As we have previously discussed, the term-document co-occurrence matrix is an approach to represent the occurrence of words in the documents of a collection. We have argued that this matrix can be big and noisy and one may expect that some patterns occur in it. In this section we introduce approaches previously proposed to reveal parts of the structure of this matrix whose goal is to model the meaning and the semantics of words as well as the thematical content of the documents that these words occur to.

One may identify two families of approaches. The models of the first try to decompose a word-context matrix by relying on matrix decomposition methods. Popular models of this family include Latent Semantic Analysis (LSA) [101] and the Hyperspace Analogue to Language [117]. The second family of methods was mainly motivated by the success of models like LSA. These models alleviate some of the limitations of LSA and also provide more interpretable outputs. To this end, they rely more on probabilistic groundings and utilize latent variables to model the latent themes that are assumed to generate the documents of a corpus. Popular models of this family are the Probabilistic Latent Semantic Analysis (pLSA) [84] and the Latent Dirichlet Allocation (LDA) [23]. They take as input the documents of a collection and return a number of topics that can be used to semantically describe their content.

The tools used by the models of the first family are inspired by Linear Algebra or geometry, while the second family of probabilistic models employs Bayesian approaches. Both families of approaches allow for calculating the similarity between terms: spatial models compare terms using distance metrics in a high-dimensional space, while probabilistic models measure similarity between terms according to the degree to which they share the same topic distributions [37]. In the rest of the section, we briefly review LSA (Section 2.2.1) and pLSA (Section 2.2.2) and then describe the Latent Dirichlet Allocation (Section 2.2.3) in more detail.

2.2.1 Latent Semantic Analysis

Latent Semantic Analysis (also known as Latent Semantic Indexing) [101] is an algebraic method used to analyze the term-document co-occurrence matrix. For LSA the context of word is the document it appears (the term-document matrix is

used), while this varies for other methods: for the Hyperspace Analogue to Language the context is the surrounding words and instead of the term-document matrix a term-term matrix is used whose frequencies are calculated using a sliding window.

The goal of LSA is to find a low-rank approximation of the term-document co-occurrence matrix which results in the combination of some of the dimensions that may depend on several terms. Given a collection $C = \{d_1, \ldots, d_M\}$ whose vocabulary is $V = \{w_1, \ldots, w_N\}$, the term-document matrix is $X \in \mathbb{R}^{M \times N}$ and $x_{i,j}$ denotes the number of the occurrences of word w_j in d_i. X can also be transformed as a result of the application of a term-weighting scheme like the term frequency-inverse document frequency (tf-idf) scheme [164]. As rows of the matrix represent documents, the dot product between them calculates document similarity *i.e.*, the more common terms two documents contain the more similar they are. Further, column product represents word similarity *i.e.*, terms that occur in the same documents should be similar.

The fundamental part for LSA is the application of the truncated SVD algorithm that approximates X with the product of three other matrices:

$$X \approx U_t \Sigma_t V_t^T, \tag{2.1}$$

where t denotes the number of the largest singular values that are kept. For more details on SVD we refer the interested reader a Linear Algebra textbook (e.g., [167]). Due to the decomposition that results in the combination of some dimensions, in the reduced dimensional space a term can be calculated to be similar with others if they have occurred in similar contexts, regardless of whether those contexts are in the same documents. One drawback of LSI is that it lacks in terms of a solid probabilistic foundation and it may be difficult to interpret the resulting word representations [23].

2.2.2 Probabilistic Latent Semantic Allocation

Probabilistic Latent Semantic Analysis (pLSA: also known as probabilistic latent semantic indexing pLSI) [84] is another model that can be used to analyze the term-document co-occurrence matrix. The model has evolved from LSA and instead of relying on a matrix decomposition approach like SVD it is based on a *mixture* decomposition. Latent topics are assumed to have generated the collection's documents and the goal is to identify them. PLSA can be considered as a

generative model, although strictly speaking it is not one [23] due to its inability to model unseen documents.

The model assumes that topics are distributions over the words of the vocabulary of a collection and are modeled as multinomials. This means that given a particular topic k, identified by the value of the random variable z, a word w has conditional probability of occurrence such as:

$$p(w|z=k) = \phi_{k,w} \text{ and } \sum_{w=1}^{V} \phi_{k,w} = 1.$$

For instance, word like "ball" and "athlete" would have higher probability given a topic "Sports" than given a topic "Science".

Further, the model assumes that each document is associated with a distribution over the topics of a collection. For the conditional probability of a topic k in a document d one has:

$$p(z=k|d) = \theta_{d,k} \text{ and } \sum_{k=1}^{K} \theta_{d,k} = 1.$$

The generative process for the documents of the collection is then:

- For each document d:
  - For each word position i within d:
    - Choose a topic $z \sim Mult(1, \theta_d)$
    - Choose a word $w \sim Mult(1, \phi_z)$

Then, the probability of each occurrence of a word in a document, that is the probability of a non-zero element of the term-document matrix, is modeled as a mixture of conditionally independent multinomial distributions:

$$p(w,d) = \sum_{z=1}^{K} p(z)p(d|z)p(w|z) = p(d)\sum_{k} p(z|d)p(w|z), \tag{2.2}$$

where $k \in [1,K]$ is the topic identifier and z the random variable that denotes the topic. These are two, equivalent formulations of the joint probability $p(w,d)$ of pLSA. Applying the Bayes rule reveals their equivalence [85]:

$$p(w,d) = p(d)\sum_{z=1}^{K} p(z|d)p(w|z) = p(d)\sum_{z=1}^{K} \frac{p(d|z)p(z)}{p(d)}p(w|z) = \sum_{z=1}^{K} p(z)p(d|z)p(w|z).$$

During inference one needs to estimate the parameters of the model, that is $\theta_{d,k}$ and $\phi_{k,w}$, which can be achieved by applying and an expectation-maximization algorithm [84, 85]. One pitfall is the lack of parameters for $p(d)$, so we don't know how to assign probability to a new document. Another is that the number of parameters for $p(z|d)$ grows linearly with the number of documents, which may lead to overfitting. To overcome this limitations, Latent Dirichlet Allocation that is presented in the next section extend pLSA by proposing a complete generative story.

2.2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is probably the most popular and representative topic model today. It was first proposed by Blei et al. [23], and its introduction has motivated a large part of research work as well as applications in several domains such as text classification systems [23], image processing [36], computational biology analyses [110] and countless others.

LDA offers a robust way for identifying the topics and builds on a principled set of assumptions that enable efficient inference algorithms compared to pLSA. Significant research attempts have been devoted to relaxing the assumption that govern LDA.

The generative story of LDA consists of the following steps:

- for each topic $k \in [1, K]$: sample per-word topic distributions $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$
- for the document d_i, $i \in [1, M]$:
  - sample the per-document topic distribution $\boldsymbol{\theta_i} \sim Dir(\boldsymbol{\alpha})$
  - for the word position n of d_i, $n \in [1, N_i]$
    - Sample the topic z of the word: $z_{i,n} \sim Mult(1, \boldsymbol{\theta}_i)$
    - Sample the term for the word position: $w_{i,n} \sim Mult(1, \boldsymbol{\phi}_{z_{i,n}})$

The generative story of LDA is a process that results in the terms of corpus $w_{i,n}$ partitioned into documents d_i. The number of the topics K as well as $\boldsymbol{\alpha}, \boldsymbol{\beta} : \boldsymbol{\alpha} \in \mathbb{R}^K, \boldsymbol{\beta} \in \mathbb{R}^V$ that are priors of the Dirichlet per-document and per-words distributions are required for the generative process. First, the per-word topic distributions $\boldsymbol{\phi}_k$ are sampled for the whole corpus. To achieve that, for each document d_i a topic proportion $\boldsymbol{\theta}_i$ is sampled, and from this topic proportion the document terms are emitted: for each word position a topic $z_{i,n}$ is sampled which denotes the topic that

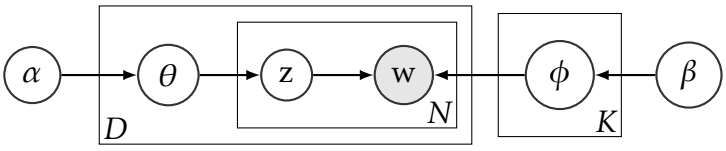


Figure 2.2: The graphical representation of LDA.

will generate the word and, finally, the word is drawn from that topics per-word distribution $\boldsymbol{\phi}_{z_{i,n}}$.

Complementary to the generative process described above, Figure 2.2 depicts the graphical model of LDA. The plate D denotes the documents of a collection. For each document the topic mixture θ is sampled from $Dir(\boldsymbol{\alpha})$. Then, for each word position (plate N) as topic z is sampled. Given the topic z and the per-word topic distributions $\boldsymbol{\phi}$ the terms of the document $\boldsymbol{w}$ are sampled. In the figure, the node denoting the terms is gray because the words of the documents are observed. Due to the fact that in reality we observe the documents and, therefore, we know their size in words, in the generative story we omitted the step where the size of the document is sampled. In some resources, one may find an additional step where the document size N_i is sampled, usually from a Poisson distribution. However, this choice does not affect the inference steps and can be safely omitted.[1]

We further elaborate here on our comment above that LDA is a mixture model. Mixture models use a convex combination of some base distributions in order to model the observations. A convex combination refers to a weighted sum over some base observations, whose sum of weighs equals to one. Due to the probabilistic nature of LDA and its flexibility to assign words of a document to different topics, for the topic proportions of a document one has:

$$\sum_{k=1}^{K} p(z=k) = \sum_{k=1}^{K} \theta_k = 1. \tag{2.3}$$

Further, for the per-word topic distributions that model the probability of a word given a topic:

$$\sum_{k=1}^{K} p(w|z=k) = \sum_{k=1}^{K} \phi_{z_k} = 1. \tag{2.4}$$

As documents are mixture of topics (Eq. (2.3)) and topics themselves are mixtures of word probabilities (Eq. (2.4)) LDA is also referred to as an admixture model. Admixture denotes mixtures whose basic components are mixtures themselves.

[1] In the rest of the dissertation we omit the steps when sizes of documents or other sub-document text spans are sampled, provided they are observed during inference.

Inference While the generative story and the graphical representation of LDA describe an iterative process that is assumed to have generated a document collection, inference tries to achieve the opposite. Instead of generating the corpus, an inference strategy aims at discovering LDA's parameters when observing the words of a corpus. The parameters of the model are the per-document topic distributions ($\boldsymbol{\theta}_i$) and the per-word topic distributions ($\boldsymbol{\phi}$). Estimating those is equivalent to uncovering the latent themes (topics) of a collection. In particular, given $\boldsymbol{\phi}$ one may identify the words with the highest probability for a topic while $\boldsymbol{\theta}_i$ is a vector representation of d_i in the space of the topics. As a result, documents with similar topic distributions are expected to be semantically similar.

The two, most popular inference strategies are variational inference [23] and collapsed Gibbs sampling [73]. We review the details of the collapsed Gibbs sampling approach as it will be further used for inference of the LDA-extensions that will be proposed in the next chapters. Gibbs sampling algorithms obtain posterior samples by sweeping through each block of variables and sampling from their conditional, while the remaining blocks are fixed. In practice, for LDA the algorithm initializes randomly the topics of words. Then, during the Gibbs iterations and until convergence, it samples topics for the words occurring within documents as Multinomial draws. The probabilities of the Multinomial draw for sampling the topic of a word position i where word t is observed are given by [78, 73]:

$$p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w}) \propto \frac{\Psi^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V} \Psi^{(t)}_{k,\neg i} + \beta_t} (\Omega^{(k)}_{m,\neg i} + \alpha_k) \tag{2.5}$$

where "$\neg i$" in a subscript of a count variable signifies the exclusion of the counts due to the word position i and, β_t, α_k are the t-th and k-th coordinate of $\boldsymbol{\beta} \in \mathbb{R}^V$ and $\boldsymbol{\alpha} \in \mathbb{R}^K$ respectively. Ψ and Ω are count variables, whose content is shown in Table 2.1 that summarizes the notation used for LDA. The Gibbs sampling algorithm is then an iterative process over the words of a collection C, where Eq. (2.5) is applied and a topic for each word is sampled until convergence. Although checking convergence for Markov Chain Monte Carlo approaches is a field of research (e.g., [113]), for topic models one may use as criteria how well semantically similar words of documents are clustered. From the count matrices Ψ and Ω one may yield the per-word and per-document topic distributions by normalizing their rows.

Symbol	Description
K	number of topics
V	The size of the vocabulary
α	concentration hyper-parameter of per-document topic distribution prior
β	concentration hyper-parameter of per-word topic distribution prior
k	topic variable placeholder, $k \in [1, K]$
i	document variable placeholder, $i \in [1, D]$
θ_i	topic distribution of the i-th document
ϕ	per word topic distribution
Ψ	Counter variable: per topic word assignments, $\Psi \in \mathbb{R}^{V \times K}$
Ω	Counter variable: per document word assignments, $\Omega \in \mathbb{R}^{D \times K}$

Table 2.1: Notation used for the development of Latent Dirichlet Allocation.

Inference on unseen documents As LDA has a complete generative story, one may identify the topic distributions of unseen (held-out) documents by performing the Gibbs sampling inference process on those documents. In this case, for the Ψ counters the values that were observed during training are used the model is queried. Typically very few iterations (<10) are needed for the topic distributions of the unseen documents to be inferred [78].

2.2.4 Distributional Semantics

The models presented in the previous sections are popular approaches used to represent the meaning of words or text spans. These methods, however, provide only a quantitative estimate of the semantic similarity (or meaning) between terms that is estimated by operations that quantify vector similarities. This is to differentiate them from other resources like ontologies or controlled vocabularies that have been extensively used to represent meaning or specific types of relationships, which can not be easily achieved by using the outcome of topic models.

Topic models like pLSA or LDA manage to determine the meanings (or semantics) of terms within a collection empirically from the way in which these terms are distributed across the text. From these distributional statistics, it is possible to obtain meaningful estimates of the semantic similarity between terms in an unannotated corpus of text without human intervention. This is why they can be seen as models that implement the distributional hypothesis. Being unsupervised and easily applicable to unannotated text as well as computationally efficient, is a strong advantage of these methods.

2.3 Multilingual Topic Models

As more and more multilingual content is becoming available online, there is a pressing need for developing models that can cope with text data written in different languages. Topic models like LDA manage to uncover the latent topics of a corpus and have been used to numerous applications. Their success has motivated research that resulted in bilingual and multilingual topic models [127, 178]. These are models that extend the concept of probabilistic topic models in the case where documents are written in more than a single language. The goal is not only to learn consistent topics for each language, but also to learn topics that are aligned across the input languages. Before providing an overview of these models, we first describe their inputs and discuss the concept of parallel and comparable corpora.

2.3.1 Parallel and comparable corpora

The section provides some basic definitions about the properties of multilingual corpora that are important for the rest of the presentation.

Definition 1 A *comparable corpus* in two or more languages $\ell_1, \ell_2, \ldots$ is a set of corresponding text collections $C^{\ell_1}, C^{\ell_2}, \ldots$. Each collection consists of documents such that $C^{\ell_1} = \{d_1^{\ell_1}, d_2^{\ell_1}, \ldots, d_{N_{\ell_1}}^{\ell_1}\}$, $C^{\ell_2} = \{d_1^{\ell_2}, d_2^{\ell_2}, \ldots, d_{N_{\ell_2}}^{\ell_2}\}, \ldots$ that discuss similar topics. It is not required for the documents of $C^{\ell_1}, C^{\ell_2}, \ldots$ to have one-to-one explicit alignments, that is the content of $d_1^{\ell_1}$ to be thematically comparable to the content of $d_1^{\ell_2}$ etc.. Therefore, it can also be $N_{\ell_1} \neq N_{\ell_2}$.

A characteristic example of a comparable corpus is the set of documents comprising the English and the French Wikipedia. The number of entries for the two languages (or any other two or more languages) varies. It is not necessary for an entry in English to have a counterpart entry in French and vice-versa. However, concerning the themes (topics) underlying the collection, one can safely assume though that at least, to some extent, the English and the French documents cover similar topics like Arts, Science, Geography, ... due to the nature of Wikipedia.

Definition 2 A *comparable corpus with explicit document alignments* is a comparable corpus in two or more languages $\ell_1, \ldots, \ell_k$, where $N_{\ell_1} = \cdots = N_{\ell_k}$. Further, the documents have explicit thematic alignments, that the contents of $d_i^{\ell_1}, \ldots, d_i^{\ell_k}$ are thematically comparable.

A comparable corpus with explicit document alignments is a special case of a comparable corpus that requires topical alignments between documents written

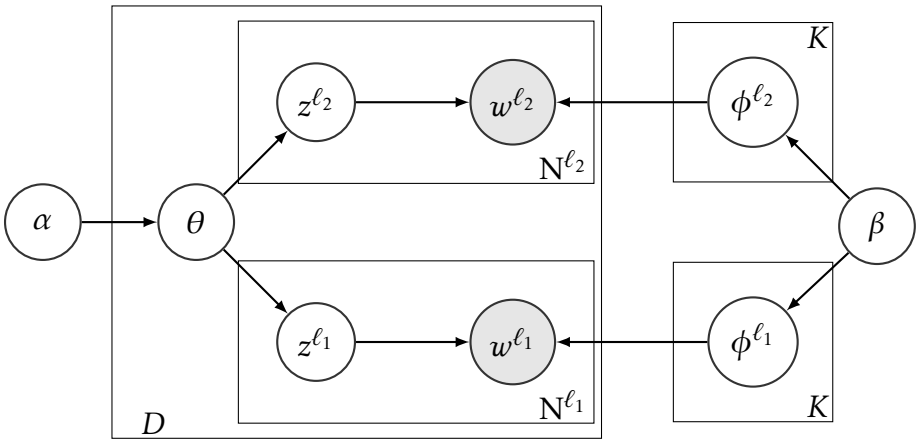


Figure 2.3: The graphical model of bilingual LDA.

in different languages. In the above-mentioned example of Wikipedia articles, one may obtain a comparable corpus with explicit document alignments by only keeping entries that have a counterpart entries in the rest of the languages. Assuming an entry about "Dog", there must be an entry about "Chien" (French for dog) and so forth.

Definition 3 A *parallel corpus* is a comparable corpus in two or more languages $\ell_1, \ell_2, \ldots$. The documents have explicit thematic alignments, and the content of $d_i^{\ell_1}$ is *identical* to the content of $d_i^{\ell_2}, \ldots$.

The most natural way to build or obtain a parallel corpus is by translating the documents of a language ℓ_1 to one or more languages $\ell_2, \ldots$ Parallel corpora are the most costly to develop as they require either human translations or high quality automatic translation systems. Notably, parallel corpora are very important for different types of applications like machine translation [96], multilingual topic models [178] and plenty others.

2.3.2 Bilingual Latent Dirichlet Allocation

Following the success of topic models like LDA, whose application on monolingual data has enabled various types of applications, topic models for multilingual content were proposed. Bilingual LDA[2] (BiLDA: Figure 2.3) is a direct extension of LDA in the bilingual setting. The input collection is assumed to be either a parallel [196] or a comparable one [137, 127, 44, 149]. Its generative story is as follows:

[2]Also commonly referred to as multilingual LDA depending on the number of the input languages

- for each topic $k \in [1, K]$: $\phi_k^{\ell_1} \sim Dir(\boldsymbol{\beta})$, $\phi_k^{\ell_2} \sim Dir(\boldsymbol{\beta})$
- for each document pair d_i:
  - sample $\boldsymbol{\theta_i} \sim Dir(\boldsymbol{\alpha})$
  - for each language $\ell \in \{\ell_1, \ell_2\}$
    - for each of the N_i^{ℓ} words:
      - sample $z \sim Mult(1, \boldsymbol{\theta_i})$
      - sample $w \sim Mult(1, \phi_z^{\ell})$

It can be seen from the generative story that BiLDA assumes that the documents of an aligned pair $d_i = (d_i^{\ell_1}, d_i^{\ell_2})$ have identical topic distributions as there is a single, shared θ_i topic distribution per pair. Also, as the model is a direct extension of LDA assumes the documents to be a bag-of-words.
The collapsed Gibbs sampling updates [178] for the topic of word j of document d_i is $\forall \ell \in \{\ell_1, \ell_2\}$:

$$p\left(z_{ij}^{\ell_1} = z_k | \boldsymbol{z}_{\neg ij}^{\ell_1}, \boldsymbol{z}^{\ell_2}, \boldsymbol{w}^{\ell}, \boldsymbol{w}^{\ell_2}, \alpha, \beta, \right) \propto \frac{\Psi_{k,w,\neg ij}^{\ell_1} + \beta}{\Psi_{k,\cdot,\neg ij}^{\ell_1} + V_{\ell}\beta} (\Omega_{i,k,\neg ij} + \alpha).$$

Notice that there are two counters, Ψ^{ℓ_1}, and Ψ^{ℓ_2} used to model the per-word topic distributions. Each hold the counts for the respective language. On the other hand, since the documents are assumed to have a shared topic distribution, there is a single Ω counter variable to model the per-document topic distribution. Further, the model handles the documents written in the different input languages symmetrically and, therefore, its extension to more than two languages is straightforward. For inference using the Gibbs sampling iterative method as well as for deriving topic distributions for unseen documents, the same process with that of LDA can be applied.

2.4 Summary

The chapter provided a short overview of basic tools and topic models that will be extended in the remaining of this manuscript. We have described the distributional hypothesis which is behind very recent models that will be used later in the thesis for several text mining applications. We also introduced and discussed the Multinomial and Dirichlet distributions that are central for several topic models as

prior and posterior distributions. Following that, we provided a brief introduction to some of the most representative Bayesian topic models: we begun with LDA and pLSA before detailing LDA as well as its multilingual extension BiLDA.

Chapter 3

Preliminaries for Neural Networks

Our presentation of topic models like pLSA and LDA as well as their predecessors from the Hyperspace Analogue to Language (HAL) and LSA in Chapter 2 showed that contextual information can be used to generate vectors that successfully model word meaning. We argued that this is achieved because semantically similar words tend to have similar contextual distributions, an idea known as the "distributional hypothesis" first stated in the early 60's [58, 77]. The distributional semantic models of the previous chapter learn word (and document) vectors. Those vectors are obtained by counting how words occur in contexts denoted by documents and applying geometric (LSA) or probabilistic techniques (pLSA, LDA) to the resulting word-document co-occurrence matrices.

Lately, a new family of distributional semantic models have gained popularity. Instead of relying on word-document co-occurrence matrices factorization, they rely on predicting words [124, 126]. The representations learned through those methods are commonly referred to as word embeddings. Embeddings can be efficiently learned using (shallow) neural network architectures. This novel way of training the distributional semantic model and the word representations thereof is attractive because it replaces the transformation steps of the earlier approaches with a prediction step, which is a well-defined supervised learning step. In particular, given a word or a word context, one tries to predict the word context or the word that maximize the performance of the classification task. While the task of trying to predict words or word context may be of low value, the learned embeddings that are a by-product of the prediction task, have been shown to capture interesting semantic and syntactic properties of words. Recall, however, that the idea of learning parameter vectors based on an objective optimum function is also shared by Latent Dirichlet Allocation (LDA) models [23], where the per-word

and per-document topic distribution are parameters learned to optimize the joint probability distribution of words and documents.

There are three main advantages of word embeddings: (i) the dimensions of the learned representations models can be interpreted as general "latent" semantic properties and therefore simple algebraic operations like the addition encode word properties like gender or word relationships like Country-Capital. (ii) The supervised task is defined in such a way that no annotated resources are required: the contexts can be directly extracted from free text. (iii) Some of the models like those that we will present in the rest of the chapter, scale well for huge amounts of data inputs, which is not directly achieved with probabilistic models like LDA. LDA on the other hand, has the advantage of learning representations with some cognitive plausibility as the latent topic are shown to capture the themes of a collection, whereas such interpretations are more difficult for word embeddings.

Word embeddings have shown promising results in a plethora of tasks [15, 146, 124] and constitute another family of models that implement the distributional hypothesis. In the rest of the manuscript we will evaluate the performance of models on top of word embeddings or as a way to incorporate prior knowledge to topic models. Therefore, in this chapter we review popular and high-performing models for training word embeddings also providing a concise introduction to neural networks.

3.1 Word Embeddings with shallow Neural Networks

We review here two models, the continuous bag-of-words model and the skip-gram model [124, 126], both released as part of the word2vec[1] tool, that have shown to perform well across several tasks [15, 158, 140] and have, thus, gained a lot of popularity. The models rely on a shallow neural network with a single hidden layer for learning representations of words and short phrases. Naturally, every feed-forward neural network that takes words from a vocabulary as input learns word embeddings: it embeds the vocabulary identifiers into a vector space of dimension lower than the vocabulary cardinality, and those vectors are then fine-tuned through back-propagation to improve the performance on the task. This first layer is commonly referred to as the "Embedding Layer". Figure 3.1 depicts the embedding layer of a neural network. A word, vectorized using the one-hot-encoding scheme, is associated with a dense vector of size D, where D needs to

[1] https://code.google.com/archive/p/word2vec/

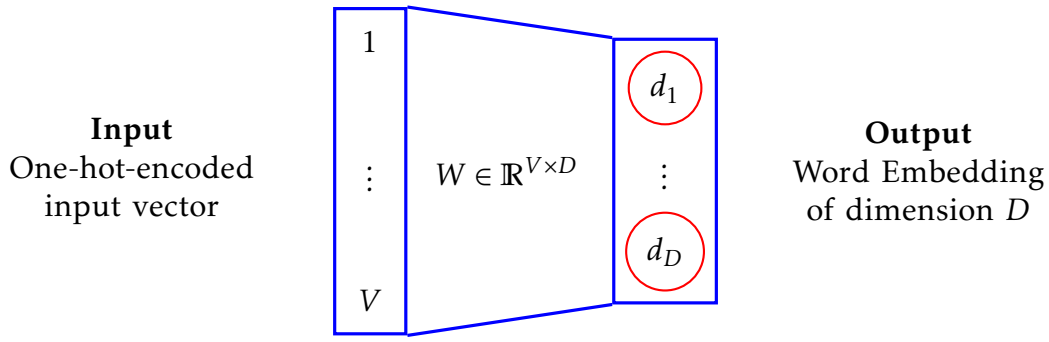


Figure 3.1: Given an one-hot-encoded text input, a dense layer produces a word embedding whose output propagates to the neural network. The matrix W that holds the word embeddings is fine-tuned for the particular task.

be tuned for a particular task. Since the input vector that encodes the word is of dimension V with a single non-zero element, the word's identifier i, the output of the embedding layer is a vector that corresponds to the i-th line of W. Therefore, W is a matrix that holds the embeddings of the V words.

The main difference between such an arbitrary network that learns word embeddings as a by-product of the main task and a method such as word2vec whose explicit goal is to learn the word embeddings is its computational complexity. Generating word embeddings with a deep architecture is simply too computationally expensive for a large vocabulary. On the contrary, the models presented next use shallow networks and are efficient. As an introductory side-note, embeddings learned with tools like word2vec are generally used as initializations of the embedding layers of networks that solve a task wit text inputs, which is similar on how pre-trained networks like VGGNet [161] are used for computer vision architectures: common weight initializations that generally provide useful features without the need for expensive training.

3.1.1 The Skipgram Model

Figure 3.2 illustrates the model used to learn the skipgram word embeddings [124, 126]. The input of the network is a word, while the output of the network is a softmax layer over the words of a corpus. It follows, that the output layer is of dimensionality V. The task used to train the word embeddings is as follows: given a word, maximize the probabilities of the words in its context. The context (also referred to as window) is up to n words before and after the word-occurrence of the input word in the text. In the example of Figure 3.2 we assume an input sequence of words "the cat sat on the mat". For illustration purposes we do not apply any

type of text normalization steps (e.g., stemming). Then, given the word "sits" the network needs to maximize the probabilities of the words of the context (the, cat, on, mat). Notice, that the input layer, which in detail was shown at Figure 3.1, is linear and the only non-linear function is the softmax layer applied in the network output.

In terms of implementation, there are some important details that have been shown to improve the performance of the network if tuned carefully. The window size n is not fixed but dynamic, and it is sampled uniformly from 1 to n, where n is a parameter of the model. The main bottleneck of the model as shown in Figure 3.2, is the softmax layer. Due to the high dimensionality of the output, which is equivalent to V, calculating it analytically is expensive. As an alternative, negative sampling estimates the probability of an output word by learning to distinguish it from draws from a noise distribution. The number of these draws (number of negative samples) is given by a parameter k, usually set to $10 \sim 15$. Negative sampling is very appealing computationally because computing the loss function scales with the number of noise words that we select (k), and not all words in the vocabulary (V), which accelerates training. Furthermore, to limit the effect of very frequent words like stopwords (e.g., and, the, in ...), one can discard them during training with a probability that is proportional to their frequency. Since such words are uninformative, subsampling them results at limiting their effect on the learned embeddings and accelerates training as the prediction step is not performed for every occurrence of them. The subsampling is performed creating the windows to be considered during prediction, which entails that the actual windows used may by larger than n.

The cost function J of the skipgram model as illustrated at Figure 3.2 is:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log p(w_{t+j}|w_t) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log \frac{\exp(\boldsymbol{w}_{t+j}^T \boldsymbol{v}_t)}{\sum_{w=1}^{V} \exp(\boldsymbol{w}_w^T \boldsymbol{v}_t)}, \quad (3.1)$$

where w_t is the embedding of the word t in hidden layer and v_t the output embedding of the term. Notice that the calculation of softmax output of Eq. (3.1) the summation in the denominator over all the words of the vocabulary is a computational bottleneck.

To overcome this problem, [124] suggest another formulation of the problem based on the negative samples. At the same work, another approach based on a hierarchical version is proposed which yields lower results. Negative sampling,

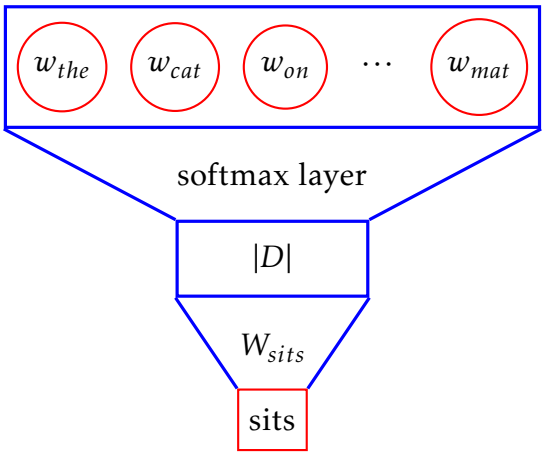


Figure 3.2: The skipgram model.

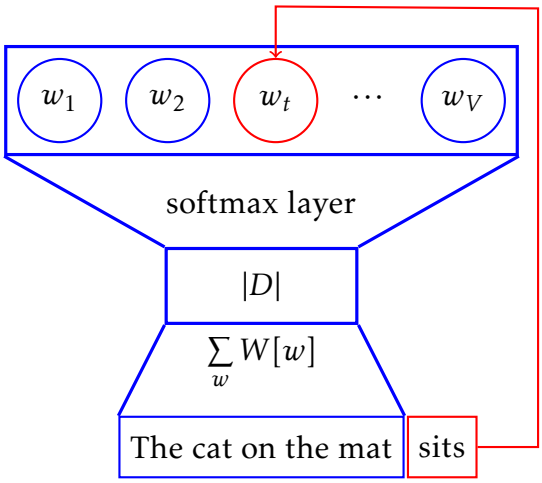


Figure 3.3: The continuous bag-of-words model.

that is an approximation of the Noise Contrastive Estimation (NCE) [75, 130], is a sampling-based approach that tries to approximate the normalization in the denominator of the softmax with some other loss which is cheaper to compute.

Considering a word-context pair, where the context is defined using the window as discussed above, the objective of the skipgram model with negative sampling is as follows. The probability $p(D = 1|w, c)$ stands for the probability the pair comes from the data, while the probability $p(D = 0|w, c) = 1 - p(D = 1|w, c)$ is the probability that the pair does not come from the data but it is rather a negative sample. Further, one has:

$$p(D = 1|w, c) = \sigma(\boldsymbol{w} \cdot \boldsymbol{c}) = \frac{1}{1 + e^{-\boldsymbol{w} \cdot \boldsymbol{c}}},$$

where the model parameters $\boldsymbol{w}, \boldsymbol{c}$ are to be learned. Negative sampling is a special case of NCE that approximates the probability that a word w comes from the empirical training distribution of the training data given a context c with [69]:

$$p(y = 1|w, c) = \frac{e^{\boldsymbol{w_i}, \boldsymbol{c}}}{1 + e^{\boldsymbol{w_i}, \boldsymbol{c}}} = \sigma(\boldsymbol{w_i}, \boldsymbol{c}).$$

The objective is to maximize $p(D = 1|w, c)$ for the observed pairs and also maximize $p(D = 0|w, c)$ for the negative samples:

$$p(D = 0|w, c) = 1 - \frac{1}{1 + e^{-\boldsymbol{w} \cdot \boldsymbol{c}}} = \frac{(1 - 1 + e^{-\boldsymbol{w} \cdot \boldsymbol{c}})e^{\boldsymbol{w} \cdot \boldsymbol{c}}}{(1 + e^{-\boldsymbol{w} \cdot \boldsymbol{c}})e^{\boldsymbol{w} \cdot \boldsymbol{c}}} = \frac{1}{1 + e^{\boldsymbol{w} \cdot \boldsymbol{c}}} = \sigma(-\boldsymbol{w} \cdot \boldsymbol{c}),$$

which results in the objective function of the skipgram model with negative sampling:

$$J_\theta = -\sum_{w \in V} \Big(\log \sigma(\boldsymbol{w} \cdot \boldsymbol{c}) + \sum_{j=1}^{k} \log \sigma(-\boldsymbol{w}_j \cdot \boldsymbol{c}) \Big).$$

While the basis of the skipgram model with negative sampling is the prediction step, it has been shown that in fact this is equivalent to factorization the matrix of the pointwise mutual information (PMI) values of the word-context co-occurrence [108, 109, 146] matrix. This implies that depending on how the context of the co-occurrence matrices is defined (context as document/window..) and how the matrix elements are populated (frequencies, PMI values,..) those models are rather different computational means to arrive at the same type of semantic model. The common factor between them is the underlying distributional hypothesis.

3.1.2 The Continuous Bag-of-Words Model

Another popular model for learning word embeddings is the continuous bag-of-words (cbow) [124, 126]. It is very similar to the skipgram model, and it is illustrated in Figure 3.3. The cbow model learns to predict the word in the middle of a symmetric window based on the sum of the vector representations of the words in the window. The words of the window are summed and, as a result their order is not taken into account explicitly. Therefore, the model assumes a bag-of-words representation, that is highlighted in its name.

The example of Figure 3.3 is analogous to that of Figure 3.2. The input sequence of words is assumed to be "the cat sits on the mat". Given the words that surround "sits", the model's goal is then to maximize the probability of that word given as input the sum of the representations of the words "the, cat, on, the, mat".

The fact that the cbow model performs an averaging operation statistically signifies that it smoothes over a lot of the distributional information because it treats an entire context as one observation. This becomes more intense as the window size increases. In the original paper [124] the authors note that this turns out to be a useful thing for smaller datasets. On the other hand, skipgram treats each context-target pair as a new observation, and this tends to do better when we have larger datasets.

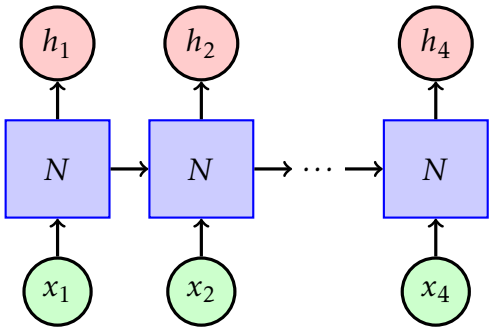


Figure 3.4: An unrolled Recurrent Neural Network. N maybe an arbitrarily deep neural network, that given an input x_1, which can be a word embeddings for instance, propagates a hidden state to its successor. The RNN can be very long [157].

3.2 Text Representations using deep neural networks

The skipgram and cbow models of the previous section learn word embeddings using a prediction task that relies on how words co-occur with their contexts in free text. When more information about a task is available like labeled examples, such general purpose embeddings could be used to initialize the weights of neural networks. Then, allowing the backpropagation gradients to modify the embeddings would result in fine-tuning them for the given task. A shortcoming of the traditional feed-forward neural networks is that they do not model sequences effectively. On the other hand, as text is a sequence of words, being able to capture the dependencies that this suggests may be advantageous for several tasks.

Recurrent neural networks (RNNs) are a family of networks that address this shortcoming. They are deep neural networks: once unfolded they resemble to traditional neural networks with several hidden layers. RNNs contain loops that allow information from step t to be passed to step $t+1$. They can be seen as copies of the same network, each passing a message to its successor. We illustrate at Figure 3.4 an unrolled RNN whose inputs are the embeddings of T words. In the figure N is a neural network, typically consisting of a sigmoid hidden layer, although deeper architectures can be considered.

RNNs are appealing as they are supposed to model information until input t and pass it to its successor $t+1$. In practice, while RNNs may be able to achieve that when the sequence of inputs is small, when the sequence becomes larger than a few elements they fail [19, 81, 80] due to the problem of vanishing gradients, which make it difficult to train the network efficiently as the length of the input sequence increases.

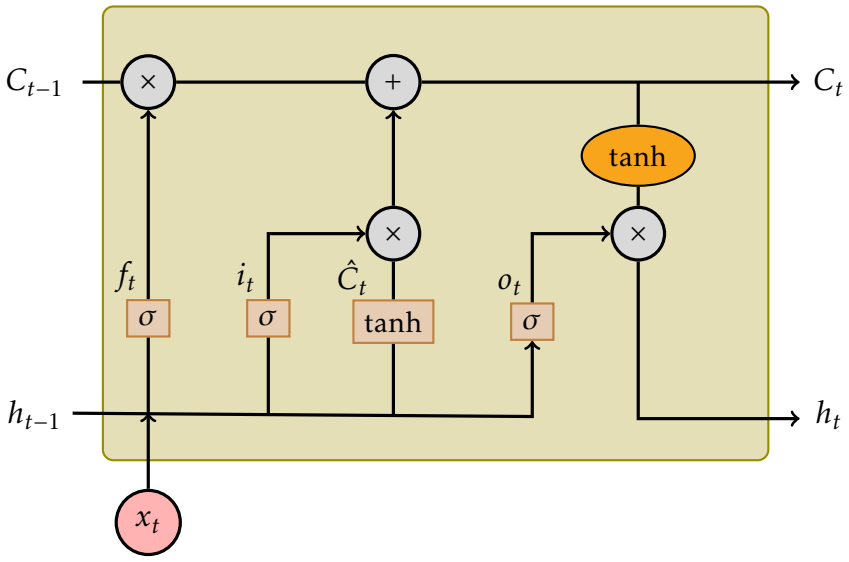


Figure 3.5: Illustration of the architecture of a long short-term memory unit. The inputs of the LSTM unit are the cell state (C_{t-1}), the output of the previous unit (h_{t-1}) and the input of the sequence that is currently modeled x_t. Rectangles denote hidden layers while the algebraic operators and the *tanh* function in the ellipse are applied element-wise to the vectors that are inputs to the blocks.

The long short-term memory network [81] (LSTM) is a popular and state-of-the-art network that alleviates the limitations of the vanilla RNN networks described above. To manage to encode information from long periods of time, the layers N of LSTM are more complex than those of RNNs. Instead of a single hidden neural network they have four that interact in a particular way in order to encode information from previous states efficiently. Figure 3.5 shows the internals of an LSTM unit.

We provide a brief description of the LSTM unit describing the purpose of the four hidden layers shown at Figure 3.5. A central idea for LSTMs is the cell state, illustrated in the Figure as the top continuous line $[C_{t-1}, C_t]$. The cell is an information flow whose content is updated via one element-wise multiplication and one element-wise addition operation. The rest of the hidden layers, known as gates, control how the information of the cell is updated.

The left-most bottom sigmoid layer is the "forget gate". Given the concatenation of h_{t-1} and the current input x_t, it outputs a vector of values within $[0,1]$ that are weights quantifying how much of the C_{t-1} will be kept. Given the notation of the figure we have:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3.2}$$

where the square brackets "[]" applied to two or more vectors e.g., $[h_{t-1}, x_t]$ denote the concatenation of their values.

While the forget gate decides which information will be removed from the cell state, the input gate decides in a two-step process which information will be added in the cell state. First, a sigmoid layer, whose output is denoted by i_t, decides which values will be updated. Subsequently, a *tanh* layer (whose output is $\hat{C}_t$) suggests and candidate values for the state. The candidate values $\hat{C}_t$ are combined with element-wise multiplication with i_t and are added to the cell state. Therefore, the calculation of the information to be added in the cell state and the updates of its values are given by:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \hat{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \\ C_t &= f_t \times C_{t-1} + i_t \times \hat{C}_t. \end{aligned} \tag{3.3}$$

The last gate calculates the output of the current LSTM unit, which is a filtered version of the updated cell state C_t. A sigmoid function is therefore applied to the concatenation of h_{t-1} and x_t and the output is multiplied (element-wise) with the squashed values of C_t that are obtained by applying *tanh* on its elements. Therefore:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \times \tanh(c_t). \end{aligned} \tag{3.4}$$

The equations (3.2),(3.3), (3.4) describe the updates performed in each of the gates of the LSTM. The values of the matrices W_f, W_i, W_c and W_o as well as the corresponding biases are updated gradient descent and the back-propagation through time algorithm [81].

3.3 Cross-lingual Word Embeddings

In the previous sections of the chapter we restricted the presentation of models for learning word embeddings of modeling text sequences in a single language. As we discussed in Chapters 1 and 2 however, the amount of multilingual content online steadily increases. In conjunction with the fact that resources, training data, and benchmarks are mostly for English language, which may result in a disproportionate focus or even a bias (complementary to the gender/racer bias observed by [26]) for this language.

To overcome such issues, cross-lingual embeddings aim at learning embeddings for the words of two or more languages that share the same space. The hope is, then, that by projecting examples from a language in this space and training a model, the model will have the ability of predictions for the rest of the languages also. This of course assumes remedies to several problems, like having effective compositional models to that capture the semantics of text spans larger than words.

Recently, several models have been proposed for learning cross-lingual word embeddings. Among them, several approaches build on the successful models proposed for monolingual collections and extend models like the skipgram model with negative sampling in the bilingual or multilingual space [125, 118, 71, 40]. The models for learning cross-lingual embeddings can be grouped with regard to the type of approach used to align the multilingual embeddings [154]:

- *Monolingual mapping*: this family of methods begin by learning monolingual word embeddings and try to learn a linear transformation from one space to another. It was first proposed by [125] and followed by more works [190, 106] trying to relax some of the underlying assumptions.

- *Pseudo-cross-lingual*: The methods of this family aim at generating pseudo-cross-lingual datasets where dictionaries are used to replace words with their translations in order to obtain artificial contexts and then train models like skipgram [72, 50] or generate multilingual documents are concatenated and their content is shuffled [179].

- *Cross-lingual training*: the approaches of this family optimize a cross-lingual training loss using parallel, sentence aligned corpora. For instance, use autoencoders to encode sentences or documents in a source language and reconstruct it in another language [104, 9].

- *Joint optimization of monolingual and cross-lingual losses*: the approaches of this family optimize both a cross-lingual and monolingual losses. For instance, [118, 71] extend the skipgram model to the bilingual case: the former uses the words in the source language to additionally predict their aligned words in the target language while the later introduces an L_2 sampled loss for cross-lingual regularization.

3.3.1 Bilbowa

As discussed above, the goal of models used to learn cross-lingual embeddings is to learn feature that generalize across languages. The goal is to learn word embeddings such that similar words in each language are close in the induced space and furthermore similar words across languages are also close. For example, the words "cat", "dog" and "chien" (French for dog) are expected to be close in the shared space. Assigning similar embeddings stands for assigning vectors that are close in terms of a distance metric like Euclidean distance or cosine similarity in the induced space. To accomplish this, they optimize a monolingual loss that encourages similar words to have similar embeddings in each languages and a cross-lingual loss that encourages similar words written in different languages to be close.

To achieve that Bilbowa [71] that stands for "Bilingual Bag-of-Words without Alignments" optimizes a monolingual objective function $\mathcal{L}(\cdot)$ and the cross-lingual objective is enforced as regularization by a term Ω. Therefore the overall loss function to be optimized is:

$$\mathcal{L} = \min \sum_{\ell \in \{s,t\}} \sum_{w_t, h \in D^\ell} \mathcal{L}^\ell(w_t, h; \theta) + \lambda \Omega(\theta^s, \theta^t). \tag{3.5}$$

The first term of Eq. (3.5) captures the monolingual objective over the *source s* and *target t* languages while the second term encourages the embeddings of similar words across languages to be close. One of the advantages of this formulation is that one may use unlimited corpora for learning the monolingual embeddings and a smaller collection of parallel sentences to enforce the regularization.

For the monolingual objective Bilbowa uses the objective of the skipgram model that we presented in Section 3.1.1. For the cross-lingual objective the model utilizes sentence aligned data and approximates with a sampling method the loss:

$$\Omega = \sum_i \sum_j a_{i,j} \parallel \boldsymbol{r}_i^s - \boldsymbol{r}_j^t \parallel^2, \tag{3.6}$$

where $a_{i,j}$ encodes a translation score for the words i and j approximated using the sentence aligned parallel data and $\boldsymbol{r}_i^s$, $\boldsymbol{r}_j^t$ are the embeddings of the words i, j in the source and target languages respectively. Intuitively, Eq. (3.6) is weighted sum that is minimized when pairs of words with high translation scores $a_{i,j}$ have small Euclidean distances in the embedding space. Instead of relying on an alignment tool like Giza++ [139] which is computationally expensive for finding the $a_{i,j}$ from the aligned sentences, the model uses a sampling mechanism that accelerates the process.

3.3.2 Concept Net

Word embeddings have been shown to capture the semantics of words as well as some of their syntactic properties [124]. The models we presented so far however, from the topic models approaches like LDA to the word emebedding models like skipgram implement distributional semantics models using strictly free text. Their cross-lingual extensions also rely on free text for each language and may also require some aligned comparable or parallel corpora. One would expect however, that combining those successful models with external information sources like knowledge graphs would yield even better representations. The work of [166] proposed to extend the models for learning word representations to incorporate knowledge from ConceptNet, that is linked open data resource.

ConceptNet, first released by [112], is a knowledge graph that connects words and phrases with labeled, weighted edges. The edges encode relations of "is_a" type (*e.g.* The word *cold* in English is *studený* in Czech). Its graph-structured knowledge is particularly useful for models like those used to train word embeddings as such relations can be utilized to learn semantic spaces that are more effective than using distributional semantics alone, as in the case of the previously presented models.

The fundamental idea behind the model used to incorporate the knowledge graph in the process of learning word embeddings is to fine-tune embeddings learned with a model like skipgram using the knowledge graph relationships. Assuming, for instance, the objective function:

$$\mathcal{L}(Q) = \sum_{i=1}^{n} \Big[a_i \parallel q_i - \hat{q_i} \parallel^2 + \sum_{(i,j)\in E} \beta_{i,j} \parallel q_i - \hat{q_i} \parallel^2 \Big]. \tag{3.7}$$

The loss of Eq. (3.7) describes the fine-tuning process used to derive the embedding of word i to be $\hat{q_i}$ from an initially learned embedding q_i. This process is referred to *expended retrofitting* [166, 165]. The first term encourages the updated embeddings to be close to the original ones by minimizing their Euclidean distance $\parallel q_i - \hat{q_i} \parallel^2$. The second term, which is a weighted summation over the edges of the knowledge graph encourages the embeddings of words that have some relationships to be close. The weights $\beta_{i,j}$ of the sum are also taken from the knowledge graph. This makes it possible to learn embeddings for words that were out-of-vocabulary of the initially learned embeddings q_i, by effectively setting $\alpha_i = 0$ for a word i and relying only on the knowledge graph connections. Another significant advantage of the expanded retrofitting as described above is that it can benefit

from the multilingual connections in ConceptNet (*e.g.* The word *cold* in English is *studený* in Czech). The model learns more about each language via the translations of words in other languages, and also aligns the embeddings of similar semantically similar terms written in different languages.

3.4 Summary

The chapter presented an overview of word embeddings. Their interesting properties of capturing semantic and syntactic properties of the words were highlighted as they will be later used as external sources of information for probabilistic models or as text representations for different tasks. We also presented popular models that are based on shallow neural networks for learning embeddings for text written in one (cbow or skipgram) or more languages (BilBowa, ConceptNet). Apart from shallow neural networks, we also presented recurrent neural networks, which although computationally expensive are well suited for modeling text sequences. For a deeper analysis of deep learning models and their properties, one may refer to the introduction of [138] or the excellent reviews of [157, 70].

Chapter 4

Incorporating Prior Knowledge of Text Structure to Topic Models

Probabilistic topic models aim at uncovering the latent topics of a collection of documents. To identify the topics, one typically represents the documents of the collection as a bag-of-words and applies an inference approach like Gibbs Sampling [73]. We discussed in Chapter 2 that popular topic models (*e.g.*, LDA [23]) represent documents as bag-of-words. We argue in this chapter that this can be limiting and propose to overcome it.

The exchangeability assumption, that follows from the bag-of-words, dictates that given the topic distribution of a document, the words of the document are conditionally independent. While this assumption greatly benefits the involved computations and, in particular, the calculations of the conditional probabilities, it is rather naive and unrealistic [78]. A shortcoming concerns the loss of information from not accounting for the grouping of words in topically coherent spans. These can be contiguous words that form text spans like sentences that are important in the use of language.

Text structure contains useful information that could be leveraged during inference. Sentences or phrases, for instance, are by definition text spans complete in themselves that convey a concise statement. To better illustrate how text structure could help in topic identification, consider the example of Figure 4.1. It shows the topics inferred by LDA for the words (excluding stop-words) of a sentence drawn from a Wikipedia page. At the sentence level, one could argue that the sentence is generated by the topic "Cinema" since it discusses a film and its authors. LDA, however, fails and assigns several topics to the words of the sentence. Importantly, several of those topics like "Elections" and "Inventions" are unrelated. In finer textual granularity, LDA also fails to assign consistent topics in noun-phrases like

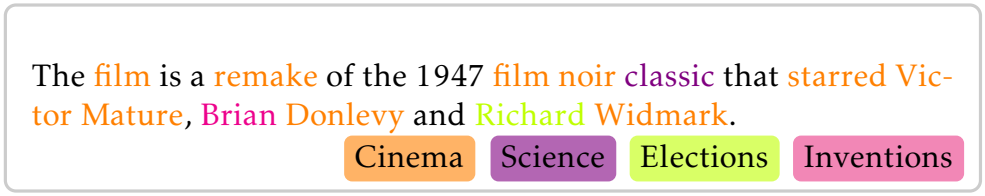


Figure 4.1: Applying LDA on Wikipedia documents. Notice how LDA assigns several, unrelated topics to the excerpt.

"film noir classic" and entities like "Brian Donlevy". A binding mechanism among the topics of the words of a sentence, or a phrase, could have prevented those inconsistencies in the topic assignment process. Hence, the hypothesis we investigate in this chapter is whether taking simple text structure into account benefits topic models.

To evaluate our hypothesis, we aim at extending LDA with prior knowledge of text structure. We suggest that such knowledge can be in the form of boundaries of topically coherent text spans, like the noun-phrases of Figure 4.1. We propose two approaches to achieve that:

(i.) The first assumes that the words within a coherent segment are generated by the same topic and proposes a collapsed Gibbs sampling inference process that uncovers the topic while taking into account the per-word topic distributions of the words that compose the segment [11].

(ii.) The second shares the motivation that segments should be topically coherent but proposes a more flexible approach that utilizes copulas in the sampling process and, therefore, allows a few (instead of one), related topics to occur within the segment [12].

Both models assume some level of dependence between the topics of the words of segments. This dependence is maximal for the first model. The second, through the use of copulas has bigger modeling capacity and is more flexible. It is to be noted, that for both types of models, the documents are assumed to be segmented *a priori*. Different segmentation mechanisms can be used, from linguistically motivated (e.g., parsing) to statistically motivated (e.g., n-grams). We expect different approaches to text segmentation to have different advantages; we intend to evaluate the impact of this choice also.

The remainder of the chapter discusses those points in detail. It is organized as follows:

- Section 4.1 presents an overview of the related work.
- Section 4.2 defines what segments are, as the concept is fundamental for the topic models to be proposed.
- Section 4.3 describes the two novel topic models: *segmentLDA* and *copulaLDA* and details their inference processes.
- Section 4.4 presents the experiments performed to assess the quality of the proposed topic models, and
- Section 4.5 concludes with a summary of the chapter.

4.1 An overview of the relevant literature

Despite the success that vector-space models [156] have enjoyed, they come with a number of limitations. We mention, for instance, their inability to model synonymy and polysemy and the sparse, high-dimensional induced representations. Many research studies have pointed out these problems, and Probabilistic Latent Semantic Analysis [84] was among the first attempts to model textual corpora using latent topics. In this chapter, we build on LDA [23], which is often used as a building block for topic models. In its context, the corpus is associated with a set of latent topics, and each document is associated with a random mixture of those topics. The words are assumed exchangeable, that is their joint probability is invariant to their permutation. Previous work proposed a variety of extensions to LDA in order to incorporate additional information such as class labels [24] and temporal dependencies between stream documents [184]. Here, our goal is to extend LDA by incorporating simple text structure in its generative and inference processes using copulas.

One may identify two lines of research to address the limitations due to the exchangeability assumption in LDA: extensions to account for the boundaries of text spans like sentences and extensions to account for the word order. With respect to the first line, [181] combines a unigram language model with topic models over sentences so that the latent topics are represented by sentences instead of terms. In [74], the authors investigate a combination of a topic model with a Hidden Markov Model (HMM). They assume that the HMM generates the words that handle the long-range dependencies (semantic dependencies) and the topic model the words

that handle the short range dependencies (syntactic dependencies). Also, [29] proposed the Syntactic Topic Model whose goal is to integrate the text semantics and the syntax in a non-parametric topic model. Having the tree, the semantic consistency of each document is given by a distribution over latent topics, as in topic models, and the syntactic consistency by the fact that each element in the tree has also a distribution over the topics of its children. In another effort, [197] propose *TagLDA*, where they replace the unigram word distributions by a factored representation that is conditioned on the topic and the part-of-speech tag of a term.

The second line of research investigates how topic models can be extended to incorporate word order. In [159], the authors propose a four-level hierarchical structure where the latent topics of paragraphs are decided after performing a nested word-based LDA operation. A particularly interest body of work concerns collocations. They can be defined as a sequence of consecutive words that have the characteristic of a syntactic and semantic unit, such as *stock market*, *Los Angeles Premier League* [35]. Previous work has mainly explored the idea of bigram collocations. For instance, [182] studied when bigrams should be assigned as a whole to particular topics or as two disjoint unigrams in other topics. Later, [89] proposed a model that combines the LDA and adaptor grammars to incorporate word collocations in the process of topic modeling. Despite their theoretical elegance these models come with higher computational overhead. To this end, [103] explored how various strategies of selecting bigrams to be used as artificial tokens can impact the quality of topic models. Their models although interesting and inspiring for our development, have the shortcoming of increasing the vocabulary size and therefore the sparsity of the model.

Another interesting line of research studied the task of discovering and partitioning text in topically coherent spans. In [47, 48] the authors rely on hierarchical Bayesian models to accomplish it. In our work here, contrary to identifying such spans, we assume documents to be topically coherent *a priori*, and we investigate how to leverage and incorporate this information to LDA.

4.2 The coherent text segments

In this section we discuss the idea of *coherent text spans*, or *segments* hereafter, as they are central for the subsequent development of the topic models. Given a document, there are several types of text spans that can be regarded as the document's

The film is a remake of *the 1947 film noir classic* that starred *Victor Mature*, *Brian Donlevy* and *Richard Widmark*.

Figure 4.2: Shallow parsing of a sentence with the Stanford Parser. Contiguous words in italics like "Richard Widmark" denote a noun-phrase.

building blocks. These spans usually constitute semantic units pertaining to a single or few related topics and are topically coherent in this sense [12]. From a linguistic point-of-view, a document consists of sentences, which are meaningful text spans that convey a concise statement. In a finer level, syntactic analysis of sentences like shallow parsing reveals coherent segments like noun-phrases. Figure 4.2 illustrates the output of a shallow parsing step that identifies the noun-phrases of the example sentence of Figure 4.1, generated using the Stanford Parser.[1]

Both sentences and noun-phrases are text spans composed by contiguous words and can be considered semantically coherent. Further, from a computational aspect, sentence segmentation can be performed efficiently in several languages. Also, there are several pre-trained models for shallow parsing in a variety of languages. For the above reasons, in the rest of our development we will evaluate how assuming sentences and noun-phrases to be topically coherent affects the topic modeling performance.

N-grams are consecutive words of length N. Compared to sentences or noun-phrases no syntactic information is used to obtain them as they only rely on count statistics. Despite their simplicity, their positive effects on topic modeling have been shown in previous work (e.g., [103, 182]). Therefore, n-grams are another type of segments we may consider coherent.

The segmentation of the documents in coherent segments creates a hierarchical document representation. From the higher to the lower levels this structure is described as follows: (i) documents consist of segments, which are independent between them, and (ii) segments consist of words, which are their basic units. Therefore, each document is a *bag-of-segments* and each segment is a *bag-of-words*. These independence assumptions are important while performing inference for the topic models that will presented in the next sections.

[1] `http://nlp.stanford.edu/software/lex-parser.shtml`

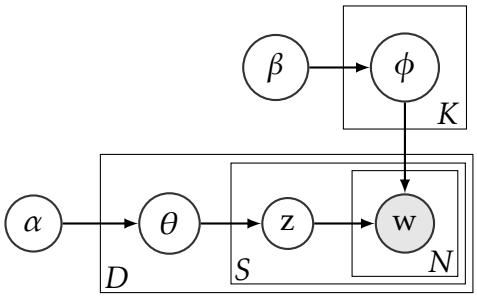


Figure 4.3: The graphical model of *segLDA*.

4.3 Incorporating text structure to topic models

The previous sections provided an overview of the relevant literature with a focus on work that incorporates parts of text structure to topic models and defined coherent text segments. We continue by presenting the two contributions of this chapter. In particular, we first describe *segmentLDA* in Section 4.3.1 and then introduce in Section 4.3.3 *copulaLDA*. These are two novel topic models that relax the bag-of-words assumption that is a fundamental premise of LDA. They propose different mechanisms for incorporating text structure in their generative process.

4.3.1 segmentLDA: Integrating segment boundaries to LDA

Recall that a probabilistic topic model represents the words in a collection of D documents as mixtures of K "topics", which are multinomials over a vocabulary of size V. In the case of LDA, for each document d_i a multinomial over topics is sampled from a Dirichlet prior with parameters $\boldsymbol{\alpha}$.

We extend LDA by adding an extra plate denoting the coherent text segments of a document. The graphical representation of this novel mode, called *segmentLDA* (segLDA) model is shown in Figure 4.3. The generative process of a document collection according to segLDA is as follows:

- For each topic $k \in [1, K]$, choose a per-word distribution: $\boldsymbol{\phi_k} \sim Dir(\boldsymbol{\beta})$, with $\boldsymbol{\phi}_k, \boldsymbol{\beta} \in \mathbb{R}^V$
- For each document d_i, $i \in \{1, \ldots, D\}$:
  - Choose a per-document topic distribution: $\boldsymbol{\theta_i} \sim Dir(\boldsymbol{\alpha})$, with $\boldsymbol{\theta_i}, \boldsymbol{\alpha} \in \mathbb{R}^K$
  - For each segment $s_{i,j}, j \in \{1, \ldots, S_i\}$ of d_i:
    - Sample the topic underlying the segment's words: $z_{i,j} \sim Mult(1, \boldsymbol{\theta_i})$

$\star$ Sample the words of the segment: $(w_1, \ldots, w_{N_{i,j}}) \sim Mult(N_{i,j}, \boldsymbol{\phi}_{z_{i,j}})$

As it is evident from the generative story, a single topic is assigned to the words of a segment as each word of the segment is sampled from $Mult(N_{i,j}, \phi_{z_{i,j}})$. However, a topic is assigned to each and every word of the document, as in LDA. Therefore, words (and not segments) remain the basic units of the documents. This signifies that comparing topics models such as LDA and segLDA on tasks that use the topics of each word (like perplexity that we discuss in Section 4.4) is fair.

The generative process we presented above describe the generation of the collection of documents. Meanwhile, given a corpus words are observed and the goal is to infer the latent topics. For inference, we propose to use a collapsed Gibbs sampling method [73]. We now derive the Gibbs sampler equations by estimating the hidden topic variables. In segLDA the joint distribution of words $\boldsymbol{w}$ and topics $\boldsymbol{z}$ can be decomposed as:

$$p(\boldsymbol{w}, \boldsymbol{z}|\alpha, \beta) = p(\boldsymbol{w}|\boldsymbol{z}, \beta)p(\boldsymbol{z}|\alpha), \tag{4.1}$$

because the first term is independent of α[2] and the second from β (cf. Fig. 4.3). After standard manipulations as in the paradigm of [78] one arrives at:

$$p(\boldsymbol{z}, \boldsymbol{w}|\alpha, \beta) = \prod_{z=1}^{K} \frac{\Delta(\Psi_z + \beta)}{\Delta(\beta)} \prod_{i=1}^{D} \frac{\Delta(\Omega_i + \alpha)}{\Delta(\alpha)}, \tag{4.2}$$

where $\Delta(\vec{x}) = Beta(x_1, \ldots, x_m) = \frac{\prod_{k=1}^{dim\vec{x}} \Gamma(x_k)}{\Gamma(\sum_{k=1}^{dim\vec{x}} x_k)}$ is a multidimensional extension of the beta function used for notation convenience, and Ω_i, Ψ_z refer to the occurrences of topics with documents and topics with terms respectively. To calculate the full conditional probability we take into account the structure of the document i and the fact that $\boldsymbol{w_i} = \{\boldsymbol{w}_{i\neg s_{i,j}}, \boldsymbol{w}_{\neg s_{i,j}}\}$, $\boldsymbol{z} = \{\boldsymbol{z}_{i\neg s_{i,j}}, \boldsymbol{z}_{\neg s_{i,j}}\}$. The subscript $s_{i,j}$ in $\boldsymbol{w}_{s_{i,j}}, \boldsymbol{z}_{s_{i,j}}$ denotes the words and the topics respectively of segment $s_{i,j}$, that is the j-th segment of the i-th document. For the full conditional of topic k we have:

$$\begin{aligned} p(z_{s_{i,j}} = k|\boldsymbol{z}_{\neg s_{i,j}}, \boldsymbol{w}) =& \frac{p(\boldsymbol{w}, \boldsymbol{z})}{p(\boldsymbol{w}, \boldsymbol{z}_{\neg s_{i,j}})} = \frac{p(\boldsymbol{w}|\boldsymbol{z})}{p(\boldsymbol{w}_{\neg s_{i,j}}|\boldsymbol{z}_{\neg s_{i,j}})p(w_{s_{i,j}})} \frac{p(\boldsymbol{z})}{p(\boldsymbol{z}_{\neg s_{i,j}})} = \\ =& \frac{p(\boldsymbol{w}, \boldsymbol{z})}{p(\boldsymbol{w}_{\neg s_{i,j}}, \boldsymbol{z}_{\neg s_{i,j}})} \propto \frac{\Delta(\Psi_z + \beta)}{\Delta(\Psi_{z,\neg s_{i,j}} + \beta)} \frac{\Delta(\Omega_i + \alpha)}{\Delta(\Omega_{i,\neg s_{i,j}} + \alpha)}. \end{aligned} \tag{4.3}$$

[2]Hereafter, we consider $\boldsymbol{\alpha}, \boldsymbol{\beta}$ to be symmetric, that is $\alpha_1 = \ldots = \alpha_K, \beta_1 = \ldots = \beta_V$, and we denote by the scalars α, β the values of each dimension of the vector.

For the first term of equation Eq. (4.3) we have:

$$\frac{\Delta(\Psi_z+\beta)}{\Delta(\Psi_{z,\neg s_{i,j}}+\beta)} = \frac{\frac{\prod_{w\in s_{i,j}}\Gamma(\Psi_z+\beta)}{\Gamma(\sum_{w\in s_{i,j}}(\Psi_z+\beta))}}{\frac{\prod_{w\in s_{i,j}}\Gamma(\Psi_{z,\neg s_{i,j}}+\beta)}{\Gamma(\sum_{w\in s_{i,j}}(\Psi_{z,\neg s_{i,j}}+\beta))}} = \prod_{w\in s_{i,j}}\left(\frac{\Gamma(\Psi_z+\beta)}{\Gamma(\Psi_{z,\neg s_{i,j}}+\beta)}\right)\frac{\Gamma(\sum_{w\in s_{i,j}}(\Psi_{z,\neg s_{i,j}}+\beta))}{\Gamma(\sum_{w\in s_{i,j}}(\Psi_z+\beta))} =$$

$$= \underbrace{\frac{\overbrace{\prod_{w\in s_{i,j}}(\Psi_{w,k,\neg s_{i,j}}+\beta)\cdots(\Psi_{w,k,\neg s_{i,j}}+\beta+(N_{i,j,w}-1))}^{\text{A}}}{(\sum_{w\in V}(\Psi_{w,k,\neg s_{i,j}}+\beta))\cdots(\sum_{w\in V}\Psi_{w,k,\neg s_{i,j}}+\beta+(N_{i,j}-1))}}_{\text{B}}. \tag{4.4}$$

Here, for the generation of A and B we used the recursive property of the Γ function: $\Gamma(x+m) = (x+m-1)(x+m-2)\cdots(x+1)x\Gamma(x)$; w is a term that can occur many times in a sentence and $N_{i,j,w}$ denotes the frequency of w in segment $s_{i,j}$; $N_{i,j}$ denotes the number of words in sentence s.

The development of the second factor in the final step of Eq. (4.3) is similar to the LDA calculations. The difference is that the counts of topics per document are estimated given the allocation of every word of a segment to the sampled topic. On the other hand, compared to segLDA, LDA does not incorporate any part of the local structure when sampling topics for the words. From Eq. (4.3) one yields:

$$p(z_{s_{i,j}}=k|\vec{z}_{\neg s_{i,j}},\vec{w}) = (\Omega_{i,k,\neg s_{i,j}}+\alpha)\times\frac{\prod_{w\in s_{i,j}}(\Psi_{w,k,\neg s_{i,j}}+\beta)\cdots(\Psi_{w,k,\neg s_{i,j}}+\beta+(N_{i,j,w}-1))}{(\Psi_{\cdot,k,\neg s_{i,j}}+V\cdot\beta)\cdots(\Psi_{\cdot,k,\neg s_{i,j}}+V\cdot\beta+(N_{i,j}-1))} \tag{4.5}$$

where $\Omega_{i,k,\neg s}$ denotes the number of words from document i assigned to topic k excluding the words of the segment currently sampled. Further, the product in the numerator of the second term results from the bag-of-words assumption for the words within the segments of d_i. The possibly multiple occurrences of w in $s_{i,j}$, generated by the topic k, are taken into account by the factor $(\Psi_{w,k,\neg s_{ij}}+\beta)$, which is incremented by one for every other occurrence of the word after the first. This reflects the fact that every occurrence of w comes from the same topic. For instance, if w appears twice in $s_{i,j}$, then $N_{i,j,w}=2$, and the factor $(\Psi_{k,w,\neg s_{ij}}+\beta)(\Psi_{k,w,\neg s_{ij}}+\beta+1)$ denotes the contribution of the occurrences of w to the probability that $s_{i,j}$ is generated by the topic k. The product in the denominator acts as a normalization term. The progressive increase of its values can be explained by the bag-of-words assumption within a segment: the product normalizes the probability of assigning

Algorithm 1: A Gibbs Sampling iteration for segLDA

```
Input: documents' words grouped in segments, α, β, K
//Initialize counters Ψ,Ω
for document d_i, i ∈ [1,D] do
    for segment s_{i,j} : j ∈ {1,...,S_i} do
        Decrease counter variables Ψ,Ω according to the previous topic
          assignments of the words of s_{i,j}
        Calculate the probabilities of the new topic of the words of s_{i,j} (Eq. 4.5)
        Sample the topics of the words of s_{i,j} using the calculated probabilities
        Increase counters Ψ,Ω
    end
end
```

the topic k to a word of the segment, given that the previous words have also been assigned to this topic. Algorithm 1 presents the steps one needs to follows during the Gibbs sampling updates of segLDA.

Note that segLDA is an extension of LDA. If the coherent text spans are reduced to words that is $\forall i, j : N_{i,j} = N_{i,j,w} = 1$ then segLDA reduces to LDA and Eq. (4.5) reduces to the standard LDA collapsed Gibbs sampling inference equations of Eq. (2.5).

4.3.2 Copulas and random variables (intermezzo)

In the previous section we proposed segLDA, a novel topic model that assigns the same topics to the words of a segment. This entails that the dependency between the topics of the words of a segment is maximal. The question that arises in this case is whether one can come up with a more flexible binding mechanism. If such a mechanism exists, one could then incorporate to a topic model that would account for text structure but would also allow more flexibility within topics. In this section, we introduce copulas, a statistical tool that can be used to solve this problem.

Copulas allow one to explicitly relate joint and marginal distributions, through Sklar's theorem [162]:

Theorem 4.3.1. *Let F be a p-dimensional distribution function with univariate margins $F_1,\ldots,F_p$. Let A_j denote the range of F_j. Then there exists a copula C such that for all $(x_1,\ldots,x_p) \in \mathbb{R}^p$*

$$F(x_1,\ldots,x_p) = C(F_1(x_1),\ldots,F_d(x_p)) \tag{4.6}$$

Furthermore, when $F_1,\ldots,F_p$ are all continuous, then C is unique.

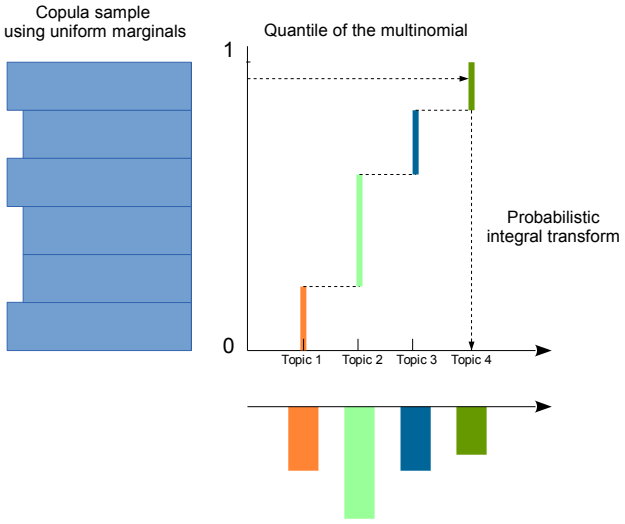


Figure 4.4: The transformation of a random variate to multinomial (or arbitrary) marginals. The arrows illustrate the generalized inverse; the histograms in y (resp. x) axis depict the distributions of the initial (resp. transformed) samples.

Formally [134, 173], a p-dimensional copula C is a p-variate distribution function with $C : \mathbb{I}^p = [0,1]^p \rightarrow [0,1]$ whose univariate marginals are uniformly distributed on $\mathbb{I}$ and $C(u_1,\ldots,u_p) = P(U_1 \leq u_1,\ldots,U_p \leq u_p)$.

As a result any multivariate distribution F can be decomposed into its marginals F_i, $i \in \{1,\ldots,p\}$ and a copula, allowing to study the multivariate distribution independently of the marginals. Sklar's theorem also provides a way of sampling multivariate distributions with a large number of random variables using copulas: $F(x_1,\ldots,x_p) = F\left(F_1^{-1}(u_1),\ldots,F_p^{-1}(u_p)\right) = P[U_1 \leq u_1,\ldots,U_p \leq u_p] = C(u_1,\ldots,u_p)$. Hence, to sample F it suffices to sample the dependence structure modeled by copulas and then transform the obtained sample in the marginals of interest using the probabilistic integral transform. We illustrate this transformation for one variable in Figure 4.4. Sampling the copula returns, for each variate, a sample as the one indicated in the histogram of the y axis. One can then transform the sample using the quantile (F^{-1}) of an arbitrary marginal.

Before proceeding further, we visit some extreme conditions of dependence illustrating the respective copulas that model them: (1) *Independence*, which is a frequently assumed simplification in topic models and is obtained with $\prod_{i=1}^{p} u_i$, and (2) *Co-monotonicity*, which is the complete, positive correlation between the random variables u_p, obtained with $\min(u_1,\ldots,u_p)$.

In the rest of our development we will be using a particular family of copulas,

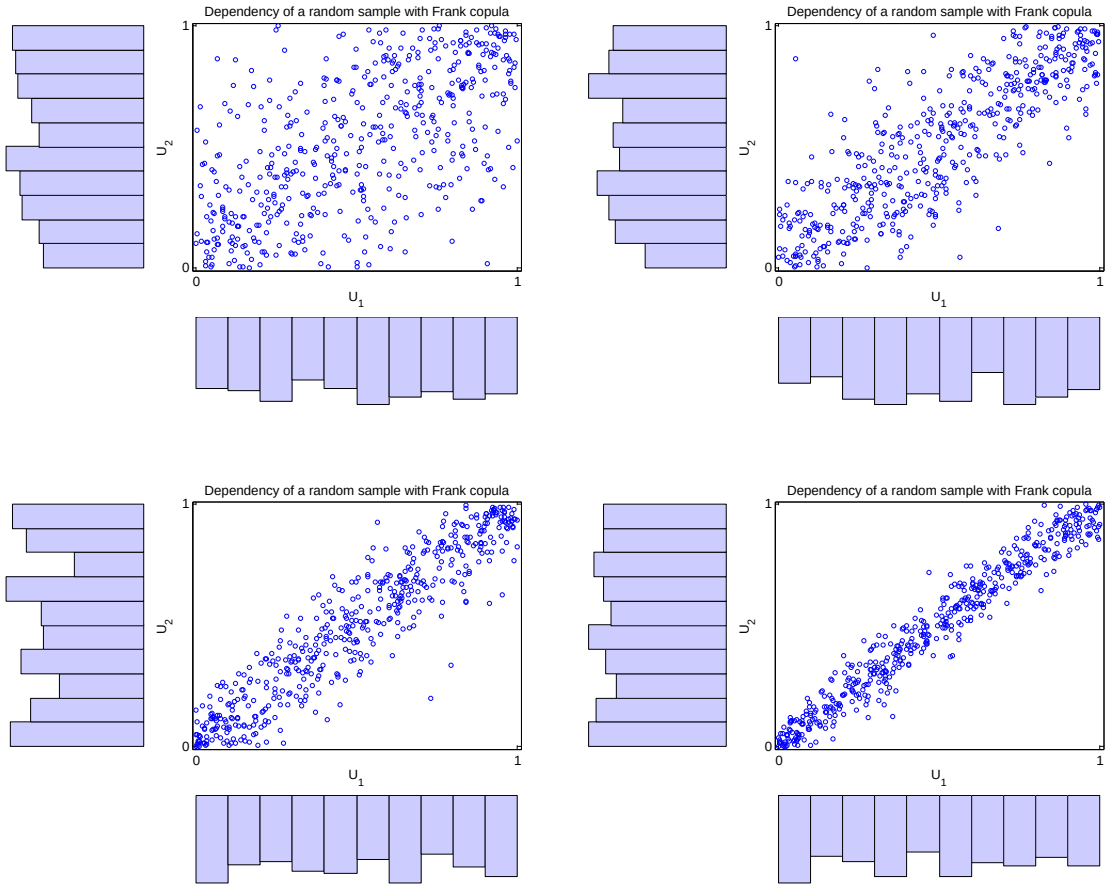


Figure 4.5: The positive correlation imposed to two random variates when sampling from a Frank copula with increasing values of λ. λ ranges in $[5, 10, 15, 25]$ from top-left to bottom right respectively.

the Archimedean copulas. Archimedean copulas are widely used copulas and are defined with respect to a generator function ψ. They take the form: $C(u_1, \cdots, u_d) = \psi^{-1}(\psi(u_1) + \cdots + \psi(u_d))$. A special case of Archimedean copulas corresponds to Frank copulas, which are obtained by setting: $\psi_\lambda(u) = \frac{-1}{\lambda}\log(1-(1-e^{-\lambda})e^{-u})$. When $\lambda \to 0$, the Frank copula approaches the independency copula; when $\lambda \to \infty$ it approaches the co-monotonicity copula. Hence, the Frank copula allows one to model all dependencies between complete independence to perfect dependence while varying λ from 0 to ∞. Therefore, λ can be seen as an additional hyper-parameter to be tuned from the data. Figure 4.5 illustrates the positive dependence between two random variables sampled from a Frank copula. To highlight the effect of λ in the correlation imposed at the sample we visualize samples while increasing its value. Following the increase of the λ values the correlation between the values of the variates increases. To sample from the Archimedean copulas, we rely on the algorithm proposed by [119], which was further improved in [123, 82]

and implemented in the R language [83].

Lately, there is an increasing interest over the integration of copulas in machine learning applications [53] such as classification [52] or structure learning [111]. Interestingly, [186] have shown how to incorporate copulas in Gaussian processes in order to model the dependency between random variables with arbitrary marginals with a practical application on predicting the standard deviation of variables in the financial sector (volatility estimation). In another generic framework, [172] have shown the benefits of using copulas to model complex dependencies between latent variables in the general variational inference setting.

The idea of using copulas with topic models was recently investigated in the interesting work of [3]. In the context of document streams they proposed a topic model where the dependencies between the topic distributions of two consecutive documents are captured by copulas. Here, instead of modeling the dependence between topic distributions of consecutive documents, we model the dependence between the topics assigned to the words of segments we consider coherent.

4.3.3 copulaLDA: Integrating segment boundaries to LDA using copulas

In the previous section (Section 4.3.2) we introduced copulas, a powerful framework for modeling the joint distribution of random variables. The capacity of copulas to model joint distributions by decoupling the underlying dependence of the variables from their marginals can be a useful property for topic modeling. Our motivation lies in the way that we sample topics for segments: we want a mechanism to model the joint distribution using information from the marginals. For topic modeling, the marginals describe how topics occur with each word of a segment while the joint distributions concern the topics assigned to the words of the segments. As discussed previously in the framework of segLDA, one expects a dependence mechanism to apply between the topics of segments.

In this section we develop *copulaLDA* (hereafter copLDA), that extends LDA by integrating simple text structure in the model using copulas. We assume that the topics that generate the terms of coherent text spans are bound. A strong binding signifies high probability for the terms to have been generated by the same topic. Therefore, as we show, the conditional independence of topics given the per-document topic distributions does not hold.

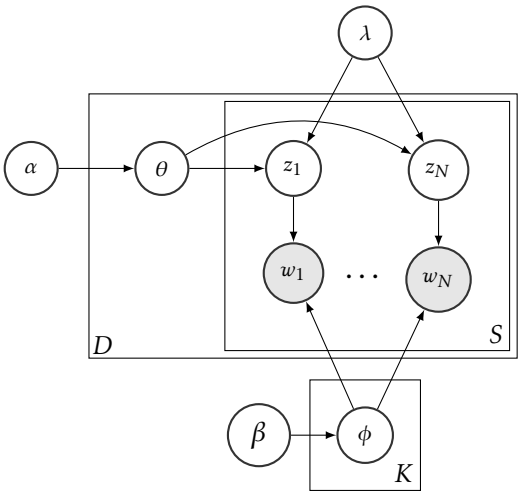


Figure 4.6: The copLDA generative model. We model the dependency between the topics underlying a segment with copulas. Notice in the graphical model how the topics of $(z_1,\ldots,z_N)$ of a segment depend both on θ and the copulas, illustrated with its λ parameter.

Copulas provide an intuitive way to bind random variables. We are making use of them here to bind word-specific topics (the z variables in LDA) within coherent text spans, the rationale being that coherent text spans can not be generated by many different, uncorrelated topics. This leads us to the following generative model:

- For each topic $k \in [1,K]$, choose a per-word distribution: $\phi_k \sim Dir(\beta)$, with $\phi_k, \beta \in \mathbb{R}^{|V|}$
- For each document d_i, $i \in \{1,\ldots,D\}$:
  - Choose a per-document topic distribution: $\theta_i \sim Dir(\alpha)$, with $\theta_i, \alpha \in \mathbb{R}^{|K|}$
  - Sample number of segments in d_i: $S_i \sim Poisson(\xi)$;
  - For each segment $s_{i,j}, j \in \{1,\ldots,S_i\}$:
    - Sample number of words: $N_{i,j} \sim Poisson(\xi_d)$;
    - Sample topics $\mathcal{Z}_{i,j} = (z_{i,j,1},\ldots,z_{i,j,N_{i,j}})$ from a distribution admitting $Mult(1,\theta_i)$ as margins and C as copula;
    - Sample words $W_{i,j} = (w_{i,j,1},\ldots,w_{i,j,N_{i,j}})$: $w_{i,j,n} \sim Mult(1,\phi_{z_{i,j,n}})$, $1 \leq n \leq N_{i,j}$.

There are two main differences between copLDA and LDA. Firstly, the former assumes a hierarchical structure in the documents: the topics that generate the

words in the coherent segments exhibit topical correlation, hence the conditional independence assumption between the terms of a segment given the document per-topic distribution (θ_i) no longer holds. Secondly, this topical correlation is modeled using copulas. Figure 4.6 provides the graphical model for copLDA. For clarity, we draw each word in a coherent segment S, $(w_1,\ldots,w_N)$ to make the dependencies explicit. Notice how the topics of those words depend on both the copula parameter λ and the per-document topic distribution θ.

There is also an important difference between copLDA and segLDA. While the latter assumes that the words $(w_1,\ldots,w_N)$ of a segment are generated by a single topic, the former allows more flexibility. As a result, more topics may be observed within a segment. The copula hyperparameter λ as well as the family of the copulas chosen control this flexibility. Notice that in the limit of total dependence (when $\lambda \to \infty$) copLDA becomes equivalent with segLDA.

The hyper-parameters α and β correspond to priors of the model. Their values can be set according to values that previous work has proposed (e.g., [23]) or can be tuned using the data. Similarly, the hyper-parameter λ can be chosen after exploration of a grid of possible values.

Inference with Gibbs sampling The parameters of the above model, that are ϕ, θ and the topics of each segment $\mathcal{Z}_{i,j} = (z_{i,j,1},\cdots,z_{i,j,N_{i,j}})$, can be directly estimated through Gibbs sampling. Denoting Ω and Ψ the count matrices such that $\Omega = (\Omega_{i,k})$ (resp. $\Psi = (\Psi_{k,v})$) represents the count of word belonging to topic k assigned to document d_i (resp. the count of word v being assigned to topic k), the Gibbs updates for θ and ϕ are the same as the ones for the standard LDA model [23]:

$$\theta_i \sim Dir(\alpha + \Omega_i) \quad \text{and} \quad \phi_k \sim Dir(\beta + \Psi_k) \tag{4.7}$$

The update for the variables z is obtained as follows:

$$p(\mathcal{Z}_{i,j}|\mathcal{Z}_{\neg i,j}, W, \Theta, \Phi, \alpha, \beta, \lambda) = \frac{p(\mathcal{Z}_{i,j}, \mathcal{Z}_{\neg i,j}, W|\Theta, \Phi, \alpha, \beta, \lambda)}{p(\mathcal{Z}_{\neg i,j}, W|\Theta, \phi, \alpha, \beta, \lambda)} =$$
$$\frac{p(\mathcal{Z}_{i,j}, W_{i,j}|\Theta, \Phi, \lambda)p(\mathcal{Z}_{\neg i,j}, W_{\neg i,j}|\Theta, \Phi, \lambda)}{p(W_{i,j}|\Theta, \phi)p(\mathcal{Z}_{\neg i,j}, W_{\neg i,j}|\Theta, \Phi, \lambda)} = \frac{p(\mathcal{Z}_{i,j}, W_{i,j}|\Theta, \Phi, \lambda)}{\sum_{\mathcal{Z}_{i,j}} p(\mathcal{Z}_{i,j}, W_{i,j}|\Theta, \Phi, \lambda)} =$$
$$\frac{p(W_{i,j}|\mathcal{Z}_{i,j}, \Phi)p(\mathcal{Z}_{i,j}|\Theta, \lambda)}{\sum_{\mathcal{Z}_{i,j}} p(W_{i,j}|\mathcal{Z}_{i,j}, \Phi)p(\mathcal{Z}_{i,j}|\Theta, \lambda)} \sim p(W_{i,j}|\mathcal{Z}_{i,j}, \Phi)p(\mathcal{Z}_{i,j}|\Theta, \lambda) =$$
$$p(\mathcal{Z}_{i,j}|\Theta, \lambda)\prod_{n=1}^{N_{i,j}} \phi_{w_{i,j,n}, z_{i,j,n}} \tag{4.8}$$

where W, Θ and Φ stand for the whole parameter set of w, θ and ϕ and the probability outside the product in the last step admits a copula C_λ and $Mult(1, \theta_i)$ as margins. The notation $-i, j$ means excluding the information for i, j. Note that in case where $\lambda \to 0$, the words of a segment become conditionally independent given the per-document distribution and one recovers the non collapsed Gibbs sampling updates of LDA.

From the expression of Eq. (4.8), a simple acceptance/rejection algorithm can be formulated: (1) Sample a random variable of pdf $p(\mathcal{Z}_{i,j}|\Theta, \lambda)$ using copula, and, (2) Accept the sample with probability $p(W_{i,j}|\mathcal{Z}_{i,j}, \Phi) = \prod_{n=1}^{N_{i,j}} \phi_{w_{i,j,n}, z_{i,j,n}}$. Algorithm 2 summarizes the inference process.

Computational Considerations As the values of $\phi_{w_{i,j,1}, z_{i,j,1}} \times \cdots \times \phi_{w_{i,j,n}, z_{i,j,n}}$ tend to be very low, the acceptance/rejection sampling step described above is very slow in practice (see below). We propose here to speed it up by considering, for each word $w_{i,j,n}$ in a given segment, not the exact probability of $z_{i,j,n}$, but its mean (noted M) over all the other words in the segment:

$$M(z_{i,j,n}|\mathcal{Z}_{-i,j}, W, \Theta, \Phi, \alpha, \beta, \lambda) = \sum_{w_{ij,l}, l \neq n} \sum_{z_{ij,l}, l \neq n} P(\mathcal{Z}_{i,j}|\mathcal{Z}_{-i,j}, W, \Theta, \Phi, \alpha, \beta, \lambda) \propto \phi_{w_{i,j,n}} \theta_{d, z_{i,j,n}}$$

as $\sum_{w_{ij,l}} \phi_{w_{i,j,l}} = 1$. In the experimental part we will empirically illustrate the effect of the mean approximation as described above.

4.4 The Experimental Evaluation

In this section we evaluate the copLDA and segLDA models presented above. To this end, we propose both *intrinsic* and *extrinsic* evaluation tasks. The assessment

Algorithm 2: A Gibbs Sampling iteration for copLDA

Input: documents' words grouped in segments, α, β, K, Copula family and its parameter λ
//Initialize counters Ψ, Ω
for document $d_i, i \in [1, D]$ **do**
  for segment $s_{i,j} : j \in \{1, \ldots, S_i\}$ **do**
    Draw a random vector $U = (U_1, \ldots, U_{N_{i,j}})$ that admits a copula C_λ
    do /* If the mean approximation is used, the loop is done once, ignoring the acceptance condition */
      for words $w_{i,j,k}, k \in [1, W_{N_{i,j}}]$ in $s_{i,j}$ **do**
        Decrease counter variables Ψ, Ω
        Get $z_{i,j,k}$ by transforming U_k to Mult. marginals with the generalized inverse
        Assign topic $z_{i,j,k}$ to $w_{i,j,k}$
        Increase counters Ψ, Ω
      end
    while Accept the new segment topic assignments with probability $\phi_{w_{i,j,1}, z_{i,j,1}} \times \cdots \times \phi_{w_{i,j,n}, z_{i,j,n}}$
  end
end

of the performance of topic models in an intrinsic way signifies that no application is used. Common ways to intrinsically evaluate topic models is by measuring the coherence of the produced topic or by estimating their generalization performance. On the other hand, extrinsic evaluation of topic models requires an application like text classification or document retrieval.

Models In the experiments of this section we compare the following topic models:

(i.) LDA[3] as proposed in [23] using the collapsed Gibbs sampling inference [73],

(ii.) segLDA$_{\text{bi}}$ as described in Section 4.3.1 with the 1,000 most frequent bigrams to be considered as coherent segments,

(iii.) segLDA$_{\text{tri}}$ as described in Section 4.3.1 with the 1,000 most frequent trigrams to be considered as coherent segments,

(iv.) segLDA$_{\text{np}}$ as described in Section 4.3.1 with noun-phrases as coherent segments,

[3]We dub with typewriter font the implementations of the models we use from the experiments we performed.

(v.) segLDA$_{\text{sent}}$ as described in Section 4.3.1 with sentences as coherent segments,

(vi.) copLDA$_{\text{bi}}$, copLDA$_{\text{tri}}$, copLDA$_{\text{np}}$, copLDA$_{\text{sent}}$ that use copulas to extend the previous models following the development of Section 4.3.3.

In total, we considered nine models in our experiments that extend LDA using different types of segments and different binding mechanisms between the topics of the words that constitute the segments. For copLDA$_{\text{x}}$ models, we use the Frank copula which was reported to obtain the best performance in similar tasks [3] and was also found to achieve the best performance in our local validation settings compared to Gumbel and Clayton copulas. We have implemented the models using Python. For sampling the Frank copulas we used the R copula package [83] and rPY.[4] Also, λ is set to values which we found to perform well in every dataset we tried, that is to 2 for copLDA$_{\text{sent}}$ and to 5 for copLDA$_{\text{np,bi,tri}}$. Furthermore, the hyper-parameters α and β where set to $1/K$ and 0.01 respectively following [73], where K is the number of topics. For the shallow parsing step, required for copLDA$_{\text{np}}$, we used the Stanford Parser [94]. The text pre-processing steps performed are: lower-casing, stemming using the Snowball Stemmer and removal of numeric strings.

Datasets We use the following publicly available data collections to test the performance of the topic models:

- 20NG (20 news groups), which is a standard text dataset for such tasks as provided by [20],
- Reuters (Reuters-21578, the "ModApte" version), also discussed in [20],
- TED, that are transcriptions of TED talks released in the framework of the International Workshop on Spoken Language Translation 2013 evaluation campaign[5] (we have merged the train, development and test parts and we selected the transcriptions with at least one associated label among the 15 most common in the data[6]),
- Wiki$_x$, with $x \in \{15, 37, 46\}$ and PubMed, both excerpts[7] from the Wikipedia dataset of [141] and the PubMed dataset of [174] used in [11]

[4] https://pypi.python.org/pypi/rpy2

[5] http://workshop2013.iwslt.org/59.php

[6] Technology, Culture, Science, Global Issues, Design, Business, Entertainment, Arts, Politics, Education, Art, Creativity, Health, Biology and Music.

[7] https://github.com/balikasg/topicModelling/tree/master/data

	Basic Statistics of the datasets used				
	Docs.	$\|N\|$	$\|V\|$	Categories	Categories/Instance
TED	1,096	1.16M	30.4K	15	2.42
PubMed	5498	1.09M	28.7K	50	1.32
Reuters	10,788	875K	21.4K	90	1.23
20NG	19,056	1.7M	75.4K	20	1.0
Wiki15	1,198	162K	13.4K	15	1.0
Wiki37	2,459	317K	19.7K	37	1.0
Wiki46	3,657	478K	23.4K	46	1.0
Austen	5,262	170K	6.3K	-	-

Table 4.1: The basic statistics of the datasets used for evaluating the topic models. $|N|$ denotes the total words in the corpus, $|V|$ the vocabulary size, while "Categories" and "Categories/Instance" the size of the category set of the corpus and the average number of categories per document respectively.

- "Austen", where we concatenated three books[8] written by Jane Austen, available from the Gutenberg project (each paragraph is considered as a document).

Table 4.1 presents some basic statistics for these datasets. The goal when selecting them was to evaluate the models on data of different types, ranging from forum messages (20NG) to news stories (Reuters) or literature ("Austen"). In that way we exclude any possible bias due to the type or source of the text.

The effect of the mean approximation during inference Figure 4.7 compares the perplexity scores achieved in 200 documents from the "Wiki46" Wikipedia dataset of Table 4.1 by the copLDA model, when considering noun-phrases as coherent spans, with and without rejection sampling. We repeat the experiment 10 times and also plot the standard deviation. We first note that approximating Algorithm 1 by ignoring the rejection sampling step results in slightly worse performance. On the other hand, without the rejection sampling, copLDA converges faster in terms of iterations. Furthermore, the cost in terms of running time of a single iteration is significantly smaller: for instance, for 30 iterations with rejection sampling, the algorithm needs almost 6 hours, that is 100 times more than the 3.5 minutes needed without the rejection sampling. Hence, in the rest of the study, for scaling purposes, we adopt the above mean approximation.

[8]We used the books: Emma, Persuasion, Sense. We considered each paragraph as a document.

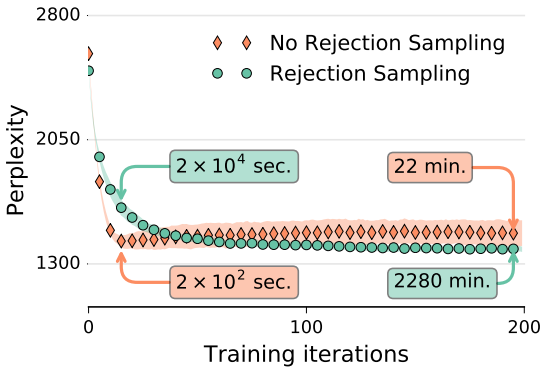


Figure 4.7: The effect of rejection sampling in efficiency and perplexity performance for copLDA.

4.4.1 Intrinsic Evaluation

Intrinsic evaluation of topic models is a way of evaluating the topic models without using a real application. Such types of evaluation usually assess the coherence of the produced topics, either by manual inspection or by calculating some scores that are indicative of how often words occur in similar contexts. The most-used intrinsic measure to evaluate topic models is probably perplexity. As, however, it was shown that perplexity does not always correlate well with human judgments of the quality of the produced topics [31], various measures have been proposed as alternatives [135, 128, 102]. In this section, we evaluate the proposed topic models with respect to several of these measures. We begin by visualizing the learned topics, we continue by reporting the perplexity scores and then, we discuss the topic coherence performance with regard to the normalized point-wise mutual information scores.

Manual inspection of the topics We begin by comparing the topics learned by LDA and copLDA$_{\text{np}}$. We choose to visualize the results for those two models as LDA is our main baseline while copLDA$_{\text{np}}$ integrates noun-phrase boundaries which are short and easy to visualize. For presentation purposes, we train the two topic models using the Wiki$_{47}$ dataset with 10 topics and we illustrate the top-10 words learned for each topic by the two models in Table 4.2. As one can note, since the two models have been trained on the same data with the same training parameters, the identified topics are very similar. This said, copLDA$_{\text{np}}$ manages to produce arguably better topics. This is for example the case for the topic "Birth"; although both models assign high probability to words like "born" and "american" due to

the content of the dataset, copLDA$_{np}$ manages to identify several words corresponding to months which makes the topic more thematically consistent and easier to interpret compared to its LDA counterpart.

Profession	Science	Books	Art	Cinema	Places	Music	Birth	Elections	Inventions
profession	univers	book	art	film	state	record	born	elect	california
world	research	new	new	televis	unit	music	american	canadian	plant
footbal	scienc	work	work	role	us	band	known	parti	use
wrestl	professor	american	paint	appear	township	album	best	member	invent
play	work	publish	york	also	school	song	actress	liber	flower
born	institut	time	american	actor	univers	also	decemb	minist	compani
american	award	author	artist	born	serv	produc	june	hous	north
championship	prize	also	museum	play	war	releas	april	canada	patent
team	born	year	painter	seri	nation	new	juli	serv	inventor
first	receiv	york	studi	star	build	singer	januari	conserv	found
known	univers	book	art	film	township	record	play	elect	work
wrestl	research	new	new	born	state	music	footbal	canadian	first
born	scienc	american	york	televis	counti	band	born	serv	year
world	professor	author	paint	role	us	album	american	parti	photograph
profession	work	publish	american	actor	california	song	tour	member	design
american	institut	novel	work	appear	michigan	also	golf	liber	state
name	born	time	artist	also	plant	singer	year	hous	new
wrestler	prize	also	painter	seri	civil	releas	profession	minist	use
best	studi	writer	museum	actress	popul	produc	first	state	also
championship	award	magazin	born	american	flower	american	season	born	build

Table 4.2: The topics learned by copLDA (upper half) and LDA (lower half) in the Wiki46 dataset.

Kiss of Death is *a 1995 crime thriller film* starring *David Caruso Samuel L. Jackson* and *Nicolas Cage*. *The film* is a *very loosely* based remake of *the 1947 film noir classic* of *the same name* that starred *Victor Mature*, *Brian Donlevy* and *Richard Widmark*.	Kiss of Death is *a 1995 crime thriller film* starring *David Caruso Samuel L. Jackson* and *Nicolas Cage*. *The film* is a *very loosely* based remake of *the 1947 film noir classic* of *the same name* that starred *Victor Mature*, *Brian Donlevy* and *Richard Widmark*.
Bertram Stern (born 3 *October 1929*) is *an American fashion and celebrity portrait photographer*.	*Bertram Stern* (born 3 *October 1929*) is *an American fashion and celebrity portrait photographer*.
Dana Hill (born *Dana Lynne Goetz* in *Los Angeles*, California; *May 6, 1964 - July 15, 1996*) was *an American actress and voice actor* with *a raspy voice and childlike appearance*, which allowed her to play *adolescent roles* well into *her 20s*.	*Dana Hill* (born *Dana Lynne Goetz* in *Los Angeles*, California; *May 6, 1964 - July 15, 1996*) was *an American actress and voice actor* with *a raspy voice and childlike appearance*, which allowed her to play *adolescent roles* well into *her 20s*.

Table 4.3: The discovered topics underlying the words of example documents for LDA (left) and copLDA (right). The parts of the documents in italics indicate the noun-phrases obtained by the Stanford Parser. The text colors refer to the topics described in Table 4.2.

In the same line, Table 4.3 visualizes the inferred topics for parts of the Wiki$_{47}$ dataset. Recall, the first sentence of the table is the example out in the beginning of the chapter (Fig. 4.1) that motivated the work presented in the chapter. Notice here that given the topic interpretations of Table 4.2, both models manage to identify intuitive topics. Note, however, how in most of the cases the text structure

information used by copLDA$_{np}$ helps to obtain consistent topics for noun-phrases like "crime thriller film" and "raspy voice", a consistency that LDA is lacking. Of course, this is not the case for every noun-phrase of the corpus: there are cases like "Dana Lynne Goetz" where two (or more) topics are assigned to the words of the phrases. This was expected as, by definition, the strength of the bound that copLDA$_{np}$ applies for sampling the topics of words of a segment is not maximal but is controlled by the λ hyperparameter of the copula.

Intrinsic Evaluation: Perplexity We continue our evaluation by presenting perplexity scores of held-out documents, calculated for each of the topic model datasets of Table 4.1. Achieving lower perplexity score means that a topic model can explain unseen data more efficiently, thus it generalizes better and it is, in turn, a better model.

As a result, a good model with low perplexity should be able to infer better representations for the unseen documents. As perplexity does not use any real application to evaluate the topic models it is also an intrinsic metric. For a set of test documents C consisting of N words $\{w_1, \ldots, w_N\}$ the perplexity is calculated using:

$$perpl(C) = \exp\left\{-\frac{\sum_{i=1}^{N} \log p(w_i)}{N}\right\} \tag{4.9}$$

Hence, the best the model fits the data, the higher will be the $p(w_i)$ and consequently the lowest the perplexity score achieved.

In order to estimate perplexity, the topic distributions of the unseen documents are required. They can be obtained by repeating the Gibbs sampling inference process for the unseen (held-out) documents. During this process, however, the per-word topic distributions learned during training are kept constant.

In our experiments here, we split the documents of a dataset randomly in two parts with 80%/20% of the documents: we use the former for learning the model and the second for calculating the perplexity scores. We use exactly the same splits for training and evaluating each of the topic models. Table 4.4 illustrates the achieved perplexity scores for the datasets of Table 4.1 and for the number of topics $K \in \{25, 50, 75, 100, 150\}$. The best (lowest) perplexity score per dataset and number of topics is shown in bold. There are several observation to be made from the table.

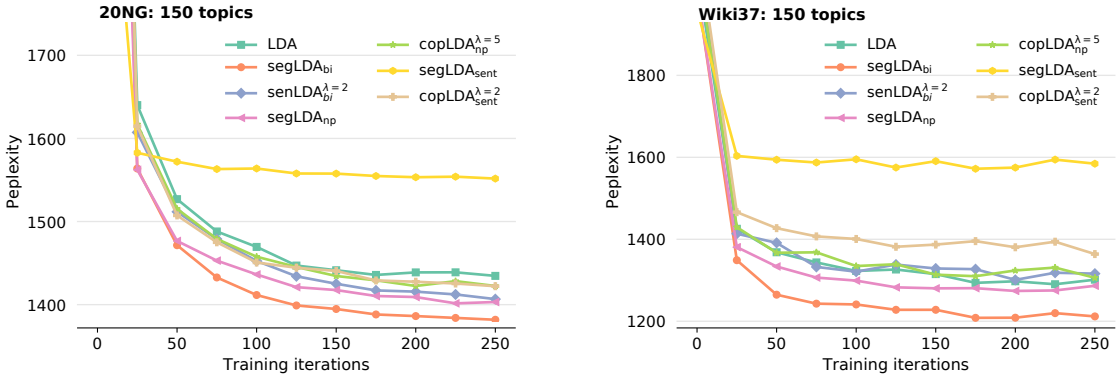


Figure 4.8: The perplexity curves of the investigated models for 200 Gibbs sampling iterations and different datasets.

First, notice that the lowest perplexity is consistently achieved by segLDA$_{bi}$ and copLDA$_{bi}$. This suggests that assuming short segments to be coherent benefits perplexity scores. Such a finding is intuitive in that frequent bigrams usually refer to entities like destinations (e.g., New York, United Kingdom) and assuming them to be coherent benefits the model's generalization performance. On the other hand, models that assume larger segments, like sentences (segLDA$_{sent}$ and copLDA$_{sent}$) to be coherent are not competitive.

Another interesting observation concerns the effect of copulas when various segments are considered. One can identify two families of models: those whose perplexity scores greatly improve due to the use of copulas: copLDA$_{sent}$ and copLDA$_{np}$ and those that do not benefit much: copLDA$_{bi}$ and copLDA$_{tri}$. These results suggest that the size of segment and the frequency of co-occurrence is important for deciding whether to consider copulas or not. Short and frequent segments like bigrams are by definition coherent and having a flexible binding scheme like copulas does not improve the perplexity scores. On the other hand, sentences and noun-phrases that are less thematically consistent benefit from the flexibility that copulas offer.

Figure 4.8 illustrates the perplexity curves of the hold-out documents for seven of the models on two of the datasets of Table 4.1 for 250 Gibbs sampling iterations. Note that segLDA$_{sent}$ is the model with the fastest convergence rate with respect to the number of Gibbs iterations. On the other hand, LDA, copLDA$_{sen}$ and copLDA$_{np}$ require the same number of iterations, which depends on the dataset. copLDA$_{bi}$ manages to achieve the lowest perplexity scores. Notice its steep curves in the first iterations.

Dataset	K	LDA	segLDA$_{np}$	copLDA$_{np}$	segLDA$_{sent}$	copLDA$_{sent}$	segLDA$_{bi}$	copLDA$_{bi}$	segLDA$_{tri}$	copLDA$_{tri}$
	25	1,626	1,654	1,616	1,747	1,642	1,618	1,618	1,613	**1,610**
	50	1,508	1,501	1,485	1,625	1,520	**1,469**	1,482	1,487	1,483
20NG	75	1,464	1,460	1,435	1,600	1,464	**1,426**	1,435	1,437	1,432
	100	1,431	1,435	1,415	1,547	1,451	**1,385**	1,414	1,412	1,408
	150	1,434	1,401	1,422	1,550	1,422	**1,381**	1,406	1,406	1,405
	25	761	764	773	948	831	**748**	770	773	775
	50	748	744	748	939	815	**709**	754	751	751
Austen	75	748	730	746	945	826	**708**	745	745	760
	100	757	729	755	953	825	**696**	751	746	756
	150	762	728	761	966	845	**699**	763	761	766
	25	1,093	1,122	1,098	1,249	1,093	**1,092**	1,098	1,118	1,117
	50	961	987	972	1,160	990	**957**	968	992	999
PubMed	75	928	951	932	1,130	954	**904**	926	955	954
	100	930	922	914	1,110	946	**884**	927	930	957
	150	938	891	939	1,095	957	**867**	937	940	962
	25	**548**	568	551	594	590	555	549	620	615
	50	487	502	489	545	530	**479**	486	543	554
Reuters	75	468	470	469	528	511	**448**	464	518	527
	100	464	452	459	508	499	**436**	455	505	518
	150	454	437	455	491	495	**426**	456	495	511
	25	1,645	1,645	**1,640**	1,681	1,636	1,652	1,640	1,651	1,661
	50	1,579	1,587	**1,568**	1,647	1,579	1,571	1,574	1,585	1,590
Ted	75	1,543	1,546	**1,544**	1,641	1,551	1,551	1,545	1,560	1,563
	100	1,518	1,531	1,522	1,637	1,533	**1,512**	1,527	1,531	1,537
	150	1,501	1,503	1,505	1,634	1,505	**1,492**	1,508	1,518	1,519
	25	1,161	1,213	1,156	1,410	1,195	1,150	**1,148**	1,237	1,239
	50	1,095	1,133	1,096	1,418	1,143	**1,077**	1,104	1,181	1,184
Wiki15	75	1,101	1,120	1,092	1,426	1,144	**1,037**	1,092	1,167	1,188
	100	1,090	1,097	1,113	1,405	1,159	**1,029**	1,107	1,168	1,205
	150	1,119	1,098	1,149	1,393	1,201	**1,023**	1,146	1,184	1,238
	25	1,357	1,422	**1,343**	1,636	1,386	1,358	1,356	1,446	1,451
	50	1,270	1,311	1,272	1,558	1,315	**1,254**	1,271	1,348	1,356
Wiki37	75	1,260	1,292	1,271	1,584	1,305	**1,207**	1,269	1,325	1,355
	100	1,271	1,273	1,281	1,562	1,312	**1,209**	1,274	1,335	1,358
	150	1,290	1,273	1,306	1,556	1,363	**1,208**	1,301	1,346	1,405
	25	1,366	1,427	1,366	1,561	1,417	**1,358**	1,360	1,441	1,442
	50	1,259	1,278	1,261	1,455	1,302	**1,211**	1,254	1,312	1,336
Wiki46	75	1,227	1,224	1,233	1,396	1,278	**1,151**	1,214	1,274	1,288
	100	1,216	1,181	1,226	1,355	1,276	**1,153**	1,228	1,266	1,303
	150	1,236	1,125	1,243	1,301	1,298	**1,123**	1,240	1,269	1,319

Table 4.4: The perplexity scores (Eq. 4.9) achieved on each of the datasets used. The best scores for each dataset are shown in bold font.

Intrinsic Evaluation: Normalized Pointwise Mutual Information Automatically evaluating the coherence of the topics produced by topic models is a task that has received a lot of attention. The goal is to measure how coherent or interpretable the produced topics are [128]. It has been recently found that scoring the topics using co-occurrence measures, such as the pointwise mutual information (PMI) between the top words of a topic, correlates well with human judgments [135]. To achieve that, an external corpus like Wikipedia is treated like a meta-document, which is used as the basis to calculate the PMI scores of words using a sliding window and applying the equation:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}. \tag{4.10}$$

Evaluating the topic coherence requires selecting the top-N words of a topic and performing the manual or automatic evaluation. Here, N is a hyper-parameter to be chosen and its value can impact the results. Very recently, Lau and Baldwin [102] showed that N actually impacts the quality of the obtained results and in particular the correlation with human judgments. In their study they conclude that aggregating the topic coherence scores over several topic cardinalities, leads to a substantially more stable and robust evaluation.

Following these findings, we present in Table 4.5 the topic coherence scores as measured by the normalized pointwise mutual information (nPMI). The scores of nPMI range in [-1,1], where in the limit of -1 two words w_1 and w_2 never occur together, while in the limit of +1 they always occur together (complete co-occurrence). As in [102], for each topic, we aggregate the topic coherence scores over three different topic cardinalities: $N \in \{5, 10, 15\}$. The table presents the results when training the topic models for various numbers of topics K, in particular for $K \in \{25, 50, 75, 100, 150\}$. The numbers reported are the averages of the nPMI scores and are calculated as follows: we train the topic models for 250 Gibbs sampling iterations. After the 150-th iteration (including the 150-th) we sample the top-N words of the topics every 25 iterations. As a result, we have 5 samples in total, and for those we calculate the average nPMI scores and standard deviations and report them in Table 4.5.

One may observe from the table that, in general, increasing the number of topics decreases the coherence of the learned topics. This applies for every topic model and dataset we tried. In most of the cases, the most coherent topics in terms of nPMI are obtained with 25 topics. In terms of the types of the segments

	K	LDA	segLDA$_{np}$	copLDA$_{np}$	segLDA$_{sent}$	copLDA$_{sent}$	segLDA$_{bi}$	copLDA$_{bi}$	segLDA$_{tri}$	copLDA$_{tri}$
20NG	25	$.118_{\pm.07}$	$\mathbf{.127}_{\pm.07}$	$.113_{\pm.06}$	$.108_{\pm.07}$	$.117_{\pm.07}$	$.112_{\pm.06}$	$.115_{\pm.07}$	$.084_{\pm.05}$	$.117_{\pm.06}$
	50	$.111_{\pm.07}$	$\mathbf{.114}_{\pm.06}$	$.113_{\pm.06}$	$.113_{\pm.07}$	$.111_{\pm.07}$	$.110_{\pm.06}$	$.113_{\pm.06}$	$.081_{\pm.06}$	$.107_{\pm.06}$
	75	$.105_{\pm.06}$	$\mathbf{.109}_{\pm.07}$	$.103_{\pm.06}$	$.107_{\pm.06}$	$.109_{\pm.07}$	$.103_{\pm.05}$	$.104_{\pm.06}$	$.068_{\pm.06}$	$.109_{\pm.06}$
	100	$.103_{\pm.06}$	$\mathbf{.112}_{\pm.06}$	$.105_{\pm.06}$	$.102_{\pm.06}$	$.101_{\pm.06}$	$.096_{\pm.05}$	$.097_{\pm.06}$	$.071_{\pm.06}$	$.100_{\pm.06}$
	150	$.098_{\pm.06}$	$\mathbf{.102}_{\pm.06}$	$.098_{\pm.06}$	$.097_{\pm.06}$	$.095_{\pm.06}$	$.093_{\pm.06}$	$.096_{\pm.05}$	$.058_{\pm.05}$	$.095_{\pm.05}$
Austen	25	$.076_{\pm.04}$	$.077_{\pm.04}$	$.082_{\pm.04}$	$\mathbf{.088}_{\pm.04}$	$.084_{\pm.04}$	$.082_{\pm.03}$	$.077_{\pm.04}$	$.060_{\pm.03}$	$.076_{\pm.04}$
	50	$.069_{\pm.04}$	$.072_{\pm.03}$	$.066_{\pm.03}$	$.070_{\pm.04}$	$.066_{\pm.03}$	$\mathbf{.080}_{\pm.04}$	$.068_{\pm.03}$	$.057_{\pm.03}$	$.068_{\pm.03}$
	75	$.065_{\pm.03}$	$.062_{\pm.03}$	$.066_{\pm.03}$	$.064_{\pm.03}$	$.064_{\pm.03}$	$\mathbf{.068}_{\pm.03}$	$.064_{\pm.03}$	$.052_{\pm.02}$	$.065_{\pm.03}$
	100	$.062_{\pm.03}$	$.064_{\pm.03}$	$.060_{\pm.03}$	$.064_{\pm.03}$	$.062_{\pm.03}$	$\mathbf{.066}_{\pm.03}$	$.062_{\pm.03}$	$.046_{\pm.02}$	$.063_{\pm.03}$
	150	$.057_{\pm.03}$	$.059_{\pm.03}$	$.059_{\pm.03}$	$.059_{\pm.03}$	$.059_{\pm.03}$	$\mathbf{.063}_{\pm.03}$	$.058_{\pm.03}$	$.045_{\pm.03}$	$.058_{\pm.03}$
PubMed	25	$.154_{\pm.07}$	$.165_{\pm.08}$	$\mathbf{.167}_{\pm.07}$	$.154_{\pm.06}$	$.153_{\pm.06}$	$.153_{\pm.07}$	$.153_{\pm.07}$	$.153_{\pm.06}$	$.156_{\pm.06}$
	50	$.156_{\pm.07}$	$.154_{\pm.08}$	$\mathbf{.159}_{\pm.07}$	$.163_{\pm.08}$	$.158_{\pm.07}$	$.149_{\pm.08}$	$.160_{\pm.07}$	$.136_{\pm.07}$	$.165_{\pm.07}$
	75	$.159_{\pm.08}$	$.153_{\pm.08}$	$\mathbf{.161}_{\pm.08}$	$.143_{\pm.08}$	$.156_{\pm.08}$	$.155_{\pm.08}$	$.158_{\pm.08}$	$.135_{\pm.08}$	$.159_{\pm.08}$
	100	$.150_{\pm.07}$	$.153_{\pm.08}$	$.150_{\pm.08}$	$.149_{\pm.07}$	$.153_{\pm.08}$	$\mathbf{.156}_{\pm.07}$	$.148_{\pm.08}$	$.134_{\pm.07}$	$.150_{\pm.08}$
	150	$.143_{\pm.07}$	$.144_{\pm.08}$	$.143_{\pm.08}$	$.137_{\pm.07}$	$.139_{\pm.07}$	$\mathbf{.147}_{\pm.07}$	$.146_{\pm.07}$	$.116_{\pm.07}$	$.142_{\pm.08}$
Reuters	25	$.055_{\pm.03}$	$\mathbf{.061}_{\pm.04}$	$.054_{\pm.03}$	$.062_{\pm.03}$	$.061_{\pm.03}$	$.067_{\pm.03}$	$.052_{\pm.03}$	$.047_{\pm.03}$	$.054_{\pm.03}$
	50	$.056_{\pm.03}$	$\mathbf{.073}_{\pm.04}$	$.060_{\pm.03}$	$.062_{\pm.03}$	$.057_{\pm.03}$	$.064_{\pm.03}$	$.055_{\pm.03}$	$.050_{\pm.03}$	$.048_{\pm.03}$
	75	$.057_{\pm.03}$	$\mathbf{.066}_{\pm.04}$	$.057_{\pm.03}$	$.062_{\pm.03}$	$.056_{\pm.03}$	$.064_{\pm.03}$	$.053_{\pm.03}$	$.047_{\pm.03}$	$.051_{\pm.03}$
	100	$.056_{\pm.03}$	$\mathbf{.070}_{\pm.04}$	$.055_{\pm.03}$	$.057_{\pm.03}$	$.058_{\pm.03}$	$.061_{\pm.03}$	$.056_{\pm.03}$	$.048_{\pm.03}$	$.049_{\pm.03}$
	150	$.057_{\pm.03}$	$\mathbf{.068}_{\pm.04}$	$.054_{\pm.03}$	$.060_{\pm.04}$	$.055_{\pm.03}$	$.057_{\pm.03}$	$.053_{\pm.04}$	$.046_{\pm.03}$	$.051_{\pm.03}$
Ted	25	$.070_{\pm.04}$	$\mathbf{.090}_{\pm.04}$	$.072_{\pm.04}$	$.087_{\pm.04}$	$.074_{\pm.05}$	$.073_{\pm.04}$	$.069_{\pm.04}$	$.063_{\pm.03}$	$.064_{\pm.04}$
	50	$.074_{\pm.05}$	$\mathbf{.099}_{\pm.04}$	$.081_{\pm.05}$	$.087_{\pm.05}$	$.080_{\pm.05}$	$.074_{\pm.04}$	$.079_{\pm.06}$	$.064_{\pm.04}$	$.078_{\pm.05}$
	75	$.080_{\pm.05}$	$\mathbf{.100}_{\pm.05}$	$.074_{\pm.05}$	$.089_{\pm.05}$	$.079_{\pm.05}$	$.076_{\pm.05}$	$.075_{\pm.05}$	$.054_{\pm.03}$	$.077_{\pm.05}$
	100	$.073_{\pm.04}$	$\mathbf{.094}_{\pm.05}$	$.072_{\pm.05}$	$.075_{\pm.05}$	$.076_{\pm.05}$	$.071_{\pm.04}$	$.073_{\pm.05}$	$.056_{\pm.04}$	$.074_{\pm.05}$
	150	$.067_{\pm.04}$	$\mathbf{.086}_{\pm.06}$	$.070_{\pm.05}$	$.077_{\pm.05}$	$.071_{\pm.05}$	$.067_{\pm.04}$	$.071_{\pm.05}$	$.051_{\pm.03}$	$.069_{\pm.05}$
Wiki15	25	$.107_{\pm.06}$	$.103_{\pm.06}$	$.102_{\pm.06}$	$.112_{\pm.06}$	$.099_{\pm.06}$	$.109_{\pm.06}$	$.098_{\pm.06}$	$\mathbf{.132}_{\pm.05}$	$.105_{\pm.06}$
	50	$.088_{\pm.06}$	$.086_{\pm.05}$	$.084_{\pm.05}$	$.084_{\pm.06}$	$.081_{\pm.06}$	$.086_{\pm.06}$	$.086_{\pm.06}$	$\mathbf{.106}_{\pm.06}$	$.085_{\pm.06}$
	75	$.077_{\pm.06}$	$.082_{\pm.06}$	$.072_{\pm.06}$	$.073_{\pm.06}$	$.073_{\pm.05}$	$.078_{\pm.05}$	$.076_{\pm.06}$	$\mathbf{.096}_{\pm.06}$	$.074_{\pm.06}$
	100	$.071_{\pm.05}$	$.076_{\pm.05}$	$.068_{\pm.05}$	$.071_{\pm.05}$	$.066_{\pm.05}$	$.070_{\pm.05}$	$.066_{\pm.06}$	$\mathbf{.085}_{\pm.05}$	$.069_{\pm.05}$
	150	$.063_{\pm.05}$	$.064_{\pm.05}$	$.058_{\pm.05}$	$.060_{\pm.05}$	$.054_{\pm.05}$	$.067_{\pm.05}$	$.062_{\pm.05}$	$\mathbf{.077}_{\pm.05}$	$.059_{\pm.05}$
Wiki37	25	$.115_{\pm.05}$	$.124_{\pm.06}$	$.126_{\pm.05}$	$.110_{\pm.05}$	$.105_{\pm.06}$	$.110_{\pm.04}$	$.120_{\pm.05}$	$\mathbf{.127}_{\pm.05}$	$.120_{\pm.05}$
	50	$.103_{\pm.06}$	$.107_{\pm.06}$	$.102_{\pm.06}$	$.101_{\pm.06}$	$.098_{\pm.06}$	$.104_{\pm.05}$	$.097_{\pm.06}$	$\mathbf{.112}_{\pm.05}$	$.102_{\pm.05}$
	75	$.093_{\pm.06}$	$.093_{\pm.06}$	$.089_{\pm.06}$	$.089_{\pm.06}$	$.088_{\pm.06}$	$.096_{\pm.06}$	$.088_{\pm.06}$	$\mathbf{.100}_{\pm.06}$	$.095_{\pm.06}$
	100	$.081_{\pm.06}$	$.086_{\pm.06}$	$.083_{\pm.06}$	$.083_{\pm.06}$	$.079_{\pm.06}$	$.088_{\pm.06}$	$.089_{\pm.06}$	$\mathbf{.093}_{\pm.06}$	$.083_{\pm.06}$
	150	$.078_{\pm.06}$	$.083_{\pm.06}$	$.070_{\pm.06}$	$.071_{\pm.05}$	$.071_{\pm.05}$	$.079_{\pm.05}$	$.075_{\pm.06}$	$\mathbf{.084}_{\pm.06}$	$.072_{\pm.06}$
Wiki46	25	$.119_{\pm.05}$	$.123_{\pm.06}$	$.120_{\pm.05}$	$.114_{\pm.06}$	$.105_{\pm.05}$	$.119_{\pm.05}$	$.106_{\pm.06}$	$\mathbf{.120}_{\pm.05}$	$.117_{\pm.05}$
	50	$.105_{\pm.06}$	$.109_{\pm.06}$	$.109_{\pm.06}$	$.107_{\pm.06}$	$.100_{\pm.06}$	$.108_{\pm.06}$	$.104_{\pm.06}$	$.094_{\pm.06}$	$\mathbf{.115}_{\pm.06}$
	75	$.092_{\pm.06}$	$\mathbf{.105}_{\pm.06}$	$.099_{\pm.06}$	$.094_{\pm.06}$	$.090_{\pm.06}$	$.099_{\pm.06}$	$.101_{\pm.06}$	$.084_{\pm.06}$	$.095_{\pm.06}$
	100	$.089_{\pm.06}$	$.090_{\pm.06}$	$.088_{\pm.06}$	$.083_{\pm.06}$	$.079_{\pm.06}$	$.086_{\pm.06}$	$.089_{\pm.06}$	$.079_{\pm.06}$	$\mathbf{.092}_{\pm.06}$
	150	$.081_{\pm.06}$	$\mathbf{.084}_{\pm.06}$	$.082_{\pm.06}$	$.074_{\pm.06}$	$.077_{\pm.06}$	$.082_{\pm.06}$	$.075_{\pm.06}$	$.072_{\pm.05}$	$.076_{\pm.06}$

Table 4.5: nPMI scores (Eq. 4.10) for the topic models for different values of topics. The best scores for each dataset are shown in bold font.

we consider, once more, short segments likes noun-phrases, bigrams and trigrams perform the best. This, however, also depends on the dataset and, less frequently on the number of topics considered. Further, compared to the perplexity results where copulas improved the performance of topic models with linguistically motivated segments, in the nPMI experiments they do not. In most of the experiments, the models with copulas perform worse than their counterparts that do not use copulas for sampling the segment topics.

Our last observation concerns the nPMI scores: depending on the dataset, one observes higher or lower scores. The highest scores are consistently obtained on the PubMed dataset, while the lowerst on the Reuters. This, of course, greatly depends on the corpus used as a meta-document to estimate the PMI scores (Wikipedia in our case). The results however, suggest, that the topics learned using PubMed documents are more coherent, and we believe that this is due to the consistent use of language in scientific articles, like PubMed documents. To highlight this in Table 4.6 we show the nPMI scores for each topic of the best performing system ($\texttt{copLDA}_{\text{np}}$) for 25 topics for PubMed. From the table, Topic 3 that concern cancer is the most consistent with nPMI=0.27. Interestingly, including more words of the topic in the nPMI calculation improves the scores from .21 to .29 and .31 when considering the top-5, top-10 and top-15 words respectively. On the other hand, Topic 4 that is mainly about genetics, achieves the highest top-5 nPMI scores as the words "gene", "chromosom", "mutat", "tumor", and "genet" have nPMI=.40. Adding more words in the nPMI calculation however, worsens the score and results in an average nPMI of .24. We conclude the nPMI evaluation by commenting on a limitation of the approach: Topic 14 with achieves the lowest nPMI score of .03, is still consistent. Its top-5 words ("patient", "group", "p", "signific", "age", "studi") that probably describe group studies, although coherent for scientists, are scored with nPMI=0.0 probably due to meta-document we used for calculating the PMI probabilities. As Wikipedia is out-of-domain, one typically expects few entries to discuss similar topics and this impacts the scores of the topic.

4.4.2 Extrinsic Evaluation

Compared to the previous evaluation subtasks where we presented results of intrinsic evaluation, we now present a task for extrinsic evaluation of the topic models. Extrinsic evaluation of topic models uses a real task such as clustering [147] or classification to assess the performance of the representations learned with topic models.

Topic	nPMI	$nPMI_5/nPMI_{10}/nPMI_{15}$	Top-15 words per topic
1	.17	.20/.18/.13	case patient tumor diagnosi present clinic report diseas lesion find histolog examin year one rare
2	.10	.10/.10/.09	use care health rate cost hospit system provid data payment state medic medicar new physician
3	.27	.21/.29/.31	cancer mutat polyposi patient famili apc colon adenomat adenoma tumor fap polyp colorect intestin carcinoma
4	.24	.40/.18/.13	gene chromosom mutat tumor genet region delet patient identifi loss analysi famili studi allel use
5	.15	.14/.17/.15	pressur renal rat sodium increas furosemid blood effect hypertens signific plasma decreas diet p intak
6	.17	.26/.13/.12	bind protein domain interact structur repeat activ residu region site two function peptid complex contain
7	.13	.13/.14/.12	rat activ increas mice level vascular express protein effect endotheli diabet aorta product control mrna
8	.05	.03/.05/.07	muscl chang dure observ studi activ mitochondri complex respons differ occur motor increas fiber membran
9	.18	.22/.18/.15	protein gene sequenc human express dna acid isol region encod virus transcript contain two strain
10	.10	.14/.10/.07	patient injuri trauma sever result day fractur score care hospit injur conclus admiss hour multipl
11	.12	.13/.12/.12	patient result use procedur anastomosi urinari sunscreen pouch bladder protect skin ileal function complic contin
12	.11	.14/.10/.10	effect inhibit rat activ relax ca k contract induc increas respons concentr phosphoryl cl aorta
13	.16	.20/.15/.14	acid increas effect fatti concentr group level diet enzym decreas signific activ product lipid dietari
14	.03	.00/.04/.05	patient group p signific age studi year differ risk associ factor result compar conclus n
15	.26	.38/.24/.16	hiv infect virus viral drug therapi patient treatment studi resist load test combin effect activ
16	.24	.36/.20/.14	patient seizur tempor epilepsi lobe tle later surgeri eeg result hippocamp ictal method onset left
17	.13	.14/.12/.12	neuron brain rat control loss chang increas signific epilepsi patient hippocamp activ function cell anim
18	.08	.12/.07/.07	liver hepat seal fetal male femal infant group found anim bodi speci fetus per signific
19	.10	.10/.10/.10	use test method latex studi sensit patient imag result detect specif evalu posit b assay
20	.08	.11/.07/.06	use method motion measur model result system dose time determin differ direct mean degre studi
21	.15	.15/.17/.15	express cell signal gene develop activ protein factor regul differenti growth transcript suggest role neural
22	.20	.26/.19/.15	heart cardiac patient ventricular left p cardiomyopathi myocardi function normal group signific failur lv dysfunct
23	.20	.24/.18/.17	cell express human receptor macrophag activ level studi tissu antibodi respons increas cultur apoptosi antigen
24	.23	.34/.19/.15	aortic patient arteri aorta arch thorac graft oper repair left descend valv use aneurysm replac
25	.19	.20/.21/.16	uterin leiomyoma women patient pregnanc treatment hysterectomi fibroid result myoma embol conclus month endometri studi

Table 4.6: The topics and nPMI scores per topic identified by $\text{copLDA}_{\text{np}}$ on the PubMed dataset when trained with $K = 25$. We words of the topics are ordered from left to right by their per-word topic probabilities. We report the nPMI scores of each topic for the top-N words ($nPMI_N$), with $N \in \{5, 10, 15\}$, as well as the average nPMI as suggested by [102].

Extrinsic evaluation: text classification Document classification is a supervised learning task where a document is associated with one or more categories from a pool of M categories $\mathcal{Y} = \{y_1, \ldots, y_M\}$. The document is represented as a vector $\mathcal{X}$. To perform the classification task, a learner such as a Logistic Regression [121] or Support Vector Machines [39] is trained to uncover a function $f : X \to \mathcal{Y}$. In the framework of topic modeling, each document can be represented as by its topic distribution. Then, given the document topic distributions a learner can be trained in order to perform the classification task and assign the categories to the documents.

To this end, we obtain the per-document topic distributions following a protocol similar to the calculations of nPMI: we train the topic models for 250 Gibbs sampling iterations. After the 150-*th* iteration (including the 150-*th*) we sample the per-document topic distributions every 25 iterations. The per-document topic distributions used as inputs to the SVMs is then the average of the 5 samples. This approach of subsampling the chain is called thinning and is common with Gibbs sampling. Gibbs samplers generate a Markov chain of samples where nearby samples are correlated and subsampling the chain of samples results in obtaining samples (here document distributions) that are less correlated and therefore more effective. Thinning allows to have independent samples. In addition, discarding the samples from the beginning of the chain is called burn-in, and we applied it to the first 149 samples, as they may not accurately represent the desired distribution.

We evaluate the presented topic models using the task of document classification. We use Support Vectors Machines (SVMs) as our learners, and in particular the implementation of Scikit-learn [143]. We set the regularization parameter $C = 10$, after cross-validating it in the training parts of the datasets from the range $C \in \{10^{-4}, 10^{-3}, \ldots, 10^{4}, \}$. For the multi-label datasets (TED, Reuters and PubMed) we employed one-versus-rest: the SVMs return every category with a positive distance from the separating hyper-planes. Table 4.7 reports the classification scores for the micro-averaged F_1 measure for the datasets used. The F_1 measure is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where precision is the fraction between true positives and the sum of true positives and false positives, while recall is the fraction between true positives and true positives and false negatives. Micro-averaging refers to summing up the individual true positives, false positives, and false negatives of the system for different

categories and aggregating them in order to get the total precision and recall. The reported scores in Table 4.7 are the averages of 10-fold cross-validation and the corresponding standard deviations are shown as subscripts.

As one can note, the highest classification scores are obtained using topic distributions obtained with copulas. Importantly, the addition of copulas consistently improves the scores of the MiF_1 evaluation measure. Notice the benefits of copulas in the table: the systems that use copulas when sampling the topics of the words of the coherent segments consistently achieve higher scores than their counterparts that do not use copulas. Importantly, those systems perform better that LDA, which builds on the bag-of-words representation. The latter highlights that incorporating parts of text structure is advantageous for applications that use document representations learned with topic models. Overall, the results suggest that assuming short segments like noun-phrases, bigrams and trigrams to be coherent is the optimal option with respect to the performance.

Our last observation concerns the low performance of the SVMs on the TED dataset. We believe that this is due to the fact that TED is a multilabel dataset with 2.42 categories/instance (Table 4.1), which is much higher that the rest of the multilabel dataset. The results suggest that an approach like one-versus-rest we used, can not model this successfully and one should employ a better suited multi-label strategy. Although interesting, multi-label classification was not the main topic of this chapter and therefore we do not explore more this direction.

4.5 Summary

In this chapter we presented *segmentLDA* and *copulaLDA*, two novel topic models that incorporate prior knowledge of text structure in the form of boundaries of coherent segments. *copulaLDA* uses copulas when sampling the topics of the words of segments and thus allows few, related topics to appear in a segment. On the other hand, *segmentLDA* applies a maximal binding between the topics of the words in a segment and, as a result, a single topic is assumed to generate its words.

To evaluate different versions of the proposed model, where we considered various types of segments to be thematically coherent, we performed a systematic comparison of the performance of the models: we compared their performance with respect to the generalization performance measured by perplexity, topical coherence measured by normalized pointwise mutual information (nPMI) and text

	K	LDA	segLDA$_{np}$	copLDA$_{np}$	segLDA$_{sent}$	copLDA$_{sent}$	segLDA$_{bi}$	copLDA$_{bi}$	segLDA$_{tri}$	copLDA$_{tri}$
20NG	25	$55.34_{\pm 1.12}$	$54.90_{\pm 0.65}$	$\mathbf{57.30}_{\pm 0.81}$	$48.95_{\pm 1.05}$	$56.80_{\pm 1.09}$	$54.25_{\pm 0.78}$	$56.69_{\pm 0.67}$	$55.19_{\pm 0.95}$	$56.56_{\pm 0.47}$
	50	$59.83_{\pm 0.81}$	$59.37_{\pm 0.97}$	$62.36_{\pm 0.92}$	$55.27_{\pm 1.31}$	$61.66_{\pm 0.97}$	$\mathbf{64.09}_{\pm 1.00}$	$63.71_{\pm 0.65}$	$63.76_{\pm 0.96}$	$63.79_{\pm 0.77}$
	75	$64.05_{\pm 1.07}$	$62.98_{\pm 1.26}$	$63.42_{\pm 1.06}$	$56.18_{\pm 1.00}$	$63.48_{\pm 1.29}$	$63.08_{\pm 1.20}$	$63.65_{\pm 1.17}$	$64.12_{\pm 1.36}$	$\mathbf{64.25}_{\pm 0.87}$
	100	$65.12_{\pm 1.24}$	$61.42_{\pm 0.84}$	$63.86_{\pm 1.01}$	$59.71_{\pm 1.14}$	$64.02_{\pm 1.01}$	$63.82_{\pm 0.91}$	$\mathbf{65.20}_{\pm 1.08}$	$64.78_{\pm 1.01}$	$64.88_{\pm 0.98}$
	150	$64.62_{\pm 1.15}$	$62.29_{\pm 0.95}$	$\mathbf{65.11}_{\pm 0.86}$	$56.89_{\pm 1.02}$	$63.74_{\pm 0.71}$	$64.87_{\pm 0.82}$	$64.18_{\pm 1.17}$	$64.74_{\pm 1.14}$	$64.23_{\pm 1.01}$
Wiki15	25	$72.99_{\pm 2.34}$	$68.44_{\pm 4.16}$	$\mathbf{73.66}_{\pm 2.39}$	$64.66_{\pm 2.90}$	$69.78_{\pm 3.89}$	$69.31_{\pm 2.54}$	$72.84_{\pm 2.48}$	$68.45_{\pm 1.60}$	$72.53_{\pm 3.20}$
	50	$76.61_{\pm 3.43}$	$69.64_{\pm 2.70}$	$\mathbf{77.04}_{\pm 3.50}$	$63.73_{\pm 3.57}$	$73.49_{\pm 2.21}$	$74.14_{\pm 4.30}$	$74.23_{\pm 2.72}$	$73.92_{\pm 4.29}$	$74.23_{\pm 2.42}$
	75	$76.01_{\pm 2.13}$	$72.27_{\pm 3.39}$	$74.42_{\pm 3.25}$	$65.45_{\pm 3.78}$	$73.47_{\pm 2.55}$	$74.45_{\pm 2.62}$	$\mathbf{76.36}_{\pm 2.73}$	$73.61_{\pm 2.75}$	$75.85_{\pm 2.63}$
	100	$76.01_{\pm 2.47}$	$71.68_{\pm 3.88}$	$73.81_{\pm 3.03}$	$67.37_{\pm 5.28}$	$72.30_{\pm 3.85}$	$75.86_{\pm 2.67}$	$\mathbf{76.25}_{\pm 2.50}$	$76.13_{\pm 3.81}$	$73.04_{\pm 4.12}$
	150	$73.96_{\pm 2.81}$	$71.56_{\pm 3.75}$	$72.60_{\pm 2.66}$	$66.11_{\pm 2.48}$	$72.82_{\pm 1.62}$	$74.90_{\pm 1.38}$	$\mathbf{75.75}_{\pm 2.43}$	$75.22_{\pm 2.44}$	$75.66_{\pm 3.08}$
Wiki37	25	$59.65_{\pm 2.86}$	$55.83_{\pm 3.32}$	$\mathbf{59.98}_{\pm 1.87}$	$55.21_{\pm 2.07}$	$59.41_{\pm 1.93}$	$56.34_{\pm 2.18}$	$58.51_{\pm 1.51}$	$57.72_{\pm 3.25}$	$57.78_{\pm 1.68}$
	50	$60.47_{\pm 3.02}$	$59.18_{\pm 1.70}$	$\mathbf{63.95}_{\pm 2.11}$	$55.69_{\pm 1.54}$	$63.54_{\pm 1.98}$	$62.56_{\pm 1.84}$	$63.34_{\pm 2.48}$	$62.31_{\pm 1.92}$	$63.35_{\pm 2.29}$
	75	$63.27_{\pm 2.22}$	$60.37_{\pm 2.13}$	$63.74_{\pm 2.19}$	$56.44_{\pm 3.06}$	$63.84_{\pm 1.78}$	$62.88_{\pm 1.87}$	$62.34_{\pm 2.59}$	$\mathbf{64.34}_{\pm 2.06}$	$63.57_{\pm 2.65}$
	100	$63.94_{\pm 2.64}$	$60.76_{\pm 3.01}$	$62.52_{\pm 1.38}$	$56.05_{\pm 2.49}$	$63.51_{\pm 1.58}$	$61.97_{\pm 1.76}$	$\mathbf{64.50}_{\pm 2.18}$	$63.32_{\pm 2.22}$	$64.22_{\pm 2.15}$
	150	$63.75_{\pm 2.39}$	$61.68_{\pm 1.62}$	$62.18_{\pm 2.93}$	$56.93_{\pm 2.70}$	$62.04_{\pm 2.07}$	$64.31_{\pm 2.23}$	$\mathbf{64.45}_{\pm 2.30}$	$64.38_{\pm 2.58}$	$64.02_{\pm 2.96}$
Wiki46	25	$54.77_{\pm 1.75}$	$49.75_{\pm 0.86}$	$54.59_{\pm 2.31}$	$47.52_{\pm 1.87}$	$52.52_{\pm 1.44}$	$53.01_{\pm 2.34}$	$53.77_{\pm 1.82}$	$55.11_{\pm 1.41}$	$\mathbf{56.14}_{\pm 1.69}$
	50	$60.78_{\pm 1.90}$	$54.96_{\pm 1.33}$	$61.32_{\pm 1.86}$	$50.51_{\pm 2.06}$	$58.64_{\pm 1.58}$	$58.69_{\pm 1.98}$	$60.50_{\pm 1.12}$	$60.16_{\pm 1.70}$	$\mathbf{61.68}_{\pm 1.64}$
	75	$62.35_{\pm 1.61}$	$60.25_{\pm 1.58}$	$62.89_{\pm 1.25}$	$52.19_{\pm 1.46}$	$62.71_{\pm 1.30}$	$62.95_{\pm 2.43}$	$63.69_{\pm 1.90}$	$63.05_{\pm 2.11}$	$\mathbf{63.26}_{\pm 1.99}$
	100	$64.81_{\pm 1.29}$	$60.49_{\pm 1.63}$	$\mathbf{65.22}_{\pm 1.50}$	$53.98_{\pm 2.22}$	$62.03_{\pm 1.91}$	$64.18_{\pm 1.67}$	$62.77_{\pm 1.54}$	$63.97_{\pm 1.50}$	$62.98_{\pm 1.79}$
	150	$65.36_{\pm 1.43}$	$63.68_{\pm 2.17}$	$65.82_{\pm 1.11}$	$55.03_{\pm 1.43}$	$61.38_{\pm 1.39}$	$67.30_{\pm 1.67}$	$65.41_{\pm 1.91}$	$64.89_{\pm 0.93}$	$\mathbf{67.97}_{\pm 1.68}$
PubMed	25	$43.18_{\pm 1.15}$	$42.80_{\pm 1.66}$	$44.74_{\pm 1.17}$	$41.00_{\pm 2.12}$	$49.47_{\pm 1.84}$	$46.13_{\pm 1.69}$	$49.62_{\pm 1.41}$	$49.55_{\pm 0.99}$	$\mathbf{51.62}_{\pm 1.24}$
	50	$57.59_{\pm 1.97}$	$53.80_{\pm 1.57}$	$55.26_{\pm 1.96}$	$53.86_{\pm 2.36}$	$57.37_{\pm 1.60}$	$54.91_{\pm 1.95}$	$56.95_{\pm 1.40}$	$57.58_{\pm 1.80}$	$\mathbf{58.15}_{\pm 1.45}$
	75	$64.39_{\pm 1.35}$	$61.26_{\pm 2.13}$	$63.53_{\pm 1.62}$	$52.64_{\pm 2.02}$	$63.55_{\pm 2.00}$	$64.42_{\pm 1.33}$	$61.20_{\pm 2.00}$	$\mathbf{64.53}_{\pm 1.77}$	$62.28_{\pm 2.18}$
	100	$65.50_{\pm 0.88}$	$62.81_{\pm 1.99}$	$65.75_{\pm 2.14}$	$59.30_{\pm 1.96}$	$66.02_{\pm 1.94}$	$65.50_{\pm 1.34}$	$64.15_{\pm 2.12}$	$\mathbf{66.39}_{\pm 1.55}$	$65.22_{\pm 1.69}$
	150	$68.22_{\pm 2.36}$	$65.37_{\pm 1.75}$	$68.30_{\pm 1.48}$	$56.58_{\pm 2.28}$	$67.95_{\pm 2.16}$	$67.81_{\pm 1.23}$	$68.32_{\pm 1.63}$	$\mathbf{68.94}_{\pm 1.59}$	$67.97_{\pm 1.92}$
Reuters	25	$62.92_{\pm 3.77}$	$62.15_{\pm 4.06}$	$62.38_{\pm 4.20}$	$61.19_{\pm 4.17}$	$62.30_{\pm 3.88}$	$61.49_{\pm 4.27}$	$63.34_{\pm 4.60}$	$62.93_{\pm 3.97}$	$\mathbf{65.04}_{\pm 4.01}$
	50	$66.34_{\pm 4.21}$	$65.09_{\pm 4.19}$	$65.42_{\pm 4.23}$	$65.33_{\pm 4.30}$	$65.04_{\pm 4.29}$	$66.92_{\pm 3.71}$	$66.98_{\pm 3.39}$	$67.76_{\pm 3.73}$	$\mathbf{67.23}_{\pm 3.65}$
	75	$68.56_{\pm 3.81}$	$66.25_{\pm 3.88}$	$67.85_{\pm 3.30}$	$66.78_{\pm 3.96}$	$66.39_{\pm 3.44}$	$67.88_{\pm 3.84}$	$68.67_{\pm 3.72}$	$\mathbf{69.23}_{\pm 3.69}$	$68.47_{\pm 3.79}$
	100	$70.77_{\pm 3.47}$	$68.46_{\pm 3.53}$	$69.97_{\pm 3.54}$	$66.19_{\pm 4.17}$	$67.52_{\pm 4.11}$	$69.47_{\pm 3.33}$	$70.25_{\pm 3.82}$	$70.15_{\pm 3.25}$	$\mathbf{70.89}_{\pm 3.74}$
	150	$70.95_{\pm 3.17}$	$70.72_{\pm 3.23}$	$71.34_{\pm 3.52}$	$69.00_{\pm 3.67}$	$69.66_{\pm 3.32}$	$71.38_{\pm 3.34}$	$70.41_{\pm 3.44}$	$\mathbf{72.60}_{\pm 3.07}$	$71.71_{\pm 3.57}$
Ted	25	$11.74_{\pm 3.12}$	$10.83_{\pm 2.98}$	$12.02_{\pm 2.30}$	$10.09_{\pm 2.50}$	$12.48_{\pm 3.39}$	$10.73_{\pm 2.72}$	$\mathbf{12.84}_{\pm 2.32}$	$11.10_{\pm 3.22}$	$12.11_{\pm 2.59}$
	50	$13.58_{\pm 1.35}$	$12.20_{\pm 3.62}$	$15.14_{\pm 3.41}$	$11.93_{\pm 2.72}$	$13.12_{\pm 2.84}$	$12.48_{\pm 2.18}$	$\mathbf{12.84}_{\pm 2.13}$	$13.12_{\pm 2.69}$	$12.20_{\pm 2.01}$
	75	$13.58_{\pm 1.52}$	$11.47_{\pm 3.26}$	$13.21_{\pm 2.43}$	$12.57_{\pm 2.96}$	$\mathbf{14.59}_{\pm 3.03}$	$13.39_{\pm 4.21}$	$13.39_{\pm 2.91}$	$12.75_{\pm 3.44}$	$13.67_{\pm 3.27}$
	100	$12.66_{\pm 2.65}$	$12.94_{\pm 2.77}$	$12.20_{\pm 2.43}$	$10.64_{\pm 3.51}$	$\mathbf{13.67}_{\pm 2.77}$	$13.67_{\pm 2.61}$	$12.66_{\pm 2.12}$	$12.39_{\pm 2.60}$	$12.48_{\pm 2.06}$
	150	$12.48_{\pm 2.60}$	$12.57_{\pm 2.36}$	$12.20_{\pm 3.21}$	$10.55_{\pm 2.10}$	$13.21_{\pm 2.60}$	$12.57_{\pm 1.79}$	$13.21_{\pm 2.82}$	$13.03_{\pm 2.75}$	$\mathbf{13.49}_{\pm 3.02}$

Table 4.7: MiF scores for the classification task. The best scores are shown in bold font.

classification performance measured by the MiF_1 scores. Our results strongly suggest that prior knowledge of text structure benefits the coherence of the produced topics as well as the quality of the learned per-document topic distributions. Our analysis further suggests that, in practice, assuming short text spans like bigrams and trigrams is optimal: apart from the fact that such segments achieved the best performance in most of the evaluation tasks, to obtain them one does not need resort to linguistic tools like parsers but can rely only on counting operations. Lastly, the use of copulas is advised mostly for cases where the output of the topic models will be used for a real task, like text classification, which is though the most interesting ones.

While incorporating segments with or without the use of copulas has been shown to be beneficial, one should take into account that such methods require an overhead for segmenting the documents. While the segmentation methods we presented can benefit from parallelization in a straight-forward way, this needs to be considered. Further, sampling from copulas imposes a further overhead: not only one needs sample from complex distributions [83, 82], but also needs to transform the sample using the probabilistic integral transform as discussed in the previous sections. In conjunction with the extra free parameters (copulas family, λ, segmentation methods) that these models introduce, one may experience a significant overhead tuning the models for production purposes.

Our findings open various avenues for future research. The computational overhead discussed in the previous paragraph motivates future work on accelerating inference such as combining copulas with variational inference, which to the best of our knowledge has yet to be achieved. A second question that is raised is whether one can use these findings to improve multilingual topic models. This question in fact motivates part of the contributions concerning text structure and bilingual topic models presented in the next chapter. Another question concerns the segmentation approach used. While here we relied on frequency-based approaches like n-grams and syntactic information like noun-phrases, one may ask if similar results can be obtained using an unsupervised segmentation approaches or approaches that learn the segment boundaries and the topics jointly.

Chapter 5

Extending Bilingual Topic Models

Due to the ever increasing amount of multilingual content online, people are more and more confronted with documents available in more than one language. An important challenge when developing systems for such *multilingual* document corpora is to automatically discover and extract meaningful topics that help to better organize them and comprehend their content. Following the success of topic models in the monolingual setting, there have been recent efforts that extended them to the bilingual (or multilingual) setting. Those extensions of topic models that account for text written in two or more languages enable a variety of interesting multilingual and cross-lingual applications.

In the previous chapter we discussed that probabilistic topic models like Latent Dirichlet Allocation (LDA) [23] are a family of unsupervised models that when applied to monolingual collections uncover the latent themes underlying it. Despite their success and wide adoption, we identified some of their limitations and suggested extensions to overcome them. In particular, we argued that while the bag-of-words assumption may be important during inference as it simplifies the calculation of the conditional probabilities, a more complex document model may be advantageous. Our contributions were motivated by the outcomes of linguistic or statistic pre-processing steps for text segmentation. We proposed two novel topic models that incorporate parts of the text structure in the form of boundaries of text spans and we demonstrated in a plethora of evaluation tasks the improvements that those models yield.

Our focus in this chapter is bilingual topic models. The most representative bilingual topic model is illustrated in Figure 5.2(i) and is commonly called bilingual[1] LDA (BiLDA) [127, 45, 178]. BiLDA, which as also presented in Section 2.3.2,

[1]Depending on the number of input languages the model may be referred to as either bilingual

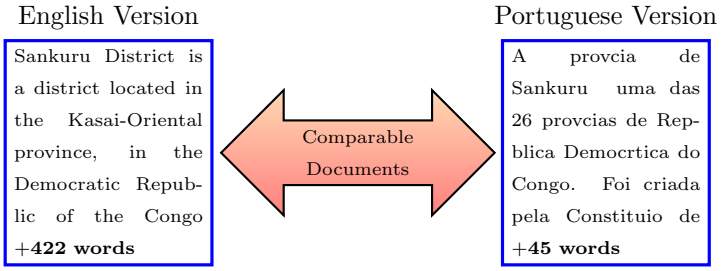


Figure 5.1: Motivating Example: excerpts from comparable Wikipedia documents. The English version is several times bigger than the Portuguese and one may reasonably assume it covers more topics.

extends LDA and does not require any prior, language dependent linguistic knowledge but the input collection to be in the form of pairs of thematically aligned documents. Given the pairs, a fundamental hypothesis of BiLDA is that the documents of a pair share a single per-document topic distribution θ, an observation we also highlighted in Section 2.3.2. This entails that the documents in a pair discuss exactly the same themes. Although reasonable for parallel corpora, whose pairs consist of documents that are translations, this assumption is strong for collections composed by pairs of *comparable* documents (e.g. [122]), that is documents similar to some extent only. Figure 5.1 illustrates an example of comparable documents written in English and Portuguese.[2] As the English document is larger one would expect it to cover more topics. Hence, having a shared topic distribution between those two documents is a strong assumption.

In this chapter we propose to extend bilingual topic models. On one hand, our goal is to relax the assumption of comparable documents sharing a single topic distribution and better adapt bilingual topic models for corpora consisting of documents like those of Figure 5.1. For this purpose, instead of a shared distribution we allow the documents of a pair to have two, separate, yet *bound* distributions. We suggest that the strength of the bound should depend on the semantic similarity of the documents of the pair. The estimation of this similarity for documents written in different languages is a task on itself. Instead of using dictionaries, which are one-to-one or one-to-N discrete word associations and do not capture different levels of similarity, or machine translation systems, which are computationally expensive to develop, we propose to use cross-lingual word embeddings.

On the other hand, motivated by our discussion in Chapter 4, where we found

or multilingual LDA.

[2]This is a real example from the $\text{Wiki}_{\text{En-Pt}}$ collection used in our experiments.

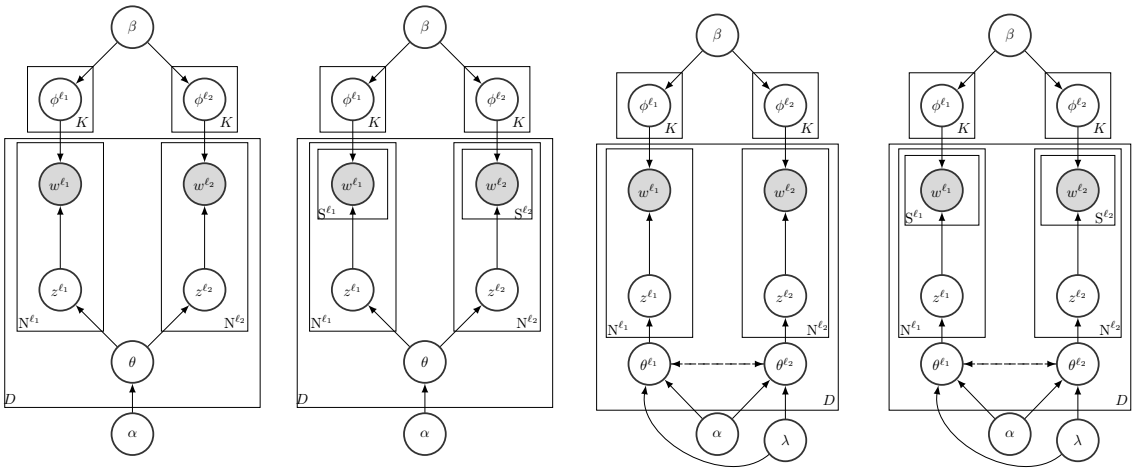


Figure 5.2: The topic models used in this work. From left to right: (i) BiLDA (ii) segBiLDA (iii) λ-BiLDA (iv) λ-segBiLDA. The difference of BiLDA and segBiLDA from their $\lambda-$ counterparts lies on the fact that the second have separate but bound topic distributions and the strength of binding is controlled by λ.

text structure to benefit topic models, we aim at validating our findings in the multilingual setting. The arguments that support incorporating text structure in topic models hold independently of the language: words that frequently co-occur (*e.g.*, frequent n-grams) or words that are grouped together due to the syntax (*e.g.*, noun-phrases) should be topically coherent in every language.

The questions we attempt to answer here are twofold: ($Q1$) How to better adapt bilingual topic models to comparable collections? ($Q2$) Does this adaptation generalize well across different types of topic models? To address these questions, the chapter proposes three contributions:

(i.) a novel approach that combines topic models with (shallow) neural networks for learning word embeddings allowing the former to extract latent distributions from comparable corpora,

(ii.) the extension of BiLDA and of a monolingual topic model that incorporates text structure in the form of boundaries of coherent text spans,

(iii.) a systematic evaluation of the novel topic models on four comparable corpora where English are paired with French, German, Italian, Spanish and Portuguese documents.

The remainder of the chapter is organized as follows: Section 5.1 presents an overview of the related work. The main contribution of this chapter is presented

in Section 5.2, where we propose to incorporate information of text structure into bilingual topic models and also extend them for comparable corpora. Then, in Section 5.3 we evaluate the presented topic models and we conclude in Section 5.4.

5.1 An overview of the relevant literature

Our work lies in the intersection of the fields of multilingual topic modeling and cross-lingual word embeddings. We review the relevant literature starting with the work on multilingual topic models.

Multilingual Topic Models There are two different lines of research in the multilingual topic modeling approaches with respect to the nature of the available training inputs. The first line assumes access or attempts to create linguistic resources such as dictionaries, in order to identify the topical links and alignments between the multilingual documents of a text corpus [27, 88, 192, 28]. The topic alignments between documents are not implicit in the input, and the models identify the topically relevant multilingual documents and the topic distributions while leveraging the available linguistic resources. For instance, [27] propose the multilingual topic model for unaligned text (MuTo) that discovers a parallelism in the documents of the corpus at the vocabulary level, while it assumes that similar themes are expressed in both languages. To perform the joint task of producing consistent topics in each of two languages and then aligning them, the model uses dictionaries. *JointLDA* is a model with similar motivations, proposed in [88]. To cope with the multilingual setting, *jointLDA* also uses dictionaries but learns topics shared among the input languages. Those topics are distributions over the vocabulary terms of the multilingual corpus, and as a result, terms of different languages occur in a topic. Despite the advantages of such models, their requirements for several language-specific resources can be seen as a limitation. Also, [28] uses multilingual topic models and incorporates a regression task like sentiment prediction to better predict sentiment.

The second, more flexible line of research, investigates topic modeling solely on the basis of the textual inputs. Those inputs, usually consist of text corpora with documents that are either parallel translations [196] or comparable translations of each other [137, 127, 44, 149]. Such topic models by not relying on any external resource are a better fit for unsupervised methods. The most representative model of

this family is BiLDA, which extends LDA in the bilingual [137, 127, 44] or the multilingual setting [98, 127]. The difference between the bi- or multilingual setting lies on the number of the input languages, which ranges from two to several. The model we propose in this work belongs to this family of models as it assumes access to a corpus whose input documents form theme-aligned pairs. However, our model is more flexible as instead of assuming a single topic distribution per pair of documents uses two topic distributions that are linked with a binding mechanism that uses cross-lingual word embeddings to account for the level of similarity between the documents. Similarly, Latent Semantic Allocation has been extended in the bilingual setting [169] with applications to translation or [**?**] proposed to use a PLSA model per-language and concatenate the learned representations; we focus however on BiLDA as in the monolingual setting LDA was shown to outperform LSA [23] and one should expect similar findings in the bilingual setting.

Cross-lingual Word Embeddings According to the *distributional hypothesis*, first stated in the early 60's [58, 77], linguistic items such as words with similar distributions should have similar meanings. In other words, semantically similar words should have similar contextual distributions. The contextual information is usually induced assuming the context to be documents or sliding windows and is represented by populating word-context co-occurrence matrices. For words, different models that learn distributed representations have been recently proposed and those models are used as implementation models of the distributional hypothesis [175]. To this end, the *distributed representations* (also known as word embeddings) associate words with dense vectors, of dimension of a few hundreds to some thousands. A distributed representation of a symbol is a vector of features that characterize the meaning of the symbol and in our case a symbol is a word. The representation is a continuous D-dimensional vector and, therefore, compact in the sense that an exponential number of symbols (words) in the number of dimensions can be efficiently represented [175], compared, for instance, to the one-hot-encoding scheme that can only represent D symbols when using D dimensions.

Among different models, the skipgram model with negative sampling [124], has been shown to be efficient and effective in several applications. Such a model is a function f that projects a word w in a D-dimensional space: $f(w) \in \mathbb{R}^D$, where D is predefined. Although the model relies on a well-defined prediction task [15], it has been shown that it is implicitly factorizing a word-context matrix, whose

cells are the pointwise mutual information (*PMI*) (Chap. 4, Eq. 4.10) of the respective word and context pairs, shifted by a global constant [108, 109]. Despite the theoretical equivalence however, an advantage of the skipgram model compared to other models that factorize such matrices like Latent Semantic Indexing [84] is its ability to practically scale to huge amounts of data.

The skipgram model was initially proposed for the single language setting. However, motivated by the idea of having a single representation space shared by more languages, cross-lingual word embeddings models extended the idea in the bilingual and multilingual settings. The models can be grouped with regard to the approach used to align the cross-lingual embeddings. Models like [125] followed by [190, 106] for instance, begin by learning monolingual word embeddings and try to learn a linear transformation from one space to the other. Another way to learn cross-lingual embeddings [72, 50, 179] is by artificially generating multilingual documents by concatenating the documents of parallel or comparable corpora and then training existing monolingual models. Lastly, models like [118, 71] perform joint optimization of monolingual and cross-lingual losses. They can benefit from very big monolingual corpora for optimizing their monolingual objectives while relying on smaller corpora for optimizing their cross-lingual objective.

In this work we use Bilbowa [71]. It belongs to the family of models that jointly optimize monolingual and cross-lingual objectives. It extends the skipgram model for cross-lingual embeddings and trains directly on monolingual data. It uses a bilingual signal from a smaller set of raw-text sentence-aligned data to align the cross-lingual embeddings.

Combining Topic Models and Word Embeddings While topic models are trained to infer the per-word and per-document topic distributions, the skipgram model is trained by trying to predict the context of a word. Different efforts have attempted to extend the models by combining them. For instance, embeddings associate words with a single vector, which may be limiting for encoding the different meanings of polysemous words. This limitation motivated works that extend word embeddings with topic models. The purpose is for the topic models to uncover the different senses of a word, so that different embeddings can be derived for each sense [116, 33]. Such efforts attempt to produce better performing word embeddings.

In a relevant line of work, the purpose is to produce better topic models while taking advantage of the fact that text embeddings model semantic similarity. To

ℓ_1, ℓ_2	Input languages e.g., ℓ_1=English, ℓ_2=German
$d_i, d_i^{\ell_1}, d_i^{\ell_2}$	doc. pair d_i, whose aligned docs. are $d_i^{\ell_1}$, $d_i^{\ell_2}$
λ_i	The semantic similarity between $d_i^{\ell_1}$ and $d_i^{\ell_2}$
V_ℓ	The size of the vocabulary in language ℓ
$\Omega_{k,i}$	The number of words in d_i assigned to topic k
θ_i	Topic distribution of d_i
$\Psi_{k,w}$	The number of assignments of word w to topic k
ϕ_k	Word distribution for topic k
$s_{i,j}$	The j^{th} segment of document d_i
$w_{i,j,k}$	The k^{th} word of segment $s_{i,j}$
N_i	The number of words of document d_i
$N_{i,j}$	The number of words in segment $s_{i,j}$
$N_{i,j,w}$	The number of occurrences of word w in $s_{i,j}$

Table 5.1: The notation used for the development of the topic models. Adding exponents ℓ_1, ℓ_2 to the symbols of the lower part of the table (below the dashed line) stands for counts specific to $d_i^{\ell_1}, d_i^{\ell_2}$.

this end, [136, 43, 193] extend topic models in order to encourage the models to group words that are *a priori* known to be semantically related into topics, where the *a priori* knowledge comes from training embeddings in large external corpora. Our work belongs to this second line of research because we use word embeddings to improve the results of topic models. Differently from previous research though, word embeddings are used only to estimate the similarity of documents written in different languages. Also, our models are multilingual, while previous work investigated the intersection of topic models and word embeddings in the monolingual setting using English.

5.2 Framework

Our primary goal in this work is to adapt the bilingual topic models for comparable corpora. To accomplish that we relax the assumption of paired documents having identical topic distributions. In the rest of this section, after presenting the notation, we briefly discuss BiLDA in Section 5.2.1. To illustrate how several classes of topic models can benefit by the adaptation to comparable corpora, we introduce a novel bilingual topic model that incorporates parts of the document's structure (Section 5.2.2). We, then, extend BiLDA and the novel bilingual topic model for comparable corpora in Section 5.2.3 and Section 5.2.4.

The notation is summarized in Table 5.1. For consistency, we keep the symbols of previous work [183] to the extent of possible. We denote by ℓ_1 and ℓ_2 the

different languages of a comparable corpus. As languages are handled symmetrically, for convenience we designate by $\ell\backslash$, the language different from language $\ell \in \{\ell_1, \ell_2\}$. The inputs of the topic models are document pairs $d_i = (d_i^{\ell_1}, d_i^{\ell_2})$, that consist of thematically aligned documents $d_i^{\ell_1}$ and $d_i^{\ell_2}$, written in ℓ_1 and ℓ_2. Depending on the model, documents are either represented as a bag-of-words, or as a bag-of-segments. Segments are text spans smaller than documents, for instance sentences, and are represented as a bag-of-words. Considering $\ell \in \{\ell_1, \ell_2\}$, $s_{i,j}^{\ell}$ is the j^{th} segment of document d_i^{ℓ}. Segmented documents have a hierarchical structure: they are composed by segments that are composed by words in turn. Depending on the model, there may exist either a single θ_i topic distribution that captures the topics present in both documents of the pair d_i, or two, separate yet *bound* topic distributions θ_i^{ℓ}, $\theta_i^{\ell\backslash}$ that capture the topics of d_i^{ℓ} and $d_i^{\ell\backslash}$ respectively. The rest of the notation in Table 5.1 stands for count matrices or vectors used during inference.

5.2.1 The bilingual LDA

`BiLDA` (Figure 5.2(i)) is a direct adaptation of `LDA` in the bilingual setting where a parallel collection is assumed to be the model's input. Due to its effectiveness we use it as a reference in this work. `BiLDA` assumes that the documents of an aligned pair d_i have identical topic distributions (a single and shared θ_i) and therefore discuss the same topics. Also, it expects the documents as a bag-of-words. Its generative story is as follows:

- for each topic $k \in [1, K]$: $\phi_k^{\ell_1} \sim Dir(\beta)$, $\phi_k^{\ell_2} \sim Dir(\beta)$
- for each document pair d_i:
  - sample $\theta_i \sim Dir(\alpha)$
  - for each language $\ell \in \{\ell_1, \ell_2\}$
    - for each of the N_i^{ℓ} words:
      - sample $z \sim Mult(1, \theta_i)$
      - sample $w \sim Mult(1, \phi_z^{\ell})$

The collapsed Gibbs sampling updates [178] for the topic of word j of document d_i is $\forall \ell \in \{\ell_1, \ell_2\}$:

$$P\Big(z_{ij}^{\ell} = z_k | \boldsymbol{z}_{\neg ij}^{\ell}, \boldsymbol{z}^{\ell\backslash}, \boldsymbol{w}^{\ell}, \boldsymbol{w}^{\ell\backslash}, \alpha, \beta, \Big) \propto \frac{\Psi_{k,w,\neg ij}^{\ell} + \beta}{\Psi_{k,\cdot,\neg ij}^{\ell} + V_{\ell}\beta} \Big(\Omega_{i,k,\neg ij} + \alpha\Big)$$

A dot "$\cdot$" occurring in the subscript of a count variable, stands for the summation over the possible values of the element it replaces, i.e.,$\Psi^{\ell}_{k,\cdot,\neg ij} = \sum_{w=1}^{V} \Psi^{\ell}_{k,w,\neg ij}$. Also, $\neg$ excludes the counts for the particular variable ($\neg ij$ excludes the counts of the j-th word of d^{ℓ}_i). Further, $Dir(\alpha)$ stands for a sample from a Dirichlet distribution with prior α and $Mult(M, \theta)$ stands for M samples from a Multinomial distribution parametrized by θ.

For BiLDA, as well as for the models we present next, we consider the Dirichlet hyperparameters $\alpha \in \mathbb{R}^K$ and $\beta \in \mathbb{R}^V$ to have fixed values, implying symmetric priors. Extending the models to asymmetric priors or even learning their values could be done as in [6] for example. Also, as commonly done we omit from the generative stories the steps where the sizes of segments or documents are sampled as their sizes are observed during inference. As noted, BiLDA uses a bag-of-words representation; next we present an extension that uses a more complex document representation.

5.2.2 Incorporating text structure into bilingual topic models

In this section, we propose segment-BiLDA (segBiLDA) that incorporates prior knowledge of text structure using a more complex document representation than bag-of-words. Although important for inference, the bag-of-words assumption is limiting. In fact, previous research in the single language domain showed the benefits of similar extensions: Wang et al. [182] proposed a model that handles bigrams as a single token or as two unigrams depending on the topic, Lau et al. [103] modeled frequent bigrams as separate tokens, Balikas et al. [11] proposed to incorporate sentence boundaries to LDA, while Boyd et al. [29] incorporated parse trees . These important contributions focused on the monolingual setting and used English texts for empirical evaluation. Here, we extend topic models to account for text structure in the bilingual case.

For our subsequent analysis, we define coherent text segments to be contiguous words of a document that are topically coherent. A topically coherent text segment refers to a segment whose constituent words discuss a single or very few related themes. For instance, one would expect frequent bigrams like "information retrieval" or even short sentences to be topically coherent as they generally convey a simple message. We model this property with (segBiLDA), which is illustrated in Figure 5.2(ii). segBiLDA assumes that the input text is segmented *a priori* and incorporates the boundaries of segments in its generative story:

- for each topic $k \in [1, K]$: $\phi_k^{\ell_1} \sim Dir(\beta)$, $\phi_k^{\ell_2} \sim Dir(\beta)$
- for each document pair d_i:
  - sample $\theta_i \sim Dir(\alpha)$
  - for each language $\ell \in \{\ell_1, \ell_2\}$
    - for the j segment ($1 \leq j \leq S^{\ell}$):
      - sample $z \sim Mult(1, \theta_i)$
      - sample segment words: $(w_1 \dots w_{N_{i,j}^{\ell}}) \sim Mult(N_{i,j}^{\ell}, \phi_z^{\ell})$

The important difference of BiLDA from segBiLDA (Figures 5.2(i) and 5.2(ii)) lies in the addition of the segment's plate. A topic is sampled per segment, and every word of a segment is associated with it. The segment boundaries limit the number of topics that appear in the segment to be equal to one. As in BiLDA though, words remain the document units and this single topic is associated with each word of the segment. Therefore, comparing the models on measures like perplexity that are calculated at the word level is fair. To infer these topics we propose a collapsed Gibbs sampling approach, that $\forall \ell \in \{\ell_1, \ell_2\}$, samples topics from:

When sampling the topic $z_{i,j}^{\ell_1}$ of the words of the segment $s_{i,j}^{\ell_1}$ one has:

$$
\begin{aligned}
&\text{sample } z_{i,j}^{\ell_1} \sim p\Big(z_{i,j}^{\ell_1} = z_k | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{z}^{\ell_2}, \dots, \boldsymbol{z}^{\ell_M}, \boldsymbol{w}^{\ell_1}, \dots, \boldsymbol{w}^{\ell_M}, \alpha, \beta\Big) \\
&\propto \iint_{\theta\ \phi} p(z_{i,j}^{\ell_1} = z_k | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{z}^{\ell_2}, \dots, \boldsymbol{z}^{\ell_M}, \theta, \alpha) \times \\
&\quad p(\boldsymbol{w}_{s_{i,j}}^{\ell_1} | z_{i,j}^{\ell_1} = z_k, \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{w}_{\neg s_{i,j}}^{\ell_1}, \phi, \beta) d\phi d\theta \\
&\propto \int_{\theta} p(z_{i,j}^{\ell_1} = z_k | \theta) p(\theta | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{z}^{\ell_2}, \dots, \boldsymbol{z}^{\ell_M}, \alpha) d\theta \times \\
&\quad \int_{\phi} \prod_{w \in s_{i,j}^{\ell_1}} p(w | z_{s_{i,j}}^{\ell_1} = z_k, \phi) p(\phi | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{w}_{\neg s_{i,j}}^{\ell_1}, \beta) d\phi \\
&= \int_{\theta_i} p(z_{i,j}^{\ell_1} = z_k | \theta) p(\theta | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{z}^{\ell_2}, \dots, \boldsymbol{z}^{\ell_M}, \alpha) d\theta_i \times \\
&\quad \prod_{w \in s_{i,j}^{\ell_1}} \int_{\phi_k} p(w | z_{s_{i,j}}^{\ell_1} = z_k, \phi) p(\phi_k | \boldsymbol{z}_{\neg s_{i,j}}^{\ell_1}, \boldsymbol{w}_{\neg s_{i,j}}^{\ell_1}, \beta) d\phi_k,
\end{aligned}
\tag{5.1}
$$

where $\boldsymbol{z}^{\ell_1}_{\neg s_{i,j}}$ denotes the topic assignments of $d^{\ell_1}_i$ excluding those referring to $s^{\ell_1}_{i,j}$. For the integrals of Eq. (5.1) using the Multinomial-Dirichlet conjugacy for ϕ and θ, one may perform the updates using their expectations, hence :

$$\mathbb{E}_{\mathrm{Dir}(\Omega^{\ell_1}_{k,i,\neg s_{i,j}}+\Omega^{\ell_2}_{k,i}+\alpha)}[\theta_{i,k}] \times \prod_{w \in s^{\ell_1}_{i,j}} \mathbb{E}_{\mathrm{Dir}(\Psi^{\ell_1}_{k,w,\neg s_{i,j}}+\beta)}[\phi^{(w)}_k],$$

where the first term is influenced by the the topic assignments within the documents of the pair, and the second by the words of the segment. The result for the conditional probability then becomes:

$$p(z^{\ell}_{s_{i,j}} = z_k | \boldsymbol{z}^{\ell}_{\neg s_{ij}}, \boldsymbol{z}^{\ell\backslash}, \boldsymbol{w}^{\ell}, \boldsymbol{w}^{\ell\backslash}, \alpha, \beta) \propto \Big(\Omega_{i,k,\neg s_{i,j}} + \alpha\Big) \times \frac{\prod_{w \in s^{\ell}_{ij}} (\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta) \cdots (\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta + (N^{\ell}_{i,j,w} - 1))}{(\Psi^{\ell}_{k,\cdot,\neg s_{ij}} + \beta V_{\ell}) \cdots (\Psi^{\ell}_{k,\cdot,\neg s_{ij}} + \beta V_{\ell} + (N^{\ell}_{i,j} - 1))}. \tag{5.2}$$

In Eq. (5.2), the product appearing in the numerator of the second term results from the bag-of-words assumption for the words of segments. The (possibly multiple) occurrences of a word w in a segment $s^{\ell}_{i,j}$, generated by the topic k, are taken into account by the factor $(\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta)$, which is incremented by one for every other occurrence of the word after the first. For example, if word w appears twice in $s^{\ell}_{i,j}$, then $N^{\ell}_{i,j,w} = 2$, and the factor $(\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta)(\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta + 1)$ denotes the contribution of the occurrences of the word to the probability that $s^{\ell}_{i,j}$ is generated by the topic k. This way, every word of the segment contributes to the probability of sampling a particular topic. Similarly, the denominator acts as a normalization term. The progressive increase of its values can also be explained intuitively: given the bag-of-words assumption of words within a segment, the product normalizes the probability of assigning the topic k to a word of the segment, given that the previous words have also been assigned to this topic. Notice, that if the size of the segment is 1, the model as well as the sampling equations reduce to `BiLDA`.

The bag-of-words assumption in `BiLDA` results in a joint distribution of random variables (here topics) being invariant to any permutation of the variables (exchangeability). This holds for `segBiLDA` only locally, within segments. Globally, within a document, words are not exchangeable as the segment boundaries are utilized. While in `BiLDA` the topics of words are conditionally independent

given the document's topic distribution, for segBiLDA they are not, as they also depend on the rest of the segment's words.

Previous work in the monolingual case suggested to incorporate various types of text structure to topic models ranging from n-grams to parse trees. segBiLDA can be considered an extension of our model [11] in the bilingual setting. Rather than extending more complex models like Boyd's [29] that may require parsing the documents, we opt for segBiLDA due to the variety of segments it can handle. For instance, one can use linguistically motivated segmentation approaches like sentence tokenization or statistically motivated segmentation approaches like frequent n-grams with the same model. Furthermore, these segmentation approaches can be accomplished efficiently and accurately across a wide range of languages without resorting to complex linguistic analysis tools like parsers.

5.2.3 Extracting multilingual topics from comparable corpora

BiLDA and segBiLDA assume a single topic distribution for the documents of a pair, which as illustrated in Figure 4.1 is a string assumption for comparable documents. Apart from that, the motivations for adapting the bilingual topic models to comparable corpora lie on two facts: on one hand, comparable corpora are more common and easy to obtain or to construct than parallel ones, which require additional linguistic knowledge and tools. On the other hand, recent advances on cross-lingual word embeddings resulted in methods that can be directly used to estimate the semantic similarity of documents written in different languages. The latter facilitates the application of our method to various pairs of languages without expensive resources.

For comparable corpora, we first propose the λ-BiLDA model, whose graphical model is shown in Figure 5.2(iii). In this case, instead of having a single, shared topic distribution we have a topic distribution per language shown as θ^{ℓ_1} and θ^{ℓ_2} in the figure. However, these distributions are bound between them. To model naturally the possible levels of dependence between θ^{ℓ_1} and θ^{ℓ_2} we need a binding mechanism flexible enough to model the two extreme conditions: total independence between the topic distributions of the aligned documents that should result in two distinct LDA models (one per language), and a complete dependence between them (identical topic distributions) which should result in BiLDA. Similar dependence mechanisms were previously explored under the setting of streaming documents [3], where topic distributions of earlier documents affect the distributions of later documents.

The generative process for λ-BiLDA is as follows:

- for each topic $k \in [1, K]$: $\phi_k^{\ell_1} \sim Dir(\beta)$, $\phi_k^{\ell_2} \sim Dir(\beta)$
- for each document pair $d_i = (d_i^{\ell_1}, d_i^{\ell_2})$:
  - estimate λ_i with respect to the documents that form the pair $d_i = (d_i^{\ell_1}, d_i^{\ell_2})$
  - sample $\theta_i^{\ell_1} | \theta_i^{\ell_2} \sim Dir(\alpha + \lambda_i \theta_i^{\ell_2})$, $\theta_i^{\ell_2} | \theta_i^{\ell_1} \sim Dir(\alpha + \lambda_i \theta_i^{\ell_1})$
  - for language $\ell \in \{\ell_1, \ell_2\}$
    - for each of the words N^ℓ:
      - sample $z \sim Mult(1, \theta_i^\ell)$
      - sample $w \sim Mult(1, \phi_z^\ell)$

The central idea is that the topic distributions of documents in one language depend on the topic distributions of documents in the other language via a binding mechanism that generates θ^ℓ with a Dirichlet distribution depending on $\theta^{\ell\backslash}$; $\theta^\ell | \theta^{\ell\backslash} \sim Dir(\alpha + \lambda_i \theta_i^{\ell\backslash})$ and vice-versa. Note that from the Hammersley-Clifford theorem [76], fixing the two conditional distributions $\theta^{\ell_1} | \theta^{\ell_2}$ and $\theta^{\ell_2} | \theta^{\ell_1}$ defines in an unique way the distribution of $(\theta^{\ell_1}, \theta^{\ell_2})$ which implies that our generative process is well-defined.

For inferring the topics of the observed words we propose a Gibbs sampling approach, whose derivation is given in the Appendix at the end of this chapter. The update equations for the topics of the words are then $\forall \ell \in \{\ell_1, \ell_2\}$:

$$p\left(z_{i,j}^\ell = z_k | \boldsymbol{z}_{\neg i,j}^\ell, \boldsymbol{w}^\ell, \alpha, \beta, \lambda_i, \theta^{\ell\backslash}\right) \propto \frac{\Psi_{k,w,\neg i,j}^\ell + \beta}{\Psi_{k,\cdot,\neg i,j}^\ell + V_\ell \beta} \cdot \left(\Omega_{i,k,\neg i,j}^\ell + \alpha + \lambda_i \theta_{d,k}^{\ell\backslash}\right). \tag{5.3}$$

Gibbs sampling algorithms obtain posterior samples by sweeping though each block of variables and sampling from their conditional, while the remaining blocks are fixed. In practice, the algorithm initializes randomly the topics of words. Then, during the Gibbs iterations and until convergence, sampling topics for words of ℓ assumes the distribution of $\theta^{\ell\backslash}$ fixed, and hence can be accessed as assumed by the generative story.

In Eq. (5.3), λ_i captures the dependency between the topic distributions of the documents of d_i. We use cross-lingual word embeddings for its estimation. We use the average (avg) compositional function of meaning, which was shown to

be robust and effective [129, 21]. Having the vectors of the document pair $d_i = (d_i^{\ell_1}, d_i^{\ell_2})$ in the embedded space, we then estimate λ_i using the cosine similarity. Calculated in this way, $\lambda_i \in [-1, 1]$ and since $\theta_{d,k} \in [0, 1]$ it follows for the second term of Eq. 5.3 that $\Omega^{\ell}_{i,k,\neg i,j} >> \lambda_i \theta_{d,k}$, which results in negligible impact for $\lambda_i \theta_{d,k}$. To circumvent that we use:

$$\lambda_i' = \lambda_i \times |N_i^{\ell\backslash}| = \Omega^{\ell\backslash}_{d,k}. \tag{5.4}$$

Notice that incorporating Eq. (5.4) to Eq. (5.3) has as an additional advantage as the model generalizes previous models. In particular, it follows that BiLDA becomes a special case of λ-MuLDA with $\lambda = 1$ (complete dependency where θ^{ℓ} and $\theta^{\ell\backslash}$ are the same topic distributions). Also, when $\lambda = 0$ (case of independence) we have two distinct LDAs, one per language.[3]

5.2.4 Combining the two models

To this point, we proposed segBiLDA that incorporates the boundaries of coherent segments like sentences, and λ-BiLDA, that assumes bound topic distributions for the paired documents in the two languages. The two models can be combined: λ-segBiLDA assigns consistent topics in the words of the segments of the documents and also assumes different topic distributions for each language.

We illustrate λ-segBiLDA in Figure 5.2(iv). We omit the generative story, since it is a direct combination of the generative stories of segBiLDA and λ-BiLDA. The inference process is given by the following equation, whose derivation is given in the Appendix at the end of the chapter. To sample the topics of segments from the conditional distribution for $\forall \ell \in \{\ell_1, \ell_2\}$:

$$\begin{aligned} p\Big(z^{\ell}_{i,j} = z_k | \boldsymbol{z}^{\ell}_{\neg i,j}, \boldsymbol{w}^{\ell}, \alpha, \beta, \lambda_i, \theta^{\ell\backslash}\Big) \propto [\Omega^{\ell}_{i,k,\neg s_{i,j}} + \alpha + \lambda_i \Omega^{\ell\backslash}_{d,k}] \times \\ \frac{\prod\limits_{w \in s^{\ell}_{ij}} (\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta) \cdots (\Psi^{\ell}_{k,w,\neg s_{ij}} + \beta + (N^{\ell}_{i,j,w} - 1))}{(\Psi^{\ell}_{k,\cdot,\neg s_{ij}} + \beta V_{\ell}) \cdots (\Psi^{\ell}_{k,\cdot,\neg s_{ij}} + \beta V_{\ell} + (N^{\ell}_{i,j} - 1))}. \end{aligned} \tag{5.5}$$

Notice how both assumptions are relaxed is this model: the first term of the result (discussed in the Appendix of the chapter) shown in Eq. (5.5) accounts for the topic dependence between the paired documents, while the second incorporates the segment boundaries. The Gibbs sampling updates for λ-segBiLDA are

[3] Although by definition $\lambda_i \in [-1, 1]$ in all our experiments we found $\lambda_i > 0$.

Algorithm 3: A Gibbs Sampling iteration for λ-segBiLDA

Input: the words of the document pairs, cross-lingual embeddings, α, β, K
for document pair $d_i = (d_i^{\ell_1}, d_i^{\ell_2}), i \in [1, D]$ **do**
  Calculate λ_i using the cross-lingual embeddings
end
Segment the documents of each language
//Initialize counters Ψ^{ℓ},Ω^{ℓ}
for language $\ell \in [\ell_1, \ell_2]$ **do**
  for document $d_i^{\ell}, i \in [1, D]$ **do**
    for segment $s_{i,j}^{\ell} : j \in \{1, \ldots, S_i\}$ **do**
      Decrease counter variables Ψ^{ℓ},Ω^{ℓ} according to the previous topic assignments of the words of $s_{i,j}^{\ell}$
      Calculate the probabilities of the new topic of the words of $s_{i,j}^{\ell}$ (Eq. 5.5)
      Sample the topics of the words of $s_{i,j}^{\ell}$ using the calculated probabilities
      Increase counters Ψ^{ℓ},Ω^{ℓ}
    end
  end
end

also presented in Algorithm 3. One may obtain similar algorithms for each of the bilingual topic models presented in this section or adapt Algorithm 3 by selecting the appropriate equation while calculating the probabilities of the topics.

5.3 Experimental Framework

The comparable corpora In order to evaluate the proposed models, we perform a series of evaluation tasks using Wikipedia documents in four language pairs as our comparable corpora. The language pairs are English-French (En-Fr), English-German (En-Ge), English-Italian (En-It) and English-Portuguese (En-Pt). Table 5.2 shows the basic statistics of these datasets. For topic modeling purposes we have sampled subsets from the full datasets (right part of the table) consisting of 10,000 documents for each pair. Since the sampling was random, it is not the same 10,000 English documents used for every language pair. Notice in the table that for each pair English is the language with the most words (N), which was expected since often Wikipedia lemmas are first written in English and then translated to other languages. This is also why Wikipedia is a suitable comparable corpus; the English version usually includes more information on a topic compared to other languages. To extract comparable Wikipedia documents one can use the inter-

	Full Dataset			Topic Modeling Subsets		
Dataset	D	N	V	D	N	V
$\text{Wiki}^{\text{En}}_{\text{En-Fr}}$	937,991	259M	619,056	10,000	2.55M	33,925
$\text{Wiki}^{\text{Fr}}_{\text{En-Fr}}$	937,991	159M	466,423	10,000	1.64M	26,604
$\text{Wiki}^{\text{En}}_{\text{En-Ge}}$	849,955	391M	599,233	10,000	2.54M	33,198
$\text{Wiki}^{\text{Ge}}_{\text{En-Ge}}$	849,955	391M	894,798	10,000	1.81M	44,898
$\text{Wiki}^{\text{En}}_{\text{En-It}}$	732,416	200M	519,897	10,000	2.55M	33,934
$\text{Wiki}^{\text{It}}_{\text{En-It}}$	732,416	125M	360,760	10,000	1.56M	25,436
$\text{Wiki}^{\text{En}}_{\text{En-Pt}}$	540,467	160M	428,293	10,000	2.86M	34,687
$\text{Wiki}^{\text{Pt}}_{\text{En-Pt}}$	540,467	61M	222,547	10,000	1.9M	19,347

Table 5.2: Data used for evaluating topic coherence (left) and topic modeling (right) purposes. The names signify the language pair and the language that the statistics correspond to. For instance, $\text{Wiki}^{\text{En}}_{\text{En-Fr}}$ are the English documents of the En-Fr corpus.

language links. For the sake of reproducibility, we have used the bilingual corpora as made available by *linguatools*.[4] We have cleaned the documents to remove *html* tags and tables using Python v2.7 and Beautiful Soup v4.5.1.[5] The statistics of Table 5.2 are after the pre-processing steps, that include lower-casing, filtering the numerical terms out, stemming using the WordNet stemmer as implemented in [20], stop-word removal using the stopword lists of [20] and finally filtering vocabulary terms with less than 4 occurrences in the corpus.

The models We evaluate the following six models for each language pair: (i) `BiLDA` that has been proposed in [127] (ii) $\texttt{segBiLDA}_s$ that was presented above and uses sentences as coherent segments, (iii) $\texttt{segBiLDA}_b$ that is `segBiLDA` with the 1,000[6] most frequent bigrams considered as coherent segments, as well as λ-`BiLDA`, λ-$\texttt{segBiLDA}_s$ and λ-$\texttt{segBiLDA}_b$ that extend the first three models for comparable corpora. We have implemented each of these models using Python, Numpy and Scipy. As commonly done, we follow previous work e.g., [23], and we set for each model the Dirichlet hyper-parameters $\alpha = 1/K$ and $\beta = 0.01$, where K is the

[4] urlhttp://linguatools.org/tools/corpora/wikipedia-comparable-corpora/

[5] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[6] We use 1K bigrams following the work of [103] who found this number to be the optimal choice for similar corpora.

BiLDA						segBiLDA$_s$						segBiLDA$_b$					
Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]		Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]		Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]	
En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr
city	commun	team	championnat	music	group	citi	vill	team	championnat	film	the	citi	commun	team	championnat	film	film
popul	situ	play	club	album	album	popul	situ	play	club	releas	of	also	vill	play	club	releas	dan
town	vill	first	premi	releas	titre	area	commun	first	premi	also	film	popul	situ	first	premi	album	and
area	référent	game	coup	song	the	town	part	world	match	album	and	town	région	world	coup	song	sort
locat	région	player	match	record	and	locat	grand	leagu	coup	first	sort	area	villag	player	saison	music	group

λ-BiLDA						λ-segBiLDA$_s$						λ-segBiLDA$_b$					
Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]		Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]		Topic 3 [City]		Topic 5 [Sports]		Topic 8 [Art]	
En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr	En	Fr
citi	commun	team	championnat	music	group	popul	situ	team	championnat	film	the	citi	commun	team	championnat	film	film
popul	vill	play	club	album	premi	city	espec	play	premi	releas	of	also	vill	play	club	releas	dan
town	situ	first	premi	release	sort	also	vill	first	club	album	film	town	situ	first	premi	music	the
refer	villag	player	coup	song	the	area	part	world	coup	also	and	area	villag	player	coup	album	group
area	référent	game	saison	record	album	town	grand	leagu	saison	song	sort	locat	région	world	saison	song	sort

Table 5.3: For each of the topic models we present three topics: City, Sports, Art for the "En-Fr" Wikipedia corpus. Notice the strong intra-semantic (words within a topic) and inter-semantic (topics across languages) coherence.

$\ell_1 - \ell_2$	D	N^{ℓ_1}	V^{ℓ_1}	N^{ℓ_2}	V^{ℓ_2}
En-Fr	1.92M	56M	88,774	64M	130,146
En-Ge	1,92M	53M	86,691	50M	332,285
En-It	1,90M	55M	88,172	55M	155,715
En-Pt	1.96M	54M	88,241	56M	145,112

Table 5.4: Statistics for the Europarl corpus. We used the Europarl data to train the BilBowa word representations.

number of topics.

Training Bilbowa. To estimate the word representations of the En-Fr, En-Ge, En-It and En-Pt pairs we used Bilbowa [71] to generate a dictionary of word embeddings, where words from two different languages are projected to the same space. We have used the open implementation of Bilbowa[7] with its default parameters and the training epochs set to 10.[8] The model requires parallel text, and for this purpose we used the Europarl corpus [96]. The statistics of the pairs of languages for the Europarl data are shown in Table 5.4.

Visualizing the topics. As an initial qualitative evaluation of the learned topics, Figure 5.3 presents for 3 topics (`City`, `Sports`, `Art`) the five words with the highest probability for each of the six topic models. The topics were identified after training each model for 200 Gibbs sampling iterations on the Wiki$_{\text{En-Fr}}$ corpus

[7] `https://github.com/gouwsmeister/bilbowa`

[8] -size 100 -window 5 -sample 1e-4 -negative 5 -binary 0 -adagrad 1 -xling-lambda 1 -threads 12 -epochs 10

with $K = 10$ topics. Visual inspection of the topics reveals that the models produce topics that are intra-semantically coherent, that is the words that constitute the topics are semantically relevant. Further, the topics are inter-semantically coherent, that is the topics and aligned across languages and closely related words represent them. For instance, the "Sports" topic in English contains mostly word like "team", "play", "season" while in French one can find their (stemmed) translations: "équipe", "jouer", "saison". Although reassuring the visual inspection of the topics in not sufficient to compare the models. In the rest, we evaluate the models intrinsically, that is independently of an application as well as extrinsically in the framework of a cross-lingual document retrieval application.

5.3.1 Intrinsic Evaluation

Normalized PMI As we discussed in the previous chapter (Chapter 4), recent research has showed that calculating the nPMI scores for the topics learned by the models correlates well with human judgments of their quality. Following these results, we present in Table 5.5 the topic coherence scores as measured by the nPMI. Recall, the scores of nPMI range in [-1,1], where in the limit of -1 two words w_1 and w_2 never occur together, while in the limit of +1 they always occur together (complete co-occurrence). As in [102], for each topic, we aggregate the topic coherence scores over three different topic cardinalities: $N \in \{5, 10, 15\}$. The reference corpora for calculating the topic coherence for each language are the "Full Datasets" of Table 5.2 excluding the "Topic Modeling Subsets". For English we opt for $\text{Wiki}^{\text{En}}_{\text{En-Fr}}$, which is the biggest, whereas for the rest of the languages we use their respective Wikipedia datasets.

In Table 5.5, note that in most cases λ-segBiLDA$_b$ outperforms the rest of the models, while segBiLDA$_b$ and segBiLDA$_s$ follow. Notice, how BiLDA although competitive for low values of K does not perform as well. This is probably due to the fact that the concept of context is encapsulated in the calculation of the nPMI scores, and the segBiLDA topic models explicitly account for this. In general, increasing the number of topics from 10 to 25 or 50 seems to improve the performance measured by nPMI. For instance, in the lower part of the Table with the averages across languages, increasing the topics increases the best performance from .135 to .151. From the table, it is evident that adapting the topic models for comparable corpora improves the scores, apart from the case of segBiLDA$_s$. For the rest of the models (BiLDA and segBiLDA$_b$) the λ- counterparts perform better

ℓ_2	K	BiLDA	λ-BiLDA	segBiLDA$_s$	λ-segBiLDA$_s$	segBiLDA$_b$	λ-segBiLDA$_b$
En	10	$.105_{\pm.07}$	$.102_{\pm.07}$	$.090_{\pm.06}$	$.080_{\pm.07}$	$.113_{\pm.05}$	$.124_{\pm.06}$
En	25	$.124_{\pm.10}$	$.125_{\pm.04}$	$.129_{\pm.11}$	$.111_{\pm.07}$	$.140_{\pm.08}$	$.150_{\pm.07}$
En	50	$.132_{\pm.05}$	$.129_{\pm.10}$	$.125_{\pm.06}$	$.125_{\pm.10}$	$.157_{\pm.08}$	$.155_{\pm.05}$
Fr	10	$.114_{\pm.05}$	$.114_{\pm.06}$	$.105_{\pm.06}$	$.053_{\pm.05}$	$.088_{\pm.06}$	$.125_{\pm.07}$
Fr	25	$.122_{\pm.06}$	$.121_{\pm.06}$	$.124_{\pm.08}$	$.068_{\pm.03}$	$.120_{\pm.07}$	$.114_{\pm.05}$
Fr	50	$.124_{\pm.06}$	$.120_{\pm.07}$	$.136_{\pm.07}$	$.080_{\pm.06}$	$.123_{\pm.07}$	$.133_{\pm.06}$
Ge	10	$.198_{\pm.09}$	$.198_{\pm.11}$	$.234_{\pm.10}$	$.250_{\pm.08}$	$.203_{\pm.10}$	$.215_{\pm.10}$
Ge	25	$.174_{\pm.02}$	$.173_{\pm.11}$	$.235_{\pm.08}$	$.235_{\pm.11}$	$.187_{\pm.08}$	$.176_{\pm.04}$
Ge	50	$.183_{\pm.03}$	$.180_{\pm.02}$	$.230_{\pm.09}$	$.255_{\pm.10}$	$.181_{\pm.05}$	$.183_{\pm.04}$
It	10	$.096_{\pm.06}$	$.101_{\pm.05}$	$.102_{\pm.07}$	$.084_{\pm.06}$	$.119_{\pm.06}$	$.113_{\pm.05}$
It	25	$.109_{\pm.04}$	$.118_{\pm.05}$	$.104_{\pm.09}$	$.099_{\pm.02}$	$.143_{\pm.07}$	$.127_{\pm.06}$
It	50	$.119_{\pm.08}$	$.125_{\pm.05}$	$.122_{\pm.09}$	$.131_{\pm.07}$	$.137_{\pm.04}$	$.142_{\pm.08}$
Pt	10	$.099_{\pm.05}$	$.115_{\pm.07}$	$.108_{\pm.13}$	$.093_{\pm.09}$	$.098_{\pm.04}$	$.123_{\pm.06}$
Pt	25	$.129_{\pm.12}$	$.120_{\pm.11}$	$.164_{\pm.18}$	$.145_{\pm.11}$	$.131_{\pm.06}$	$.124_{\pm.09}$
Pt	50	$.120_{\pm.10}$	$.137_{\pm.07}$	$.143_{\pm.15}$	$.125_{\pm.06}$	$.141_{\pm.08}$	$.145_{\pm.10}$
avg	10	$.117_{\pm.07}$	$.118_{\pm.08}$	$.129_{\pm.10}$	$.110_{\pm.08}$	$.122_{\pm.06}$	$\mathbf{.135}_{\pm.07}$
avg	25	$.127_{\pm.07}$	$.129_{\pm.07}$	$.141_{\pm.11}$	$.127_{\pm.08}$	$\mathbf{.143}_{\pm.07}$	$.141_{\pm.06}$
avg	50	$.131_{\pm.06}$	$.137_{\pm.07}$	$.145_{\pm.09}$	$.137_{\pm.08}$	$.149_{\pm.06}$	$\mathbf{.151}_{\pm.06}$

Table 5.5: Topic coherence measured by the nPMI for each of the models. The averages are calculated for each model and K across languages. Overall, λ-segBiLDA$_b$ performs the best.

according to the columnwise comparison of the averaged results in the lowest rows of the table.

Although well-correlated with human judgments, for nPMI we only used a small part of the output of topic models, that is for each topic the top-N words. Furthermore, the evaluation of nPMI suffers in that is does not account for the topical overlap between the learned topics as well as recall gaps within a topic, i.e. lack of terms which should have been ideally included. Therefore, we also report the perplexity scores of held-out documents, whose calculation requires more information from the topic models.

Perplexity Another measure presented in Chapter 4 as an intrinsic measure for comparing topic models is perplexity. In line with our analysis in the previous chapter, we also compare the topic models based on the achieved perplexity scores for each language.

Here, for the perplexity calculations we assume that the held-out documents

		English						ℓ_2					
ℓ_2	K	BiLDA	λ-BiLDA	segBiLDA$_b$	λ-segBiLDA$_b$	segBiLDA$_s$	λ-segBiLDA$_s$	BiLDA	λ-BiLDA	segBiLDA$_b$	λ-segBiLDA$_b$	segBiLDA$_s$	λ-segBiLDA$_s$
Fr	25	$3423_{\pm 123}$	$\mathbf{3391}_{\pm 113}$	$3445_{\pm 115}$	$3383_{\pm 98}$	$3780_{\pm 234}$	$3727_{\pm 327}$	$2709_{\pm 70}$	$2633_{\pm 68}$	$2724_{\pm 55}$	$\mathbf{2617}_{\pm 89}$	$3111_{\pm 158}$	$2929_{\pm 135}$
Fr	50	$3009_{\pm 109}$	$2957_{\pm 114}$	$3002_{\pm 86}$	$\mathbf{2944}_{\pm 112}$	$3460_{\pm 263}$	$3420_{\pm 341}$	$2424_{\pm 56}$	$2320_{\pm 64}$	$2417_{\pm 58}$	$\mathbf{2312}_{\pm 72}$	$2891_{\pm 145}$	$2715_{\pm 100}$
Fr	100	$2725_{\pm 120}$	$\mathbf{2634}_{\pm 110}$	$2685_{\pm 98}$	$2636_{\pm 87}$	$3288_{\pm 339}$	$3236_{\pm 357}$	$2245_{\pm 43}$	$2092_{\pm 65}$	$2194_{\pm 49}$	$\mathbf{2085}_{\pm 80}$	$2720_{\pm 147}$	$2598_{\pm 113}$
Fr	150	$2642_{\pm 121}$	$\mathbf{2526}_{\pm 109}$	$2569_{\pm 103}$	$2527_{\pm 102}$	$3225_{\pm 309}$	$3176_{\pm 353}$	$2197_{\pm 44}$	$2035_{\pm 53}$	$2135_{\pm 51}$	$\mathbf{2028}_{\pm 70}$	$2696_{\pm 126}$	$2558_{\pm 112}$
Ge	25	$3338_{\pm 114}$	$3317_{\pm 83}$	$3350_{\pm 86}$	$\mathbf{3292}_{\pm 90}$	$3711_{\pm 171}$	$3639_{\pm 256}$	$4532_{\pm 434}$	$4419_{\pm 426}$	$4508_{\pm 437}$	$\mathbf{4386}_{\pm 430}$	$5248_{\pm 505}$	$4929_{\pm 620}$
Ge	50	$2920_{\pm 106}$	$2873_{\pm 102}$	$2934_{\pm 67}$	$\mathbf{2859}_{\pm 84}$	$3379_{\pm 221}$	$3303_{\pm 277}$	$3952_{\pm 367}$	$3791_{\pm 376}$	$3951_{\pm 374}$	$\mathbf{3772}_{\pm 361}$	$4727_{\pm 505}$	$4463_{\pm 542}$
Ge	100	$2666_{\pm 115}$	$2589_{\pm 91}$	$2625_{\pm 112}$	$\mathbf{2570}_{\pm 116}$	$3200_{\pm 308}$	$3152_{\pm 286}$	$3617_{\pm 341}$	$\mathbf{3408}_{\pm 312}$	$3577_{\pm 339}$	$3412_{\pm 310}$	$4424_{\pm 561}$	$4227_{\pm 555}$
Ge	150	$2581_{\pm 108}$	$2481_{\pm 105}$	$2528_{\pm 117}$	$\mathbf{2471}_{\pm 107}$	$3126_{\pm 290}$	$3109_{\pm 287}$	$3554_{\pm 330}$	$\mathbf{3284}_{\pm 303}$	$3484_{\pm 308}$	$3314_{\pm 298}$	$4322_{\pm 558}$	$4163_{\pm 534}$
It	25	$3393_{\pm 136}$	$3364_{\pm 117}$	$3411_{\pm 139}$	$\mathbf{3360}_{\pm 108}$	$3780_{\pm 180}$	$3659_{\pm 195}$	$2688_{\pm 137}$	$2606_{\pm 110}$	$2696_{\pm 152}$	$\mathbf{2589}_{\pm 116}$	$3140_{\pm 258}$	$2886_{\pm 233}$
It	50	$2994_{\pm 101}$	$2938_{\pm 100}$	$2983_{\pm 98}$	$\mathbf{2933}_{\pm 101}$	$3463_{\pm 167}$	$3346_{\pm 190}$	$2405_{\pm 112}$	$2304_{\pm 85}$	$2404_{\pm 103}$	$\mathbf{2292}_{\pm 88}$	$2912_{\pm 215}$	$2678_{\pm 247}$
It	100	$2714_{\pm 94}$	$2639_{\pm 88}$	$2691_{\pm 81}$	$\mathbf{2637}_{\pm 87}$	$3261_{\pm 207}$	$3147_{\pm 203}$	$2225_{\pm 108}$	$2099_{\pm 78}$	$2210_{\pm 96}$	$\mathbf{2092}_{\pm 72}$	$2787_{\pm 277}$	$2561_{\pm 246}$
It	150	$2628_{\pm 86}$	$2535_{\pm 84}$	$2579_{\pm 88}$	$\mathbf{2531}_{\pm 77}$	$3208_{\pm 225}$	$3090_{\pm 203}$	$2188_{\pm 113}$	$2036_{\pm 80}$	$2143_{\pm 102}$	$\mathbf{2030}_{\pm 71}$	$2730_{\pm 273}$	$2527_{\pm 259}$
Pt	25	$3219_{\pm 173}$	$3187_{\pm 177}$	$3218_{\pm 161}$	$\mathbf{3185}_{\pm 152}$	$3472_{\pm 332}$	$3459_{\pm 419}$	$2139_{\pm 120}$	$2042_{\pm 101}$	$2139_{\pm 108}$	$\mathbf{2040}_{\pm 81}$	$2497_{\pm 180}$	$2241_{\pm 129}$
Pt	50	$2837_{\pm 175}$	$2812_{\pm 170}$	$2832_{\pm 157}$	$\mathbf{2811}_{\pm 165}$	$3201_{\pm 364}$	$3152_{\pm 419}$	$1917_{\pm 110}$	$1809_{\pm 87}$	$1914_{\pm 99}$	$\mathbf{1797}_{\pm 87}$	$2337_{\pm 174}$	$2045_{\pm 121}$
Pt	100	$2591_{\pm 180}$	$2529_{\pm 167}$	$2555_{\pm 161}$	$\mathbf{2524}_{\pm 166}$	$2998_{\pm 403}$	$2980_{\pm 416}$	$1775_{\pm 104}$	$1638_{\pm 81}$	$1752_{\pm 98}$	$\mathbf{1636}_{\pm 78}$	$2200_{\pm 150}$	$1945_{\pm 136}$
Pt	150	$2506_{\pm 183}$	$2424_{\pm 170}$	$2448_{\pm 165}$	$\mathbf{2422}_{\pm 166}$	$2948_{\pm 383}$	$2921_{\pm 419}$	$1739_{\pm 101}$	$\mathbf{1587}_{\pm 75}$	$1699_{\pm 93}$	$1593_{\pm 72}$	$2132_{\pm 149}$	$1918_{\pm 120}$

Table 5.6: The perplexity scores achieved by the proposed topic models for four bilingual datasets when $K \in \{25, 50, 100, 150\}$. The best (lowest) score achieved per language and k is shown in bold. λ-segBiLDA$_b$ achieves the best perplexity scores in most of the experiments.

form thematically-aligned pairs (as during training) and, depending on the topic model, shared or language-dependent per-document distributions are inferred that are used at Eq. (4.9). In the next section, where we will compare the performance of the models in an extrinsic task, we will ignore the links within the pairs to demonstrate than our models perform well under both settings.

Table 5.6 presents the perplexity scores achieved by the topic models. The reported scores are the averages of 10-fold cross-validation as follows: (i) we split the datasets in 10 disjoint sets, (ii) we repeat the training and perplexity calculation steps 10 times, each time considering the i-th set to be the held-out documents and the remaining 9 sets for training. The goal is to exclude any bias due to the split. We present the scores for $K \in \{25, 50, 100, 150\}$. In terms of perplexity, λ-segBiLDA$_b$ and λ-BiLDA clearly outperform the rest of the systems consistently for each language and language pair. The third best performing system is segBiLDA$_b$ and BiLDA follows. λ-segBiLDA$_s$ and segBiLDA$_s$ achieve the worst perplexity scores for every experiment. segBiLDA$_s$, while competitive when evaluated using the nPMI scores, performs poorly in this task.

Shown from a different angle, it seems that the systems who build on the bag-of-words assumption (BiLDA and λ-BiLDA) consistently outperform those that incorporate the boundaries of large spans like sentences (segBiLDA$_s$ and λ-segBiLDA$_s$). That was expected as it is in line with previous work [11], where incorporating text structure in the form of sentence boundaries was found to lead to higher (worse) perplexity. One the other hand, incorporating the boundaries of smaller spans

like bigrams, helps perplexity performance as λ-segBiLDA$_b$ seems to be the best performing model overall, especially when the number of topics increases. This is also in line with previous work: [103] showed how bigram boundaries improve the topic model results. In fact, λ-segBiLDA$_b$ further improves segBiLDA$_b$ which is inspired by [103] since it consistently achieves better perplexity scores.

Another interesting remark concerns the effect of λ, whose goal is to adapt the topic models for comparable corpora. Notice that λ-BiLDA, λ-segBiLDA$_s$ and λ-segBiLDA$_b$ outperform BiLDA, segBiLDA$_s$ and segBiLDA$_b$ respectively for each of the experiments and topic values. This highlights the positive effect of the proposed binding mechanism on the achieved perplexity scores. What is more, that was achieved by using a simple yet powerful mechanism (aggregation of word embeddings) for calculating the value of λ for each document pair and these results can be potentially refined when applying more complex strategies. Effectively, this is the answer to the question ($Q2$) that the chapter investigates. Adapting topic models for comparable corpora improves their generalization performance and, importantly, these improvements are consistent across different topic models (here BiLDA, segBiLDA$_s$ and segBiLDA$_b$) and different pairs of languages.

Lastly, Figure 5.3 shows the perplexity curves for each language for 200 Gibbs iterations for every language pair and system we evaluated. There are two main observations from the Figure. First, as in Table 5.6, for each experiment λ-BiLDA achieves the lowest perplexity values among the systems that are shown. Second, segBiLDA and λ-segBiLDA are the fastest to converge. They need ~ 50 iterations to converge, while BiLDA and λ-BiLDA need ~ 200, that is 4x times more. In terms of computation time, the benefit is similar as the cost of Gibbs iterations are roughly the same.[9]

5.3.2 Extrinsic Evaluation

Cross-lingual Document Retrieval We conclude the evaluation of the presented topic models by reporting their performance in the framework of a cross-lingual document retrieval application. As discussed during perplexity evaluation, the model can infer the per-document topic distribution for previously unseen data. Recall, that as Figure 5.3 depicts, the learned topics are aligned. Therefore, one

[9]Measured on an Intel Xeon CPU E5-2643 v3 @ 3.40GHz segBiLDA iterations are ~20% faster. Since, this may vary across datasets depending on the number of sentences/document and their length (Eq. (5.2)), we opt for only reporting the 4x speedup due to less sampling iterations without taking the faster iterations into account.

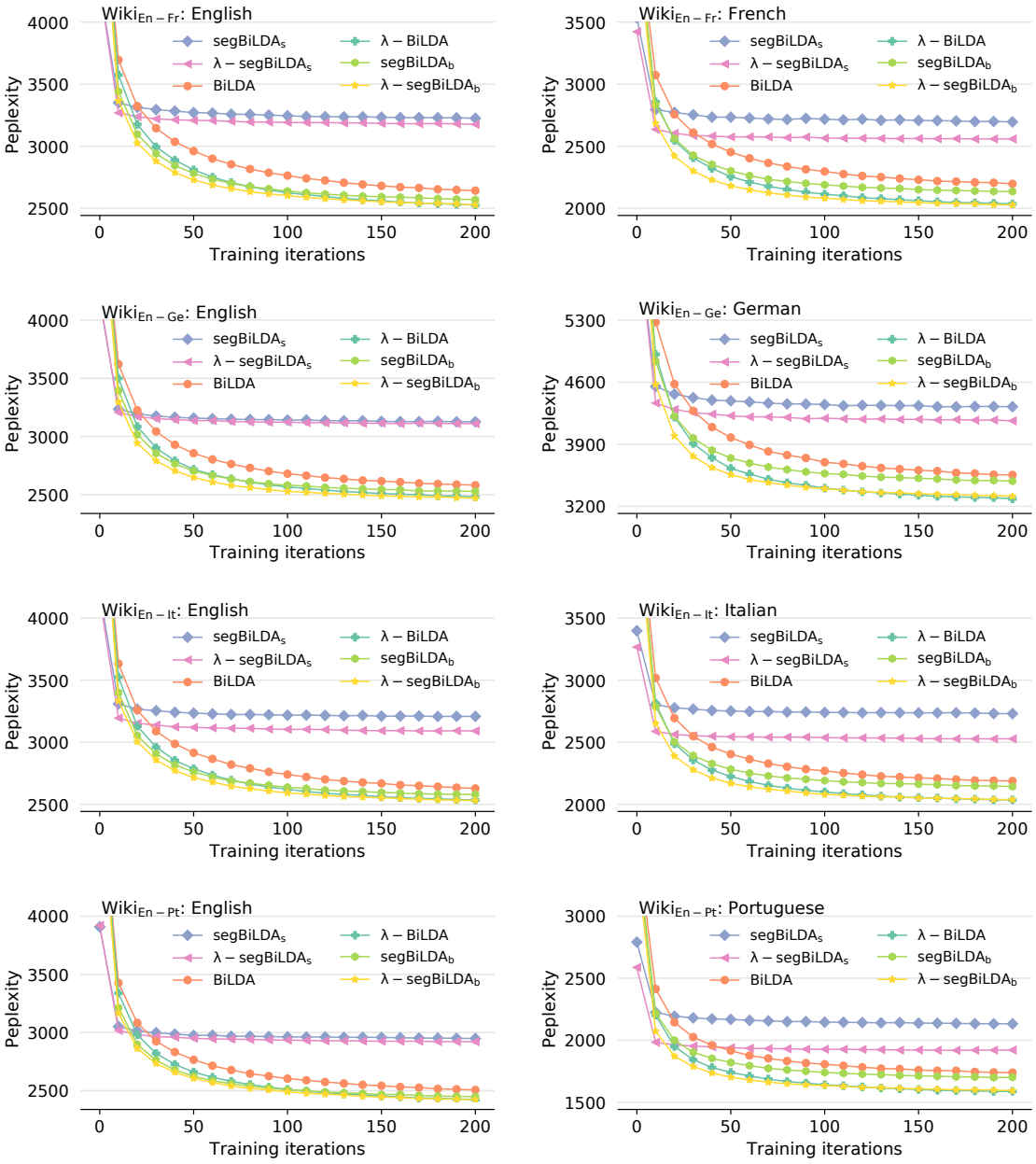


Figure 5.3: The perplexity curves of the four models for the designated datasets for 200 Gibbs sampling iterations when $K = 150$ topics. In the left column of the figures we visualize the perplexity calculated for the English documents of the comparable pairs while in the right column the documents of the second language of the pair. The proposed model λ-BiLDA consistently outperforms the rest of the systems.

may perform inference with a trained model in each language separately, without requiring the explicit links between the documents of a pair. To achieve that, the per-words topic distributions of each language are used. Then, documents with similar topic distributions written in different languages are actually similar due to the inter-semantic coherence of the topics alignments between the learned topics. This is a central observation that enables various cross-lingual applications [178] as well as cross-lingual document retrieval.

The task we propose is a cross-lingual document discovery task (CLDD), The goal is to identify counterpart Wikipedia documents due to cross-language links. In particular, given a document $d_i^{\ell_1}$ as a query, one needs to identify the corresponding document $d_i^{\ell_2}$. For instance, given an English document for "Dog" the task is to retrieve the German article for "Hund" and, vice versa given the article for "Hund" one must retrieve the article for "Dog".

Following [61, 177] who found bilingual topic models efficient for the task we address it using the following pipeline. For each of the four language pairs, we train topic models on 9,000 document pairs (18,000 documents). For the remaining 2,000 documents (that is 1,000 pairs of documents) we infer their topic distributions using one language at a time. We consider the cross-language links to be our golden standard. Then, given a document $d_i^{\ell_1}$ whose inferred topic distribution is $\theta_i^{\ell_1}$, we rank every document written at ℓ_2 according to the KL-divergence (Kullback-Leibler divergence: [99]) between $\theta_i^{\ell_1}$ and $\theta_j^{\ell_2}$ and using the golden links evaluate the performance. The KL-divergence measures the distance of probability distributions and is a suitable distance measure for our case as the topic distributions are probability distributions. We repeat the retrieval experiment 10 times by randomly selecting 500 documents (and their counterparts) out of the 1,000 held-out document pairs. As evaluation measure, we report the scores of Mean Reciprocal Rank (MRR) [176] that accounts for the rank of the true positive documents in the returned ranked list.[10] The scores of the MRR evaluation measure are given by:

$$\text{MRR} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{\text{rank}_i}$$

where $|D|$ is the number of documents (queries) at each experiment and rank_i denotes the rank of the true document to be retrieved. Further, one has $\text{MRR} \in [0, 1]$

[10] For cases where there is a single golden documents for each query, MRR is equivalent to Mean Average Precision.

and the higher the score, the higher the rank of the true positive document in the returned list is.

Table 5.7 reports the achieved scores for the document representations inferred for each topic model. The scores are the average performance of the 10 experiments accompanied by the standard deviations. In terms of notation, $\ell_1 \rightarrow \ell_2$ (e.g., En→Fr) stands for the experiment where the documents of ℓ_1 (e.g., English) are used as queries and the documents of ℓ_2 (e.g., French) are to be retrieved. The results of the table clearly establish the improvements on the task due the adaptation of the bilingual topic models for comparable corpora. Notice how λ-BiLDA, λ-segBiLDA$_b$ outperform the rest of the models and especially their counterparts BiLDA and segBiLDA$_b$ in most of the experiments. The observed improvements are consistent across the language pairs and number of topics $K \in \{25, 50, 100, 150\}$. This suggests that quantifying the semantic similarity between the documents of the pairs during training led to discovering better topics, whose performance we evaluated in the CLDD task by trying to identify the links of held-out document pairs.

It is to be noted that λ-segBiLDA$_s$ and segBiLDA$_s$ both perform poorly on the task. We believe that this is due to the fact that assuming large spans like sentences in Wikipedia documents to be thematically coherent results in per-document topic distributions unable to capture fine-grained differences between documents. In turn, such fine-grained differences are necessary for achieving high performance on the CLDD task.

Overall, our results suggest that incorporating text structure in the form of short text spans (bigrams) and adapting the bilingual topic models for comparable corpora benefits the performance on CLDD.

5.4 Summary

In this chapter we presented two extensions of bilingual topic models. First, motivated by the findings of Chapter 4 concerning prior knowledge of text structure, we proposed to incorporate such knowledge in bilingual topic models. Then, we also identified that a popular version of bilingual topic models can be better adapted to comparable corpora if a robust mechanism for calculating document similarities written in different languages is available. To achieve that we proposed to use cross-lingual embeddings that are known to capture the semantics of words. We evaluated different versions of the novel topic models with regards to

		MRR					
K	$\ell_1 \rightarrow \ell_2$	BiLDA	λ-BiLDA	segBiLDA$_b$	λ-segBiLDA$_b$	segBiLDA$_s$	λ-segBiLDA$_s$
25	En→Fr	$37.0_{\pm 1.1}$	$\mathbf{39.7}_{\pm 1.2}$	$36.0_{\pm 1.0}$	$37.1_{\pm 0.9}$	$14.6_{\pm 0.6}$	$7.6_{\pm 0.4}$
50	En→Fr	$43.8_{\pm 1.3}$	$\mathbf{44.6}_{\pm 1.1}$	$41.9_{\pm 1.3}$	$41.8_{\pm 1.5}$	$13.7_{\pm 0.6}$	$14.1_{\pm 0.7}$
100	En→Fr	$43.6_{\pm 1.4}$	$45.3_{\pm 1.9}$	$42.6_{\pm 1.9}$	$\mathbf{47.2}_{\pm 1.1}$	$13.4_{\pm 0.4}$	$11.4_{\pm 0.5}$
150	En→Fr	$38.5_{\pm 2.1}$	$39.2_{\pm 1.5}$	$39.3_{\pm 1.0}$	$\mathbf{42.7}_{\pm 1.5}$	$18.0_{\pm 1.1}$	$13.7_{\pm 0.9}$
25	Fr→En	$37.8_{\pm 0.9}$	$\mathbf{39.3}_{\pm 0.9}$	$36.7_{\pm 0.6}$	$37.6_{\pm 1.0}$	$14.3_{\pm 0.6}$	$7.5_{\pm 0.4}$
50	Fr→En	$44.0_{\pm 1.1}$	$\mathbf{47.1}_{\pm 1.3}$	$43.0_{\pm 1.2}$	$44.2_{\pm 1.1}$	$13.6_{\pm 0.8}$	$14.3_{\pm 0.7}$
100	Fr→En	$45.7_{\pm 1.2}$	$45.7_{\pm 0.9}$	$44.0_{\pm 1.2}$	$\mathbf{47.7}_{\pm 1.1}$	$13.6_{\pm 0.6}$	$10.4_{\pm 0.7}$
150	Fr→En	$39.5_{\pm 1.6}$	$42.7_{\pm 1.3}$	$41.4_{\pm 1.3}$	$\mathbf{45.2}_{\pm 1.2}$	$19.3_{\pm 0.9}$	$13.6_{\pm 0.6}$
25	En→Ge	$44.1_{\pm 1.4}$	$42.4_{\pm 1.1}$	$43.2_{\pm 1.2}$	$\mathbf{45.0}_{\pm 0.8}$	$12.6_{\pm 0.7}$	$18.5_{\pm 0.6}$
50	En→Ge	$51.7_{\pm 1.4}$	$\mathbf{55.7}_{\pm 1.0}$	$49.4_{\pm 1.0}$	$52.0_{\pm 1.1}$	$19.6_{\pm 0.8}$	$15.8_{\pm 1.1}$
100	En→Ge	$51.8_{\pm 1.2}$	$\mathbf{54.2}_{\pm 0.8}$	$51.4_{\pm 0.9}$	$51.7_{\pm 1.0}$	$21.1_{\pm 0.6}$	$16.0_{\pm 0.9}$
150	En→Ge	$48.1_{\pm 1.1}$	$49.9_{\pm 1.2}$	$47.3_{\pm 0.9}$	$\mathbf{51.5}_{\pm 1.3}$	$21.8_{\pm 0.8}$	$21.1_{\pm 0.5}$
25	Ge→En	$\mathbf{43.8}_{\pm 1.5}$	$42.9_{\pm 1.5}$	$42.6_{\pm 1.3}$	$\mathbf{43.8}_{\pm 1.3}$	$13.1_{\pm 0.6}$	$18.5_{\pm 0.8}$
50	Ge→En	$49.9_{\pm 1.2}$	$\mathbf{53.6}_{\pm 1.3}$	$48.2_{\pm 1.0}$	$50.9_{\pm 1.3}$	$17.9_{\pm 1.0}$	$16.8_{\pm 1.2}$
100	Ge→En	$51.5_{\pm 1.1}$	$\mathbf{53.7}_{\pm 1.1}$	$50.9_{\pm 0.8}$	$52.8_{\pm 1.2}$	$20.7_{\pm 0.9}$	$16.7_{\pm 0.9}$
150	Ge→En	$46.8_{\pm 1.3}$	$46.8_{\pm 1.0}$	$46.2_{\pm 1.3}$	$\mathbf{50.4}_{\pm 1.5}$	$20.6_{\pm 0.5}$	$20.3_{\pm 1.0}$
25	En→It	$\mathbf{36.4}_{\pm 1.2}$	$34.9_{\pm 0.8}$	$33.8_{\pm 0.5}$	$34.2_{\pm 1.3}$	$9.2_{\pm 0.6}$	$6.6_{\pm 0.6}$
50	En→It	$38.9_{\pm 1.7}$	$38.3_{\pm 1.2}$	$37.4_{\pm 1.4}$	$\mathbf{39.8}_{\pm 1.0}$	$13.8_{\pm 0.7}$	$10.5_{\pm 0.4}$
100	En→It	$39.0_{\pm 1.2}$	$38.9_{\pm 1.1}$	$39.0_{\pm 1.5}$	$\mathbf{41.1}_{\pm 1.4}$	$14.6_{\pm 0.5}$	$12.0_{\pm 0.6}$
150	En→It	$35.9_{\pm 1.1}$	$37.1_{\pm 0.6}$	$\mathbf{38.8}_{\pm 1.4}$	$37.8_{\pm 1.0}$	$13.0_{\pm 0.5}$	$10.8_{\pm 0.6}$
25	It→En	$\mathbf{35.5}_{\pm 0.9}$	$35.3_{\pm 1.1}$	$33.7_{\pm 0.8}$	$35.2_{\pm 1.2}$	$8.8_{\pm 0.5}$	$6.4_{\pm 0.5}$
50	It→En	$39.7_{\pm 1.5}$	$39.4_{\pm 1.0}$	$37.2_{\pm 1.5}$	$\mathbf{40.2}_{\pm 1.4}$	$13.2_{\pm 0.5}$	$10.8_{\pm 0.6}$
100	It→En	$40.6_{\pm 1.5}$	$40.0_{\pm 1.1}$	$39.1_{\pm 1.0}$	$\mathbf{40.8}_{\pm 1.2}$	$14.8_{\pm 0.7}$	$12.6_{\pm 0.4}$
150	It→En	$37.4_{\pm 1.2}$	$\mathbf{39.8}_{\pm 1.2}$	$37.4_{\pm 1.7}$	$39.3_{\pm 1.6}$	$13.4_{\pm 0.9}$	$11.2_{\pm 0.6}$
25	En→Pt	$33.8_{\pm 1.1}$	$\mathbf{34.9}_{\pm 1.1}$	$33.6_{\pm 1.3}$	$34.3_{\pm 1.3}$	$8.8_{\pm 0.4}$	$10.2_{\pm 0.8}$
50	En→Pt	$38.2_{\pm 1.2}$	$38.3_{\pm 1.5}$	$37.7_{\pm 1.1}$	$\mathbf{39.0}_{\pm 1.3}$	$14.4_{\pm 0.8}$	$11.2_{\pm 0.4}$
100	En→Pt	$38.9_{\pm 1.3}$	$38.3_{\pm 1.1}$	$38.1_{\pm 1.0}$	$\mathbf{40.2}_{\pm 1.0}$	$16.5_{\pm 0.9}$	$10.0_{\pm 0.4}$
150	En→Pt	$35.0_{\pm 1.6}$	$35.9_{\pm 1.8}$	$34.5_{\pm 1.6}$	$\mathbf{36.3}_{\pm 1.6}$	$14.8_{\pm 0.5}$	$11.2_{\pm 0.5}$
25	Pt→En	$35.8_{\pm 1.3}$	$\mathbf{36.2}_{\pm 1.4}$	$34.4_{\pm 1.2}$	$36.0_{\pm 0.8}$	$9.7_{\pm 0.6}$	$11.2_{\pm 0.8}$
50	Pt→En	$39.5_{\pm 1.6}$	$40.0_{\pm 1.6}$	$38.2_{\pm 1.1}$	$\mathbf{40.3}_{\pm 1.3}$	$15.4_{\pm 0.8}$	$12.7_{\pm 0.5}$
100	Pt→En	$41.0_{\pm 0.9}$	$40.1_{\pm 1.4}$	$40.5_{\pm 0.9}$	$\mathbf{43.4}_{\pm 0.9}$	$18.9_{\pm 0.6}$	$10.0_{\pm 0.6}$
150	Pt→En	$36.4_{\pm 1.2}$	$40.0_{\pm 1.3}$	$37.3_{\pm 1.2}$	$\mathbf{41.0}_{\pm 1.4}$	$14.6_{\pm 1.0}$	$12.6_{\pm 0.6}$

Table 5.7: The scores for the CLDD task achieved by the proposed topic models for four bilingual datasets when $K \in \{25, 50, 100, 150\}$. The best (highest) score achieved per language and k is shown in bold. The topic distributions induced by λ-segBiLDA$_b$ achieved the highest MRR scores in most of the experiments.

the topic coherence, generalization performance and cross-lingual document retrieval. Our assessment of the performance of the bilingual models suggested that both prior information of short text spans and adapting the models for comparable corpora improved the performance.

In the chapter we showed that by combining topic models with cross-lingual word embeddings one may improve the quality of the learned topics. Our future work in this setting targets to adapt the proposed models to the general setting of comparable corpora. Provided a reliable way such as high quality cross-lingual embeddings to identify links between documents, one may be able to adapt the model to the setting where no documents are not aligned in the input. On the contrary, the alignments are discovered using a document retrieval step like the one presented in the framework of CLDD.

A. Gibbs Sampling Equations for λ-BiLDA and λ-segBiLDA

We derive the Gibbs sampling equations for λ-segBiLDA:

$$
\begin{aligned}
&\text{sample } z^{\ell}_{i,j} \sim P\Big(z^{\ell}_{i,j} = k | \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \boldsymbol{z}^{\ell\backslash}, \boldsymbol{w}^{\ell}, \boldsymbol{w}^{\ell\backslash}, \alpha, \beta, \lambda_i, \theta^{\ell}, \theta^{\ell\backslash}\Big) \\
&\propto \int_{\theta^{\ell}_i} \int_{\phi} P\Big(z^{\ell}_{s_{i,j}} = k | \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \boldsymbol{w}^{\ell}, \alpha, \beta, \lambda_i, \theta^{\ell}, \theta^{\ell\backslash}\Big) d\phi d\theta^{\ell}_i \\
&\propto \int_{\theta^{\ell}_i} \int_{\phi} P(z^{\ell}_{s_{i,j}} = k | \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \theta^{\ell}, \theta^{\ell\backslash}, \lambda_i, \alpha) \times P(\boldsymbol{w}^{\ell}_{i,j} | z^{\ell}_{i,j} = k, \boldsymbol{z}^{\ell}_{\neg i,j}, \boldsymbol{w}^{\ell}_{\neg i,j}, \phi, \beta) d\phi d\theta^{\ell}_i \\
&\propto P(z^{\ell}_{s_{i,j}} = k | \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \theta^{\ell\backslash}, \lambda_i, \alpha) \times \left(\int_{\theta^{\ell}_i} \int_{\phi} P(\boldsymbol{w}^{\ell}_{s_{i,j}} | z^{\ell}_{s_{i,j}} = k, \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \boldsymbol{w}^{\ell}_{\neg s_{i,j}}, \phi, \beta) d\phi d\theta^{\ell}_i \right)
\end{aligned}
$$

For the first term, observe that sampling $P(z^{\ell}_{s_{i,j}} = k | \boldsymbol{z}^{\ell}_{\neg s_{i,j}}, \theta^{\ell\backslash}, \lambda_i, \alpha)$ is exactly the same with sampling $P(z^{\ell}_{s_{i,j}} = k | \boldsymbol{z}^{\ell}_{\neg i,j})$ in the case of standard LDA, replacing the Dirichlet parameter α with $\alpha + \lambda_i \theta^{\ell\backslash}$. The derivation of the second term where segments have several words, follows the steps shown on [11]. Hence, we deduce that:

$$P\left(z_{i,j}^{\ell}=z_k|\boldsymbol{z}_{\neg s_{i,j}}^{\ell},\boldsymbol{w}^{\ell},\alpha,\beta,\lambda_i,\theta^{\ell\backslash}\right)\propto\frac{\Omega_{d,k,\neg s_{i,j}}^{\ell}+\alpha+\lambda_i\theta_d^{\ell\backslash}}{\Omega_{d,\cdot,\neg s_{i,j}}^{\ell}+K\alpha+K\lambda_i}\times$$
$$\times\frac{\prod\limits_{w\in s_{ij}^{\ell_1}}(\Psi_{k,w,\neg s_{ij}}^{\ell}+\beta)\cdots(\Psi_{k,w,\neg s_{ij}}^{\ell}+\beta+(N_{i,j,w}^{\ell}-1))}{(\Psi_{k,\cdot,\neg s_{ij}}^{\ell}+\beta V_\ell)\cdots(\Psi_{k,\cdot,\neg s_{ij}}^{\ell}+\beta V_\ell+(N_{i,j}^{\ell}-1))}. \tag{5.6}$$

In the last result, for Gibbs sampling the fraction of the first term can be simplified by omitting the denominator as in [78, 11]:

$$\frac{\Omega_{d,k,\neg s_{i,j}}^{\ell}+\alpha+\lambda_i\theta_d^{\ell\backslash}}{\Omega_{d,\cdot,\neg s_{i,j}}^{\ell}+K\alpha+K\lambda_i}\sim\left(\Omega_{d,k,\neg s_{i,j}}^{\ell}+\alpha+\lambda_i\theta_d^{\ell\backslash}\right). \tag{5.7}$$

Integrating Eq. (5.7) to Eq. (5.6) leads to the desired result.

The equations for λ-BiLDA are simpler as ones does not have segments and the product in Eq. (5.5) and in the subsequent calculations is simplified to a simple term.

Chapter 6

Applications of word embeddings to text mining

Word embeddings, whose interesting properties of capturing the semantics of words we investigated in the previous chapter, gained a lot of attention recently. In Chapter 5 we proposed and evaluated a novel bilingual topic model that uses cross-lingual embeddings. Cross-lingual embeddings are a variant of the monolingual embeddings adapted for the case where text is written in several languages. Motivated by the performance benefits we observed, in this chapter we explore their potential in the monolingual setting in the framework of different applications.

Our main hypothesis throughout the contributions of the chapter is that incorporating expressive representations like embeddings on text mining applications should improve the performance of various text mining tasks. In other words, our goal is to develop models tailored for challenging text mining tasks that benefit from rich text representations as well as from the flexibility that neural networks offer in modeling different scenarios. While shallow architectures of neural networks can be used to learn general purpose word embeddings, such learned embeddings can be used as initializations of the first layers of deeper networks instead of using random initializations.

In the rest of the chapter we propose different models and we contribute several observations regarding three interesting tasks. In particular:

1. *Learning document representations utilizing translations of a document.* Our results in Chapter 5 demonstrated that access to multilingual versions of a document supported by cross-lingual embeddings improved the representations learned with topic models for a retrieval task. Motivated by this, we

propose to evaluate the use of embeddings for arbitrary long text spans in the document classification task. In particular, we investigate whether one can use translations of a document to obtain better-performing representations (Section 6.1).

2. *Multitask learning using neural networks.* While the question we investigate in Section 6.1 concerns the effect of multiple representations of a document for a single task, we are also interested on the effect of learning jointly correlated tasks. Having different representations such as translations of a document may result in better document representations, while having different tasks may result in a form of induced bias towards selecting hypotheses that perform well across tasks. To this end we propose a neural network architecture for jointly learning the hypotheses for two different yet correlated text classification problems (Section 6.2).

3. *Cross-lingual document retrieval as an application of the problem of optimal transport.* An important issue of text mining applications with word embeddings is to compose the representations of large text spans from the word representations provided that there exists an alternative way to calculate document distances. We argue that such a step can be omitted for several text mining applications. We propose to use a document distance metric that relies on the solution of the optimal transport problem and we demonstrate the effectiveness of this formulation on an interesting cross-lingual retrieval application (Section 6.3).

The rest of the chapter is organized as follows: Sections 6.1-6.3 contribute the models, the observations and the results for the settings described above. Then, Section 6.4 summarizes the findings of the chapter.

6.1 Polylingual text classification

Neural Networks have recently shown promising results in several machine learning and information extraction tasks [170, 195, 62]. For text classification, the use of embeddings as inputs or initializations to more complex architectures has been investigated and, for example, [90, 91] study the benefits of embeddings of sentence-length spans (sentences and/or questions). In the multilingual setting, [71] proposed an approach to learn bilingual embeddings exploiting parallel and non-parallel text in the languages, [57] proposed to use correlated components

analysis, together with small bilingual lexicons, to learn how to project embeddings in two separate languages into a common representation space and [105] proposed an approach similar to ours that uses an auto-encoder to learn bilingual representations.

In this section we propose a mechanism for combining distributed representations of documents in different languages. In this line, each document in a given language is first translated using an existing Machine Translation (MT) tool. The rationale behind is that translation offers the possibility to enrich and disambiguate the text, especially for short documents. Documents are then represented by aggregating the embeddings of their associated text spans in each language [107, 126] using a non-linear auto-encoder (AE). The AE is trained on their concatenated representations and a classifier is finally trained in the polylingual space generated by the auto-encoder. The hope is that the AE can learn language independent representations of large text spans like documents by compressing its inputs in a hidden layer and thus combining information from every language in the input. Our classification results in a subset of the publicly available Wikipedia documents show that our approach yields improved classification performance compared to the case where a classical bag-of-words space is used for document representation, especially in the case where the size of the training set is small.

The following subsections present our strategy for learning polylingual embeddings. Then, in the experimental part we empirically show that the learned representations constitute better classification features compared to several baselines. Importantly, our findings suggest that polylingual representations can strongly benefit classification settings with few labeled examples.

6.1.1 The model for learning the polylingual embeddings

Monolingual distributed representations (DRs) project text spans into a language-dependent semantic space where spans with similar semantics are closer in that space. Here, we aim to combine two distributed representations of documents corresponding to the original document and its translation using an auto-encoder. We will refer to those combined representations as *Polylingual Embeddings* (PE). We call them polylingual as two or more languages are used to derive them and they model at the same time the text written in either of them. We suppose that the auto-encoder will disentangle the language-dependent factors and will learn robust representations on its hidden layer encoding as illustrated in Figure 6.1.

Given a document d_i in English, we first translate it into French using a commercial translator,[1]. Then, we generate the distributed representations of the document and its translation using a function $\mathcal{G}$ that we will describe shortly as $\{\mathcal{G}^\ell(d_i)\}_{\ell=1}^2$ and aggregate those DRs using an auto-encoder as follows:

- $\{\mathcal{G}^\ell(d_i)\}_{\ell=1}^2$, a trained AE
- For each document d_i:
  - Concatenate $\mathcal{G}^1(d_i)$ and $\mathcal{G}^2(d_i)$
  - Get PE representation of d_i as the hidden encoding of the AE fed with the concatenation

Auto-encoders (Figure 6.1) are neural network architectures whose aim is to learn an encoding of the data by typically projecting them in a lower dimension using a single or a cascade of hidden layers **??**. To this end, we try to minimize the distance between the input and the output representations, which is commonly calculated using the Euclidean distance. Instead of learning a linear projection of the data, the activation functions of the hidden layer are non-linear (such us the sigmoid function or the hyperbolic tangent function) thus adding to the modeling capacity of the architecture.

The auto-encoder is learned over all concatenated distributed representations of documents using a stochastic back-propagation algorithm. In this work we consider two strategies to create the DR of each document. The first one is based on average pooling, where word representations are first obtained using the word2vec tool [124]. This is also the approached using in Chapter 5 for obtaining the representations of documents using the cross-lingual embeddings. DR of documents, i.e. functions $(\mathcal{G}^\ell)_{\ell\in\{1,2\}}$, are then obtained by averaging the vectors of words contained in them. In this study we consider the *continuous bag of words* (cbow) and the *skip-gram* models that generate word representations. The second strategy is based on the *distributed Memory Model of paragraph vectors* (DMMpv) and *distributed bag-of-words of paragraph vectors* (DBOWpv) models [107], that extend cbow and skip-gram respectively. DMMpv and DBOWpv, instead of learning representations for words learn representations for larger spans that is whole documents for our case here. Therefore, $(\mathcal{G}^\ell)_{\ell\in\{1,2\}}$ are defined by the output of the models without further processing.

[1] `translate.google.com`

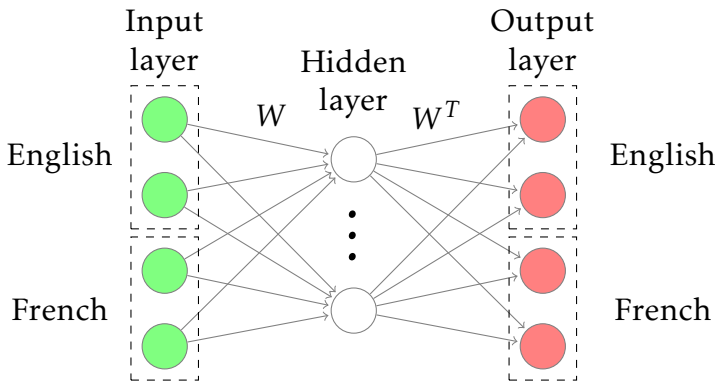


Figure 6.1: An AE that generates the PE in its hidden layer. The dashed boxes denote the document DRs in the corresponding language.

6.1.2 The Experimental Evaluation

The data Training neural network models to generate distributed representations benefits by large amounts of free text. To train the models that generate DRs we used such free texts in English and French:[2] the left part of Table 6.1 (under "Distributed Representations") presents some basic statistics for those data. We used the same number of documents for the two languages to avoid any training bias. The raw text was pre-processed by applying lower-casing and space-padding punctuation. Similarly to previous studies [124, 107], we kept the punctuation. Publicly available implementations of the models were used with their default parameters: the word2vec tool[3] for the cbow and skip-gram and the doc2vec for the DBOWpv and DMMpv from Gensim [150].

For the classification task we used the raw version of the Wikipedia dataset of the Large Scale Hierarchical Text Classification challenge [141]. The original dataset contains 60,252 categories; we restrict our study here in a fraction of the dataset with 12,670 documents belonging to the 100 most common categories. The right part of Table 6.1 presents basic statistics for this subset.

Baselines We use as a first baseline Support Vectors Machines (SVM) fed with the tf-idf representation of the documents, which is commonly used in text classification problems (denoted by SVM_{BoW}). As a second baseline, we use k-Nearest Neighbors (k-NN) and SVMs learned on the monolingual space of the DRs of English documents (denoted respectively by SVM_{DR} and $k\text{-NN}_{\text{DR}}$). These baselines

[2] http://statmt.org/
[3] https://code.google.com/p/word2vec/

	Distributed Representations			Classification			
	Docs	V	# Words	Docs	V	Avg. Doc. Len	# Labels
En	6,358,467	490,122	198M	12,670	56,886	115.32	1,17
Fr	6,358,467	713,171	177M	12,670	58,678	132.29	1,17

Table 6.1: Statistics after pre-processing the datasets. The distributed representations dataset refers to the data used to train $\mathcal{G}$. The classification data refer to the supervised dataset used for classification purposes.

aim at evaluating the value of the fusion mechanism (PE) that we propose. k-NN and SVMs were adapted to the multi-label setting (denoted respectively by SVM_{PE} and $k\text{-NN}_{\text{PE}}$). For the former, given the labels of the k nearest training instances of a test document, the algorithm returns the labels that belong to at least p% of its nearest neighbors. For each run $k \in \{13, 14, 15\}$ and $p \in \{0.1, 0.2, 0.3\}$ are decided using 5-fold cross-validation on the training data. The SVMs were used in an one-vs-rest fashion; they return every label that has a positive distance from the separating hyperplane. The value of the hyper-parameter $C \in \{10^{-1}, \ldots, 10^{4}\}$ that controls the importance of the regularization term in the optimization problem, is selected using 5-fold cross-validation over the training data.

Our approach Using the above-presented DR model, we first generate the document embeddings in English and French. These are vectors in a d-dimensional space with $d \in \{50, 100, 200, 300\}$. Then, for the AE we considered as activation functions the hyperbolic tangent and the sigmoid function. The sigmoid performed consistently better and thus we use it in the reported results. The AE was trained with tied weights using a stochastic back-propagation algorithm with mini-batches of size 10 and the euclidean distance of the input/output as loss function. The number of neurons in the hidden layer was set to be 70% of the size of the input.[4]

Experimental Results Table 6.2 presents the scores of the F_1 measure when 10% of the 12.670 documents were used for training purposes and the rest 90% for testing. We report the classification performance with the four different DR models (cbow, skip-gram, DBOWpv and DMMpv) and 2 learning algorithms (k-NN and SVMs) for different input sizes. The columns labeled $k\text{-NN}_{\text{DR}}$ and SVM_{DR} present the (baseline) performance of SVM and k-NN trained on the monolingual DRs.

[4]The code is available at `http://ama.liglab.fr/~balikas/ecir2015.zip`.

	cbow				skip-gram			
dim.	k-NN$_{DR}$	SVM$_{DR}$	k-NN$_{PE}$	SVM$_{PE}$	k-NN$_{DR}$	SVM$_{DR}$	k-NN$_{PE}$	SVM$_{PE}$
50	39.19	37.20	39.58	32.84	38.25	34.74	37.51	32.09
100	40.20	40.01	43.53	37.54	39.34	38.61	41.15	34.54
200	40.48	43.41	45.86	42.50	39.73	40.96	42.79	41.08
300	40.42	44.25	**46.33**	43.38	39.62	42.67	42.62	42.74
	DBOWpv				DMMpv			
50	24.45	25.06	30.26	24.08	24.47	25.56	29.55	24.94
100	31.20	28.53	34.63	26.88	24.74	29.31	31.21	27.22
200	27.73	29.80	36.02	30.80	18.22	30.04	29.01	32.10
300	27.79	29.92	38.71	30.82	15.98	30.49	25.20	32.01
SVM$_{BoW}$	36.03							

Table 6.2: F_1 measures of difference algorithms. The performance of 5-fold cross-validated SVM using the bag-of-words representation is 36.03

Also the last line of the table indicates the F_1 score of SVM with tf-idf representation (SVM$_{BoW}$). The best obtained result is shown in bold.

We first notice that the average pooling strategy (cbow and skip-gram) performs better compared to when the document vectors are directly learned (DBOWpv and DMMpv). In particular, cbow seems to be the best performing representation, both as a baseline model and when used as base model to generate the PE representations. On the other hand, DBOWpv and DMMpv perform significantly worse: in the baseline setting the best cbow performance achieved is 44.25 whereas the best DMMpv configuration achieves 30.49, 14 F_1 points less.

The PE representations learned on top of the four base models improve significantly over the performance of the monolingual DRs, especially for k-NN. For instance, for cbow with base-model vector dimension 200, the baseline representation achieves 40.42 F_1 and its corresponding PE representation obtains 46.33, improving almost 6 points. In general, we notice such improvements between the base DR and its respective PE, especially when the dimension of the DR representation increases. Note that the PE improvements are independent of the methods used to generate the DRs: for instance k-NN$_{PE}$ over the 200-dimensional PE DMMpv representations gains more than 11 F_1 points compared to k-NN$_{DR}$. It is also to be noted that the baseline SVM$_{BoW}$ is outperformed by SVM$_{PE}$ especially when cbow and skip-gram DRs are used.

Comparing the two learning methods (k-NN$_{PE}$ and SVM$_{PE}$), we notice that k-NN$_{PE}$ performs best. This is motivated by the fact that distributed representations

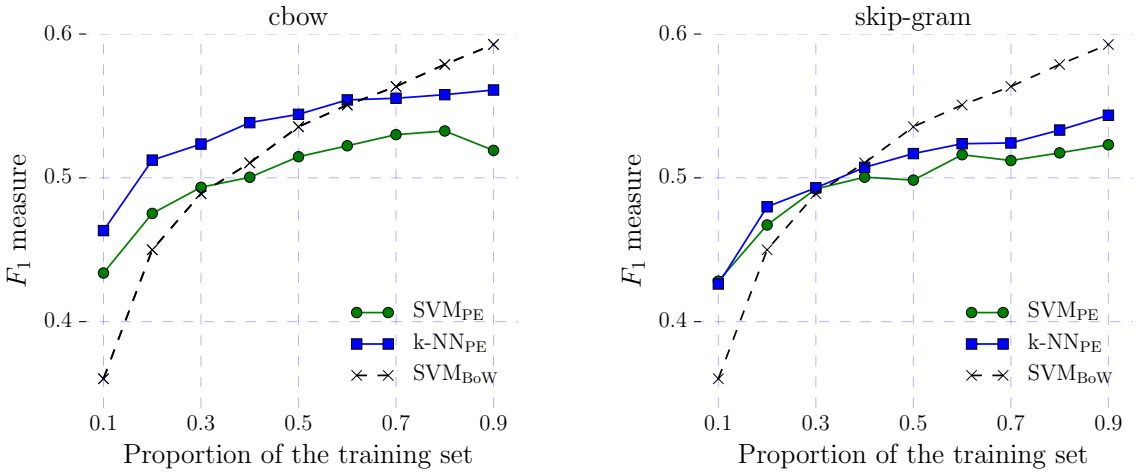


Figure 6.2: Comparison of the performance of the learning algorithms learned on different representations with respect to the available labeled data. The dimension of the PE representations is 300.

are supposed to capture the semantics in the low dimensional space. At the same time, the neighbors algorithm compares exactly this semantic distance between data instances, whereas SVMs tries to draw separating hyperplanes among them. Finally, it is known that SVMs benefit from high-dimensional vectors such as bag-of-words representations. Notably, in our experiments increasing the dimension of the representations consistently benefits SVMs.

We now examine the performance of the PE representations taking into account the amount of labeled training data. Figure 6.2 illustrates the performance of the SVM$_{\text{BoW}}$ and SVM$_{\text{PE}}$ and k-NN$_{\text{PE}}$ with PE representations when the fraction of the available training data varies from 10% of the intial training set to 90% and in the case where, cbow and skip-gram are used as distributed representations with an input size of 300. Note that if only a few training documents are available, the learning approach is strongly benefited by the rich PE representations, that outperforms the traditional SVM$_{\text{BoW}}$ setting consistently. For instance, in the experiments with 300 dimensional PE representations with cbow DRs, when only 20% of the data are labeled, the SVM$_{\text{BoW}}$ needs 20% more data to achieve similar performance, a pattern that is observed in most of the runs in the figure. When, however, more training data are available the tf-idf copes with the complexity of the problem and leverages this wealth of information more efficiently than PE does.

6.1.3 Summary

We proposed a novel approach for learning embeddings of large text spans. The novel embeddings are learned using with neural networks and in particular with a denoising autoencoder. The AE embeds translations of an input document in a language independent space the hope being that by combining information from two or more languages can benefit the performance. We empirically showed the effectiveness of the novel embeddings in the bilingual setting, in the task of document classification. Our embeddings achieved better performance compared to traditional classification approaches in the interesting case where few labeled training data are available for learning.

The main limitation of this approach is the fact it relies on a translation system for obtaining the translations of the input document which are consecutively used for obtaining the PE. An interesting extension of this work would be to quantify the effect of the quality of the translations to the performance. One could, for example, start by using dictionaries to translate between languages and then proceed with more advanced and state-of-the-art methods like the translation system we used in this work.

Another interesting extension concerns the compositional mechanism used to derive the representations of spans that are composed of several words. In our experiments we used the average pooling function that is efficient and robust but may result in loss of information when a text span has several words. To overcome this, one could use a special type of neural network architectures called recurrent neural networks that can be model sequences of elements like text. It is indeed this observation that motivates part of the contributions presented in the following section, where we will use recurrent neural networks for generating text span representations.

6.2 Multitask Learning with Neural Networks

In the previous section we showed that learning representations using translations of a text span in different languages improves the performance achieved. On the other hand, this approach required a readily available translation system, which imposes a significant development and computational overhead. Our analysis illustrated, however, that combining information in the form of translations improved the results.

Semantically, translations of a document are correlated because they discuss the same ideas in different languages. Meanwhile, one may imagine scenarios comprising different tasks, strongly or weakly correlated between them instead of correlated representations. In such a case, instead of dealing with different representations of an instance, we deal with different tasks, that may be able to benefit from each other. This can be natural language processing tasks like named entity recognition and part-of-speech tagging or multimedia classification tasks such as image classification and segmentation. For each of the previous examples, while the tasks may seem at first different, information from one task may help the performance on the other task. Motivated by our previous findings, we explore such a setting by taking advantage of the modeling flexibility that neural networks offer.

Typical scenarios of machine learning involve optimizing the performance on a task using an evaluation metric. To this end, a learning model or an ensemble of such models, are trained to perform the task while their free parameters are tuned to maximize the achieved performance. However, instead of only relying on the training signal of the given task, one may be able to do better by incorporating signals from related tasks. Multitask learning refers to the scenario where a learner is trained jointly on several interdependent tasks [30]. The hope is that the multiplicity of the tasks will result in more robust representations or a decision function that will, in turn, improve the performance on the given task. As a result, incorporating dependent tasks to the learning process of the main tasks helps selecting a better hypothesis.

We can motivate multitask learning approaches in different ways. First, as being inspired by human learning, where for learning new tasks we apply parts of the knowledge previously acquired from different yet related tasks. For instance, children or adults begin by understanding parts of language such as simple words and use this while improving and understanding larger spans like phrases and sentences. Second, from a pedagogical or a didactic point of view: we often learn a task using knowledge from previous, simpler tasks. For instance, in sports one first learns simple moves that are the basics and then elaborates on them. Third, multitask learning can be motivated from a machine learning perspective: multitask learning can be seen as a form of inductive bias or regularization. Inductive bias is anything that causes an inductive learner to prefer some hypotheses over others. An example of inductive bias is l_1, which causes a learner to select sparse solutions. In multitask learning, where a learner trains for the main task uses the signals of the other asks it is easy to see why these signals can serve as inductive

bias. It causes the learner to select solutions that perform well across tasks and therefore generalize better.

For neural networks, one may imagine two straightforward architectures for implementing multitask learning. The first, could be described as hard parameter sharing. The first layers of a network are shared across the two or more related tasks, and the last layers specialize on each of those tasks. Such an approach reduces the risk of overfitting for the parameters of the shared layers [16]. Apart from hard parameter sharing architectures, one can also imagine soft parameter sharing schemes. In the latter, given N tasks, there are N networks. Each tasks has a dedicated network and parameters. The parameters of these two networks are regularized however in order to be close. For example, [49] regularize such architectures with the l_2 norm, while [191] use the trace norm. The work presented in this section belongs in the first category. We propose a model that implements a hard parameter sharing architecture for modeling and categorizing short text spans with respect to the intensity of the sentiment the convey.

In the rest of the sections we elaborate on the task, the model and the evaluation framework used.

6.2.1 Multitask Learning for Sentiment Classification

Automatic classification of sentiment has mainly focused on categorizing tweets in either two (binary sentiment analysis) or three (ternary sentiment analysis) categories [65]. In this work we focus on the problem of fine-grained sentiment classification where tweets are classified according to a five-point scale ranging from *VeryNegative* to *VeryPositive*. To illustrate this, Table 6.3 presents examples of tweets associated with each of these categories. Five-point scales are widely adopted in review sites like Amazon and TripAdvisor, where a user's sentiment is ordered with respect to its intensity. From a sentiment analysis perspective, this defines a classification problem with five categories. In particular, Sebastiani et al. [120] defined such classification problems whose categories are explicitly ordered to be ordinal classification problems. To account for the ordering of the categories, learners are penalized according to how far from the true class their predictions are.

Although considering different scales, the various settings of sentiment classification are related. First, one may use the same feature extraction and engineering approaches to represent the text spans such as word membership in lexicons, morpho-syntactic statistics like punctuation or elongated word counts [10, 93].

Second, one would expect that knowledge from one task can be transfered to the others and this would benefit the performance. Knowing that a tweet is "Positive" in the ternary setting narrows the classification decision between the *VeryPositive* and *Positive* categories in the fine-grained setting. From a research perspective this raises the question of whether and how one may benefit when tackling such related tasks and how one can transfer knowledge from one task to another during the training phase.

Our focus in this work is to exploit the relation between the sentiment classification settings and demonstrate the benefits stemming from combining them. To this end, we propose to formulate the different classification problems as a multitask learning problem and jointly learn them. Multitask learning [30] has shown great potential in various domains and its benefits have been empirically validated [38, 148, 115, 114] using different types of data and learning approaches. An important benefit of multitask learning is that it provides an elegant way to access resources developed for similar tasks. By jointly learning correlated tasks, the amount of usable data increases. For instance, while for ternary classification one can label data using distant supervision with emoticons [68], there is no straightforward way to do so for the fine-grained problem. However, the latter can benefit indirectly, if the ternary and fine-grained tasks are learned jointly.

The research question that this section attempts to answer is the following: Can twitter sentiment classification problems, and fine-grained sentiment classification in particular, benefit from multitask learning? To answer the question, the work done brings the following two main contributions: (i) we show how jointly learning the ternary and fine-grained sentiment classification problems in a multitask setting improves the state-of-the-art performance,[5] and (ii) we demonstrate that recurrent neural networks outperform models previously proposed without access to huge corpora while being flexible to incorporate different sources of data.

6.2.2 The Experimental Framework

In his work, Caruana [30] proposed a multitask approach in which a learner takes advantage of the multiplicity of interdependent tasks while jointly learning them. The intuition is that if the tasks are correlated, the learner can learn a model jointly for them while taking into account the shared information which is expected to improve its generalization ability. People express their opinions online on various

[5] An open implementation of the system for research purposes is available at `https://github.com/balikasg/sigir2017`.

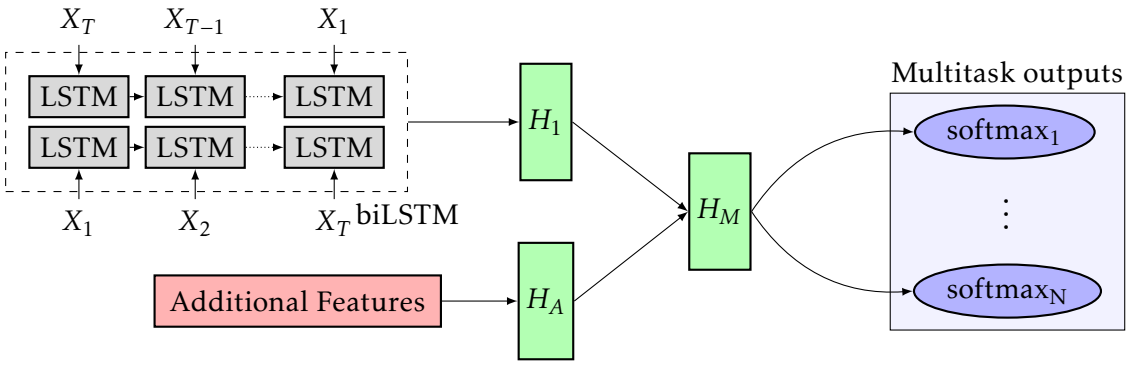


Figure 6.3: The neural network architecture for multitask learning. The biLSTM output is transformed by the hidden layers H_1, H_M and is led to N output layers, one for each of the tasks. The lower part of the network can be used to incorporate additional information.

`VeryNegative`	Beyond frustrated with my #Xbox360 right now, and that as of June, @Microsoft doesn't support it. Gotta find someone else to fix the drive.
`Negative`	@Microsoft Heard you are a software company. Why then is most of your software so bad that it has to be replaced by 3rd party apps?
`Neutral`	@ProfessorF @gilwuvsyou @Microsoft @LivioDeLaCruz We already knew the media march in ideological lockstep but it is nice of him to show it.
`Positive`	PAX Prime Thursday is overloaded for me with @Microsoft and Nintendo indie events going down. Also, cider!!! :p
`VeryPositive`	I traveled to Redmond today. I'm visiting with @Microsoft @SQLServer engineers tomorrow - at their invitation. Feeling excited.

Table 6.3: The example demonstrates the different levels of sentiment a tweet may convey. Also, note the Twitter-specific use of language and symbols.

subjects (events, products..), on several languages and in several styles (tweets, paragraph-sized reviews..), and it is exactly this variety that motivates the multitask approaches. Specifically for Twitter for instance, the different settings of classification like binary, ternary and fine-grained are correlated since their difference lies in the sentiment granularity of the classes which increases while moving from binary to fine-grained problems.

There are two main decisions to be made in our approach: the learning algorithm, which learns a decision function, and the data representation. With respect to the former, neural networks are particularly suitable as one can design architectures with different properties and arbitrary complexity. Also, as training a neural network usually relies on back-propagation of errors [155], one can have shared parts of the network trained by estimating errors on the joint tasks and others specialized for particular tasks. Concerning the data representation, it strongly depends on the data type available. For the task of sentiment classification of tweets with neural networks, distributed embeddings of words have shown great potential. Embeddings are defined as low-dimensional, dense representations of words that can be obtained in an unsupervised fashion by training on large quantities of text [146].

Concerning the neural network architecture, we focus on Recurrent Neural Networks (RNNs) that are capable of modeling short-range and long-range dependencies like those exhibited in sequence data of arbitrary length like text. While in the traditional information retrieval paradigm such dependencies are captured using n-grams and skip-grams, RNNs learn to capture them automatically [51]. To circumvent the problems with capturing long-range dependencies and preventing gradients from vanishing, the long short-term memory network (LSTM) was proposed [81]. In this work, we use an extended version of LSTM called bidirectional LSTM (biLSTM). While standard LSTMs access information only from the past (previous words), biLSTMs capture both past and future information effectively [87, 51]. They consist of two LSTM networks, for propagating text forward and backward with the goal being to capture the dependencies better. Indeed, previous work on multitask learning showed the effectiveness of biLSTMs in a variety of problems: [2] tackled sequence prediction, while [148] and [92] used biLSTMs for Named Entity Recognition and dependency parsing respectively.

Figure 6.3 presents the architecture we use for multitask learning. In the top-left of the figure a biLSTM network (enclosed by the dashed line) is fed with embeddings $\{X_1, \ldots, X_T\}$ that correspond to the T words of a tokenized tweet. Notice,

as discussed above, the biLSTM consists of two LSTMs that are fed with the word sequence forward and backwards. On top of the biLSTM network one (or more) hidden layers H_1 transform its output. The output of H_1 is led to the softmax layers for the prediction step. There are N softmax layers and each is used for one of the N tasks of the multitask setting. In tasks such as sentiment classification, additional features like membership of words in sentiment lexicons or counts of elongated/capitalized words can be used to enrich the representation of tweets before the classification step [93]. The lower part of the network illustrates how such sources of information can be incorporated to the process. A vector "Additional Features" for each tweet is transformed from the hidden layer(s) H_A and then is combined by concatenation with the transformed biLSTM output in the H_M layer.

Our goal is to demonstrate how multitask learning can be successfully applied on the task of sentiment classification of tweets. The particularities of tweets are to be short and informal text spans. The common use of abbreviations, creative language etc., makes the sentiment classification problem challenging. To validate our hypothesis, that learning the tasks jointly can benefit the performance, we propose an experimental setting where there are data from two different twitter sentiment classification problems: a fine-grained and a ternary. We consider the fine-grained task to be our primary task as it is more challenging and obtaining bigger datasets, *e.g.* by distant supervision, is not straightforward and, hence we report the performance achieved for this task. As a result, unless otherwise stated, we optimize for the performance on the fine-grained classification tasks. For completeness, however, we also report the performance we obtain for the ternary classification task.

The data Ternary and fine-grained sentiment classification were part of the SemEval 2016[6] "Sentiment Analysis in Twitter" task [133]. We use the high-quality datasets the challenge organizers released.[7] The dataset for fine-grained classification is split in training, development, development_test and test parts. In the rest, we refer to these splits as `train`, `development` and `test`, where `train` is composed by the training and the development instances. Table 6.4 presents an overview of the data. As discussed in [133] and illustrated in the Table, the fine-grained dataset is highly unbalanced and skewed towards the positive sentiment: only 13.6% of the training examples are labeled with one of the negative classes.

[6] `http://alt.qcri.org/semeval2016/task4/`

[7] The datasets are those of Subtasks A and C, available at `http://alt.qcri.org/semeval2016/task4/`.

		$\|D\|$	VeryNeg.	Neg.	Neutr.	Pos.	VeryPos.
Ternary	Train	5,500	-	785	1,887	2,828	-
	Test	20,632	-	3,231	10,342	7059	-
Fine-Grained	Train	7,292	111	884	2,019	3,726	432
	Dev.	1,778	29	204	533	887	125
	Test	20,632	138	2,201	10,081	7,830	382

Table 6.4: Cardinality and class distributions of the datasets.

Feature representation We report results using various feature sets. The first one, dubbed bow is the commonly used bow of words representation of the tweets. The representation is motivated by our previous work [10], where we showed that using n-grams with $n \in \{1, 2, 3\}$ as well as character-grams of size 4 and 5 benefits sentiment classification. Here, we use all the possible n-grams and character-grams hashed in vectors of dimension 20K and 25K respectively. The second, dubbed nbow, is a neural bag-of-words that uses text embeddings to generate low-dimensional, dense representations of the tweets. To construct the nbow representation, given the word embeddings dictionary where each word is associated with a vector, we apply the average compositional function that averages the embeddings of the words that compose a tweet. Simple compositional functions like average were shown to be robust and efficient in previous work [129]. Instead of training embeddings from scratch, we use the pre-trained on tweets GloVe embeddings of [146].[8] In terms of resources required, using only nbow is efficient as it does not require any domain knowledge. However, previous research on sentiment analysis showed that using extra resources, like sentiment lexicons, can benefit significantly the performance [93, 10]. To validate this and examine at which extent neural networks and multitask learning benefit from such features we evaluate the models using an augmented version of nbow, dubbed nbow+. The feature space of the latter, is augmented using 1,368 extra features consisting mostly of counts of punctuation symbols ('!?#@'), emoticons, elongated words and word membership features in several sentiment lexicons. The next paragraph details those extra features. Also, bow+ refers to the concatenation of these extra features with the bow representations introduced above.

Feature Engineering Similar to [93] we extracted features based on the lexical content of each tweet and we also used sentiment-specific lexicons. The features

[8] urlhttp://nlp.stanford.edu/data/glove.twitter.27B.zip

extracted for each tweet include:

- number of exclamation marks, number of question marks, number of both exclamation and question marks,
- number of words written in capitals and number of elongated words, that is words with more than three occurrences of a letter such as "cooool" and "bliiiiah",
- number of negative words in a tweet,
- number of positive emoticons, number of negative emoticons and a binary feature indicating if emoticons exist in a given tweet, and
- distribution of Part-of-speech (POS) tags [66] and distribution of POS tags in a positive and negative contexts. We consider words to occur in a negative context if a negative word proceeds then. A negative context stops when another negative words, punctuation or the end of the tweet is met.

With regard to the sentiment lexicons, we used:

- manual sentiment lexicons: the Bing liu's lexicon [86], the NRC emotion lexicon [131], and the MPQA lexicon [187],
- # of words in positive and negative context belonging to the word clusters provided by the CMU Twitter NLP tool[9]
- positional sentiment lexicons: `sentiment 140 lexicon`[10] [68] and the Hashtag Sentiment Lexicon [93]

We make, here, more explicit the way we used the sentiment lexicons, using the Bing Liu's lexicon as an example. We treated the rest of the lexicons similarly. For each tweet, using the Bing Liu's lexicon we obtain a 104-dimensional vector. After tokenizing the tweet, we count how many words (i) in positive/negative contexts belong to the positive/negative lexicons (4 features) and we repeat the process for the hashtags (4 features). To this point we have 8 features. We generate those 8 features for the lowercase words and the uppercase words. Finally, for each of the 24 POS tags the [66] tagger generates, we count how many words in positive/negative contexts belong to the positive/negative lexicon. As a results, this generates $2 \times 8 + 24 \times 4 = 104$ features in total for each tweet.

[9] `http://www.cs.cmu.edu/~ark/TweetNLP/`

[10] For a collection of sentiment lexicons the interested reader can refer to `http://saifmohammad.com/WebPages/lexicons.html`.

Evaluation measure To reproduce the setting of the SemEval challenges [133], we optimize our systems using as primary measure the macro-averaged Mean Absolute Error (MAE_M) given by:

$$MAE_M = \frac{1}{|C|}\sum_{j=1}^{|C|}\frac{1}{|\mathrm{Te}_j|}\sum_{x_i \in \mathrm{Te}_j}|h(x_i) - y_i|$$

where $|C|$ is the number of categories, Te_j is the set of instances whose true class is c_j, y_i is the true label of the instance x_i and $h(x_i)$ the predicted label. The measure penalizes decisions far from the true ones and is macro-averaged to account for the fact that the data are unbalanced. Complementary to MAE_M, we report the performance achieved on the micro-averaged F_1 measure, which is a commonly used measure for classification.

The models To evaluate the multitask learning approach, we compared it with several other models. Support Vector Machines (SVMs) are maximum margin classification algorithms that have been shown to achieve competitive performance in several text classification problems [133]. $\texttt{SVM}_{\texttt{ovr}}$ stands for an SVM with linear kernel and an one-vs-rest approach for the multi-class problem. Also, $\texttt{SVM}_{\texttt{cs}}$ is an SVM with linear kernel that employs the crammer-singer strategy [41] for the multi-class problem. Logistic regression (LR) is another type of linear classification method, with probabilistic motivation. Again, we use two types of Logistic Regression depending on the multi-class strategy: $\texttt{LR}_{\texttt{ovr}}$ that uses an one-vs-rest approach and multinomial Logistic Regression also known as the `MaxEnt` classifier that uses a multinomial criterion.

Both SVMs and LRs as discussed above treat the problem as a multi-class one, without considering the ordering of the classes. For these four models, we tuned the hyper-parameter C that controls the importance of the L_2 regularization part in the optimization problem with grid-search over $\{10^{-4}, \ldots, 10^4\}$ using 10-fold cross-validation in the union of the training and development data and then retrained the models with the selected values. Also, to account for the unbalanced classification problem we used class weights to penalize more the errors made on the rare classes. These weights were inversely proportional to the frequency of each class. For the four models we used the implementations of Scikit-learn [143].

For multitask learning we use the architecture shown in Figure 6.3, which we implemented with Keras [34]. The embeddings are initialized with the 50-dimensional GloVe embeddings while the output of the biLSTM network is set to

dimension 50. The activation function of the hidden layers is the hyperbolic tangent. The weights of the layers were initialized from a uniform distribution, scaled as described in [67]. We used the Root Mean Square Propagation optimization method. We used dropout for regularizing the network. We trained the network using batches of 128 examples as follows: before selecting the batch, we perform a Bernoulli trial with probability p_M to select the task to train for. With probability p_M we pick a batch for the fine-grained sentiment classification problem, while with probability $1 - p_M$ we pick a batch for the ternary problem. As shown in Figure 6.3, the error is backpropagated until the embeddings, that we fine-tune during the learning process. Notice also that the weights of the network until the layer H_M are shared and therefore affected by both tasks.

To tune the neural network hyper-parameters we used 5-fold cross validation. We tuned the probability p of dropout after the hidden layers H_M, H_1, H_A (cf. Fig. 6.3) and for the biLSTM for $p \in \{0.2, 0.3, 0.4, 0.5\}$, the size of the hidden layer $H_M \in \{20, 30, 40, 50\}$ and the probability p_M of the Bernoulli trials from $\{0.5, 0.6, 0.7, 0.8\}$.[11] During training, we monitor the network's performance on the development set and apply early stopping if the performance on the validation set does not improve for 5 consecutive epochs.

6.2.3 Experimental results

Fine-grained problem Table 6.8 illustrates the performance of the models for the different data representations. The upper part of the Table summarizes the performance of the baselines. The entry "Balikas et al." stands for the winning system of the 2016 edition of the challenge [10], which to the best of our knowledge holds the state-of-the-art. Due to the stochasticity of training the biLSTM models, we repeat the experiment 10 times and report the average and the standard deviation of the performance achieved.

Several observations can be made from the table. First notice that, overall, the best performance is achieved by the neural network architecture that uses multitask learning. This entails that the system makes use of the available resources efficiently and improves the state-of-the-art performance. In conjunction with the fact that we found the optimal probability $p_M = 0.5$, this highlights the benefits of multitask learning over single task learning. Furthermore, as described above, the

[11] Overall, we cross-validated 512 combinations of parameters. The best parameters were: 0.2 for all dropout rates, 20 neurons for H_M and $p_M = 0.5$.

neural network-based models have only access to the training data as the development are hold for early stopping. On the other hand, the baseline systems were retrained on the union of the train and development sets. Hence, even with fewer resources available for training on the fine-grained problem, the neural networks outperform the baselines. We also highlight the positive effect of the additional features that previous research proposed. Adding the features both in the baselines and in the biLSTM-based architectures improves the MAE_M scores by several points.

Lastly, we compare the performance of the baseline systems with the performance of the state-of-the-art system of [10]. While [10] uses n-grams (and character-grams) with $n > 1$, the baseline systems (SVMs, LRs) used in this work use the nbow+ representation, that relies on unigrams. Although they perform on par, the competitive performance of nbow highlights the potential of distributed representations for short-text classification. Further, incorporating structure and distributed representations leads to the gains of the biLSTM network, in the multitask and single task setting.

Similar observations can be drawn from Figure 6.4 that presents the F_1 scores. Again, the biLSTM network with multitask learning achieves the best performance. It is also to be noted that although the two evaluation measures are correlated in the sense that the ranking of the models is the same, small differences in the MAE_M have large effect on the scores of the F_1 measure.

Ternary problem Complementary to the performance on the fine-grained sentiment classification problem discussed above, we report the performance on the ternary task. Again, in order to replicate the setting of the SemEval 2016 challenges, we use the evaluation measure proposed by the challenge organizers that is the F_1 measure, calculated only for the positive and the negative categories. Table 6.6 summarizes the results for the baselines and the multitask architecture. An important detail before commenting on the performance achieved is that the values of the hyperparameters are those we found optimal for the fine-grained task, as that was our main task for the study. There are several observations from the table.

First, adding the additional features (bow and nbow compared to bow+ and nbow+) benefits every system we tested. In general, the representations that are based on the word embeddings perform better than those based on the bag-of-words representations. Second, the neural network architectures perform better

	bow	nbow	bow+	nbow+
SVM_{ovr}	0.993	0.840	0.786	0.714
SVM_{cs}	0.941	0.946	0.746	0.723
LR_{ovr}	0.965	0.836	0.731	0.712
MaxEnt	0.946	0.842	0.701	0.715
Balikas and Amini [10]	-	-	-	0.719
biLSTM (single task)	0.827±0.017		0.694±0.04	
biLSTM+Multitask	0.786±0.025		**0.685**±0.024	

Table 6.5: The scores on MAE_M for the systems. The best (lowest) score is shown in bold and is achieved in the multitask setting with the biLSTM architecture of Figure 6.3.

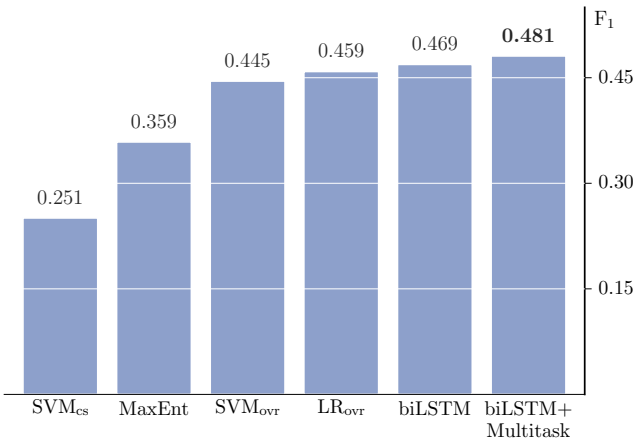


Figure 6.4: F_1 scores using the nbow+ representations. The best performance is achieved with the multitask setting.

than the traditional classification systems like SVMs and Logistic Regression. Notice that the entry "Deriu et al." [46] is the winner of the task who also used neural networks, and in particular, an ensemble of convolutional neural networks. Lastly, and perhaps most importantly, the improvements due to multitask learning are very small and therefore we can not claim any important gain. We claim however that the models perform on par and we highlight that the performance on the ternary task did not decrease while the performance in the fine-grained task increased.

6.2.4 Summary

In this section, we showed that by jointly learning the tasks of ternary and fine-grained classification with a multitask learning model, one can greatly improve the performance on the second. This opens several avenues for future research. The

	bow	nbow	bow+	nbow+
SVM_{ovr}	0.506	0.572	0.584	0.600
SVM_{cs}	0.499	0.546	0.576	0.590
LR_{ovr}	0.520	0.572	0.601	0.600
MaxEnt	0.510	0.547	0.596	0.593
Deriu et al. [46]	-	-	-	0.633
biLSTM (single task)	0.580±0.02		0.613±0.04	
biLSTM+Multitask	**0.582**±0.03		**0.617**±0.05	

Table 6.6: The scores of F_1 measure for the systems for the ternary classification task. The hyperparameter tuning was performed for the fine-grained task.

first and perhaps the most straightforward would be to verify whether our findings generalize or further improve using data of different type and language. Since sentiment is expressed in different textual types like tweets and paragraph-sized reviews, in different languages (English, German, ..) and in different granularity levels (binary, ternary,..) one can imagine multitask approaches that could benefit from combining such resources.

We showed that by using a multitask learning architecture we managed to gain in terms of performance when tuning for the fine-grained problem. For the ternary problem however we did not observe similar benefits when enabling multitask learning: there are some marginal performance improvements that do not allow for claiming significant improvements. A direct extension would be to compare this outcome with the case where the fine-tuning is performed for the ternary problem. Although this is an interesting task from an experimental point of view, one can directly improve on the ternary task using distant supervision in the form of pseudo-labeling tweets based on emoticons, following for instance the work of [46].

Another research line would be to examine a hierarchical decision function in the output of the network, perhaps motivated by common hierarchical text classification systems [168]. That way, one could apply a two-level prediction mechanism for the fine-grained task: first predict the ternary task (positive, negative, neutral) and then, in the second level, the intensity of the sentiment: between VeryPositive and Positive for the positive sentiment or between VeryNegative and Negative for the negative sentiment. Increasing the data size for the decisions of the first level by combining the ternary and fine-grained datasets should result in performance gains provided that the distribution over the classes does not change.

Exploring and comparing such as approach with multitask learning is part of our future work.

Lastly, while we opted for biLSTM networks here, one could use convolutional neural networks or even try to combine different types of networks and tasks to investigate the performance effect of multitask learning. Convolutions neural networks have been shown to be effective for text classification and comparing their performance with LSTMs as well as the effect of multitask learning is an interesting extension.

6.3 Cross-lingual text retrieval

As we have previously discussed and demonstrated in this dissertation, word embeddings have shown great potential in several natural language processing tasks [124, 126, 146, 108, 109]. Their ability to capture syntactic properties as well as their property to project semantically similar words close in the induced vector space have been particularly celebrated as they alleviate limitations of the discrete representations of words based on the vector space model [156]. The success of monolingual word embeddings, helped to develop new approaches for multilingual word embeddings. Additionally to projecting similar words (or short multiword expressions) of a single language close in the inferred vector space, multilingual embeddings project similar words across languages close in the shared vector space. Depending on the available resources and the approaches employed to learn the embeddings, there are different methods [95, 188, 71, 79, 160, 125, 189, 64, 163, 179, 166].

In the previous sections, we have also discussed that methods of composition for obtaining representations of large text spans on top of word embeddings are particularly important. To this end, in Section 6.1 we demonstrated how averaging word embeddings performs while in Section 6.2 we relied on recurrent neural networks and LSTMs in particular. In this section we are revisiting this problem by proposing a completely different approach: we argue that in several tasks having explicit representations of large spans is not necessary as long as an efficient way for obtaining distances between spans exists.

In this section, we adapt to the cross-lingual setting the work of [100], who proposed a clever re-parametrization of the transportation problem without hyperparameters for calculating document distances from cross-lingual word embeddings. Meanwhile, we will show that a simple modification of the optimization problem

allows the framework to incorporate term-weighting schemes. To validate the proposed approach, we conducted extensive experiments for cross-lingual document discovery (CLDD) in six cross-lingual settings where given a query document (e.g., English Wikipedia entry for "Dog") one needs to retrieve its corresponding document in another language (e.g., French entry for "Chien"). The novel method outperforms strong baselines previously proposed for the task by a large margin. Notably, we demonstrate the impact of the quality of the embeddings used, as well as the impact of three term weighting schemes in terms of Mean Reciprocal Rank and Precision at 1.

6.3.1 A Wasserstein-alike distance for Cross-lingual Document Retrieval

In this section, we demonstrate how calculating document distances can be seen as an instance of the Earth Mover's Distance problem [100].

Notation We assume access to the collections $\mathcal{C}^{\ell_1} = \{d_1^{\ell_1}, \ldots, d_N^{\ell_1}\}$ and $\mathcal{C}^{\ell_2} = \{d_1^{\ell_2}, \ldots, d_M^{\ell_2}\}$ where $d_i^{\ell_1}$ (resp. $d_i^{\ell_2}$) is the i-th document written in language ℓ_1 (resp. ℓ_2). Let the vocabulary size of the two languages be denoted as V^{ℓ_1} and V^{ℓ_2}. For the rest of the development we assume to have access to dictionaries of embeddings $\boldsymbol{E}^{\ell_1}, \boldsymbol{E}^{\ell_2}$ where words from ℓ_1 and ℓ_2 are projected into a *shared* vector space of dimension D, hence $\boldsymbol{E}^{\ell_1} \in \mathbb{R}^{V^{\ell_1} \times D}, \boldsymbol{E}^{\ell_2} \in \mathbb{R}^{V^{\ell_2} \times D}$ and $E_k^{\ell_1}$, $E_j^{\ell_2}$ denote the embeddings of words k, j. As learning the bilingual embeddings is not the focus of this paper, any of the previously proposed methods can be used. A document consists of words and is represented using the Vector Space Model with frequencies. Hence, $\forall i : d_i^{\ell_1} \in \mathbb{R}^{V^{\ell_1}}$, $d_i^{\ell_2} \in \mathbb{R}^{V^{\ell_2}}$ and $d_{ij}^{\ell_1}$ is the frequency of the j-th word of document $d_i^{\ell_1}$. Importantly, the vector representations of the documents need to be l_1-normalized. Calculating the distance of words in the embeddings space is naturally achieved using the Euclidean distance with lower values meaning that words are similar between them. For the rest, we denote by $c(k, j) = \|E_k^{\ell_1} - E_j^{\ell_2}\|_2$ the Euclidean distance between the words k and j in the embedding's space. Our goal is the following: given two documents $d_n^{\ell_1}, d_m^{\ell_2}$ in two languages, estimate the distance between them by utilizing the expressiveness of their word embeddings.

The distance between two documents depends on the distances between the words they consist of. In the Earth Mover's Distance setting, the words of $d_n^{\ell_1}$ can be considered as piles and the words of $d_m^{\ell_2}$ as holes of earth in the D-dimensional space of the embeddings. The amount of earth in the piles and holes is described by each word's frequency. Any word of $d_n^{\ell_1}$ can be transformed to any of the words

of $d_m^{\ell_2}$ either in total or in parts. A matrix $T \in \mathbb{R}^{V^{\ell_1} \times V^{\ell_2}}$ composed by non-negative T_{jk} elements describes how much earth from the pile of the word $d_{nj}^{\ell_1}$ is moved to the hole of the word $d_{mk}^{\ell_1}$. To transform $d_n^{\ell_1}$ to $d_m^{\ell_2}$ for the outgoing and ingoing earth flows should be $\sum_k T_{jk} = d_{nj}^{\ell_1}$ and $\sum_j T_{jk} = d_{mk}^{\ell_2}$, which intuitively means that every word must be transformed. Therefore, the linear optimization problem writes:

$$\begin{aligned} &\min \sum_{j=1}^{V^{\ell_1}} \sum_{k=1}^{V^{\ell_2}} T_{jk} c(j,k) \\ &\text{subject to:} \sum_{k=1}^{V^{\ell_2}} T_{jk} = d_{nj}^{\ell_1}, \ \forall j \in \{1,\dots,V^{\ell_1}\} \\ &\sum_{j=1}^{V^{\ell_1}} T_{jk} = d_{mk}^{\ell_2}, \ \forall k \in \{1,\dots,V^{\ell_2}\} \end{aligned} \tag{6.1}$$

As transforming the words of $d_n^{\ell_1}$ to $d_m^{\ell_2}$ comes with the cost $c(k,j)$, the optimization problem of Eq. (6.1) translates to the minimization of the associated cumulative cost of transforming all the words. The value of the minimal cost is the distance between the documents. Intuitively, the more similar the words between the documents are, the lower will be the costs associated to the solution of the optimization problem, which, in turn, signifies smaller document distances. For example, given "the cat sits on the mat" and its French translation "le chat est assis sur le tapis", the weights (earth piles and holes) after stopwords filtering of "cat", "sits", "mat", and "chat", "assis", "tapis" will be 1/3. Given high-quality embeddings, solving Eq. (6.1) will converge to the one-to-one transformations "cat↔chat", "sits↔assis" and "mat↔tapis", with very low cumulative cost as the paired words are similar.

The problem of Eq. (6.1) is a special case of the earth mover's distance [153]. Following [100] , we refer to it as word mover's distance (WMD) for bilingual collections. Since $c(k,j)$ is a metric, WMD for bilingual collections is a metric [153].

The optimization problem of Eq. (6.1) requires that the vector representations of the documents $\boldsymbol{d}_n^{\ell_1}$, $\boldsymbol{d}_m^{\ell_2}$ are l_1-normalized. Therefore, without loss of generality one may apply any term weighting scheme (that guarantees non-negative vector elements) prior to the l_1-normalization. In this work we investigate three schemes: *Term frequency* (*tf*), that represents a document using the frequency of its word occurrences.

The *term frequency-inverse document frequency* weighting scheme (*idf*), where the term frequencies are multiplied by the words inverse document frequencies:

$$tf\text{-}idf(t,d) = tf(t,d) \times \log \frac{N+1}{df(t)+1}.$$

In a collection of N documents, the document frequency $df(t)$ is the number of documents in the collection containing the word t. The inverse document's frequency idf penalizes words that occur in many documents and is smoothed in order to prevent uninformative terms that occur in all of the documents of a collection (smoothing of the numerator); and hence to avoid zero-divisions (smoothing of the denominator).

The *graph of words* (*gow*) document representation [151, 152]. Following the process of Sec. 4 of [152] we represent documents by unweighted directed graphs constructed using a sliding window. Then, the word weights are: $tw\text{-}idf(t,d) = tw(t,d) \times \log \frac{N+1}{df(t)+1}$ where $tw(t,d)$ is the in-degree (number of incoming edges) of the term t in the graph of d. *Gow* captures long term dependences (depending on the sliding window size) and the order of the terms (the graph is directed). The terms weight tw increases with the number of *contexts* the term occurs with, which was shown to be a robust signal for the term's importance. Following [22], we set the window size to 6 hereafter.

6.3.2 The Experimental Framework

The goal of CLDD is to identify corresponding documents written in different languages. Assuming, for instance, English and French Wikipedia documents, the goal is to identify the cross-language links between the articles. The challenge is to quantify the cross-lingual document distances. Traditional retrieval approaches employing bag-of-words representations perform poorly in CLDD as the vocabularies vary across language, and words from different languages rarely co-occur.

CLDD decomposes as follows: for each article from $\mathcal{C}^{\ell_1}$, one needs to retrieve the corresponding article from another collection $\mathcal{C}^{\ell_2}$. We derive three bilingual datasets using the inter-language links of Wikipedia: (i) English-French, (ii) English-German, and, (iii) French-German, which define six CLDD problems. We consider the inter-language links to be our golden standard that will be used for calculating the evaluation measures. In the pre-processing steps we lowercase the documents, we remove stopwords, punctuation and words that occur less than three times and apply the Stanford Part-of-Speech tagger [171] to keep only the nouns, which was

	CLDD			BiLDA Training		
	$\text{Wiki}_{\text{En-Fr}}$	$\text{Wiki}_{\text{En-Ge}}$	$\text{Wiki}_{\text{Fr-Ge}}$	$\text{Wiki}_{\text{En-Fr}}$	$\text{Wiki}_{\text{En-Ge}}$	$\text{Wiki}_{\text{Fr-Ge}}$
instances	500	500	500	20,000	20,000	20,000
V^{ℓ_1}	26,406	26,406	12,533	434,807	434,807	204,301
V^{ℓ_2}	12,533	32,353	32,353	204,301	576,294	576,294
W^{ℓ_1}	135,509	135,509	49,690	8.43M	8.43M	3.27M
W^{ℓ_2}	49,690	98,053	98,053	3.27M	5.56M	5.56M

Table 6.7: Statistics for the data used. W^{ℓ} denotes the size of the corpora measured in words.

recommended at [27] and we also found to improve the results in our preliminary experiments. Table 6.7 (under "CLDD") summarizes these datasets.

To evaluate WMD we use both in-house and publicly available pre-trained embeddings. First, we use the open implementation of BilBOWA[12] [71] with its default parameters.[13] BilBOWA benefits from large monolingual datasets and requires a smaller set of sentence-aligned parallel data to learn the bilingual embeddings. We construct the parallel data using the English-French and English-German parts of the Europarl v7 data [96]: using English as the pivot language we first construct a French-German parallel corpus and then for the English-French and English-German parallel corpora we only keep the sentences that occur in the French-German corpus. This results in $\sim$ 1.7M aligned sentences. Lastly, we compiled monolingual corpora with 900K articles with the same protocol using the English-French and English-German Wikipedia dumps.[14]

As far as pre-trained embeddings are concerned, we used the state-of-the-art embeddings[15] of [166], dubbed *cn* from ConceptNet Numberbatch hereafter. [166] proposed to learn word embeddings by combining distributional semantics and ConceptNet v5.5 [112] using a generalization of the retrofitting method of [56]. We expect those embeddings to perform substantially better than those we trained with BilBOWA as much more resources were used to learn them.

The systems Previous work found the bilingual Latent Dirichlet Allocation (BiLDA) [44, 137, 127, 178] which is an extension of the Latent Dirichlet Allocation [23] in the bilingual setting to yield state-of-the-art results [61, 194, 177, 185]. BiLDA is trained on either parallel or comparable corpora and learns aligned per-word topic

[12] https://github.com/gouwsmeister/bilbowa

[13] -window 5 -sample 1e-4 -negative 5 -binary 0 -adagrad 1 -xling-lambda 1

[14] http://linguatools.org/tools/corpora/wikipedia-comparable-corpora/

[15] https://github.com/commonsense/conceptnet-numberbatch

	En→Fr		Fr→En		En→Ge		Ge→En		Fr→Ge		Ge→Fr	
	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1
$\texttt{nBOW}^{\texttt{cn}}_{\texttt{tf}}$	.370	.244	.330	.210	.495	.388	.408	.274	.448	.362	.340	.262
$\texttt{nBOW}^{\texttt{cn}}_{\texttt{gow}}$	.366	.266	.305	.180	.486	.376	.423	.286	.476	.400	.348	.270
$\texttt{nBOW}^{\texttt{cn}}_{\texttt{idf}}$	.381	.278	.354	.240	.494	.380	.423	.288	.479	.394	.371	.284
BiLDA	.552	.446	.370	.288	.640	.546	.603	.502	.354	.258	.513	.398
$\texttt{Trsl}_{\texttt{tf}}$	.530	.430	.532	.424	.575	.472	.495	.380	.451	.348	.376	.256
$\texttt{Trsl}_{\texttt{gow}}$	.629	.548	.629	.544	.703	.622	.639	.538	.507	.408	.462	.360
$\texttt{Trsl}_{\texttt{idf}}$	.612	.504	.617	.506	.664	.572	.598	.482	.534	.418	.459	.322
$\texttt{WMD}^{\texttt{bw}}_{\texttt{tf}}$	.417	.336	.647	.600	.753	.714	.782	.736	.615	.564	.396	.332
$\texttt{WMD}^{\texttt{bw}}_{\texttt{gow}}$	.402	.326	.588	.542	.723	.688	.755	.720	.532	.480	.375	.308
$\texttt{WMD}^{\texttt{bw}}_{\texttt{idf}}$	.510	.436	.699	.656	.815	.786	.859	.838	.655	.608	.499	.432
$\texttt{WMD}^{\texttt{cn}}_{\texttt{tf}}$	.782	.724	.739	.684	.871	.838	.873	.834	.671	.610	.691	.630
$\texttt{WMD}^{\texttt{cn}}_{\texttt{gow}}$	.769	.718	.713	.662	.875	.841	.875	.838	.655	.606	.713	**.662**
$\texttt{WMD}^{\texttt{cn}}_{\texttt{idf}}$	**.809**	**.760**	**.782**	**.734**	**.899**	**.870**	**.906**	**.878**	**.710**	**.660**	**.719**	.658

Table 6.8: The scores achieved by the systems. WMD with ConceptNet Numberbatch embeddings and the idf term weighting scheme outperforms the rest by an important gap. $\ell_1 \rightarrow \ell_2$ (*e.g.* En→Fr) defines a CLDD problem where the query documents are written in ℓ_1 (*e.g.* En) and the retrieved in ℓ_2 (*e.g.* Fr).

distributions between two or more languages that best explain the latent themes of a collection. During inference it projects unseen documents in the *shared* topic space. In the rest, we compare:

-nBOW that represents documents by a weighted average of their words' embeddings [129, 21].

-BiLDA with 300 topics and collapsed Gibbs sampling for inference implemented using Numpy [180], with $\alpha = 50/K$ and $\beta = 0.01$. We let 200 Gibbs sampling iterations for burn-in and then sample the document distributions each 25 iterations until the $500th$ Gibbs iteration. For learning the topics we used 30K comparable Wikipedia documents (Table 6.7 under "BiLDA training") which is an order of magnitude bigger than what previous work used [177].

-Trsl that is a system that is based on off-the-shelf translation. Since in the pre-processing steps we keep only nouns, we use dictionaries between the language pairs in order to translate the documents to be retrieved in the the language of the query document. Given the translations, one may apply the different weighting schemes during vectorization. Having the vectors of the query and translated documents, we rank them according to the Euclidean distance. To generate the dictionaries we rely on Wiktionary and in particular on the methods of [1, 198].[16] Since the translation process is based on dictionaries it is possible for a given word to have several translations: in this case we sort the words given their unigram probabilities calculated on the comparable Wikipedia datasets of Table 6.7 under "BiLDA training" and select the most frequent translation.

-WMD that is the proposed metric implemented with Scikit-learn [143] and PyEMD [144, 145]. For nBOW and WMD we compare three weighting schemes (*tf*, *idf*, *gow*) and two types of embeddings (*bw*, *cn*). Hence, $\text{WMD}^{\text{cn}}_{\text{idf}}$ applies idf term weights and uses *cn* embeddings.

Results As evaluation measures we report the Mean Reciprocal Rank (MRR) [176] and the Precision at 1 (P@1) scores. MRR accounts for the rank of the correct answer in the returned documents. When only a single document is relevant P@1 counts how many times the correct document is returned at rank 1.

Table 6.8 presents the scores achieved. First, notice that the results clearly establish the important performance improvements of WMD over the rest of the methods. In particular, WMD with the BilBOWA embeddings outperforms BiLDA in every CLDD setting except for the "En→Fr" and "Ge→Fr". Further switching from BilBOWA to ConceptNet Numberbatch embeddings boosts the performance of WMD

[16] We use the open implementation of https://github.com/juditacs/wikt2dict.

significantly, achieving the best performances by very large margins for each of the six CLDD settings. This highlights the importance of high-quality embeddings. BiLDA that was found effective by previous work, outperforms nBOW, although the latter uses the ConceptNet Numberbatch embeddings that showed a great potential with WMD. While this is probably due to the weighted averaging operations which result in information loss for long documents, it further highlights the suitability of WMD for calculating document distances.

The best performing baseline system is probably Trs1 that relies on an off-the-shelf translation approach. Notably, Trs1 performs better than WMD^{bw} for several language pairs. Moreover, the best performing weighting scheme for this system is graph-of-words. It outperforms idf, that was found to be the most robust weighting scheme for the rest of the systems, by several points in both measures for most of the language pairs. The main limitation of Trs1 concerns the out-of-vocabulary words (OOV). As with WMD we ignored OOV words. We believe however that using a more robust technique for treating OOV can result in obtaining even higher scores. To this direction, an interesting approach would be to use either subword information and select for instance the translation of the word that is most similar to the OOV or use a system that would make queries to a search tool like Google to find the translation.

Another observation concerns the impact of the term weighting schemes on the performance achieved by the WMD approach. The gow scheme performs better than tf counts in most of the cases, especially when the ConceptNet Numberbatch embeddings are used. Overall, idf is the best performing weighting scheme. As an interesting direction of future work, one may improve the results of the gow by tuning parameters like the sliding window size used to construct the graph in order to generate graphs that capture the syntactic rules of the languages.

6.3.3 Summary

In this work we adapted the Word Movers Distance metric for CLDD. Our results demonstrate the effectiveness of the proposed approach which we attribute to the ability of the model to quantify document similarity using word level information and high quality word embeddings. Our study open avenues for future research.

First, one could further improve the obtained results by tuning the parameters of the weighting schemes or adapt the proposed approach to other multilingual and cross-lingual tasks like multilingual document clustering. Second, the

promising results achieved suggest that tasks like CLDD or document classification (used in [100]) are a good fit for comparing methods for learning embeddings.

A last promising research avenue would be to incorporate the distance estimation between documents that `WMD` provides into topic models. Our findings in Chapter 5 suggest that the quality of the learned topics when incorporating cross-lingual word embeddings improves and we showcases this potential by computing a similarity metric between documents based on the cosine between document representations. The latter were generated by averaging the representations of words of a document. Our results here suggest that `WMD` offers a more efficient way for calculating documents distances that considers every words separately instead of the their averaged representations. We believe that integrating this to the presented topic models has the potential to further improve them.

6.4 Chapter Summary

In this chapter we investigated how three text mining tasks can benefit from the use of word embeddings. Our goal for each case was to obtain robust representations that integrate parts of external knowledge. The knowledge source was different for each task:

- In the case of polylingual classification, we assumed access to translations of a documents and used this information to enrich the learned representation using a denoising autoencoder.
- In the case of multitask learning we assumed that there exist two (or more) correlated tasks and we argued that in such a scenario jointly learning the tasks can be beneficial.
- Lastly, for the third task that was cross-lingual information retrieval we suggested that one may not need aggregate word embeddings (which was explicitly done by averaging on via the LSTM network in the first two tasks) to learn a decision function but use a formulation of the transportation problem to derive the distances of documents written in different languages.

Overall, our findings suggest that using word representations learned with models that implement the distributional hypothesis can achieve competitive performance. Furthermore, such models are flexible enough to incorporate different types of knowledge that can further improve performance.

Chapter 7

Concluding Remarks

Text data are ubiquitous, posing a variety of interesting challenges based on a wealth of possible tasks. The goal of this dissertation was to propose, develop and implement models for text mining applications for text data written in a single or in multiple languages. In particular, our contributions target at answering two main questions:

i.) How text structure can improve the performance of unsupervised models for uncovering the latent topics of a documents collections?

ii.) How one can take advantage of polylingual content and rich text representation for improving the performance of various tasks?

Both points constitute important questions for the field of text mining and natural language processing in general. In the following sections we provide an overview of the main contributions of this thesis and discuss possible future research directions.

7.1 Summary of Contributions

The main contributions of the thesis can be summarized as follows.

Text Structure and Topic Models In Chapter 4 we proposed two novel topic models whose goal was to extend LDA by integrating prior knowledge of text structure. We defined the concept of topically coherent segments and we argued that different text spans like frequent n-grams or noun-phrases can be considered coherent. To incorporate this type of knowledge to LDA we proposed two different sampling strategies, one that assigns the same topic to the words of a segments and

a second that uses copulas which allow for more flexibility. Through extensive experimentation on various datasets and tasks we demonstrated that knowledge of text structure is indeed beneficial for topic models. As unsupervised exploration of text collections is an important task with the ever growing amount of data being generated the proposed models can help us to better understand the topics discussed in them and also extract features efficiently from them.

Bilingual Topics Models for Comparable Corpora In Chapter 5 we proposed to better adapt bilingual topic models for comparable corpora with explicit alignments. Motivated by the fact that such corpora are easier to obtain than parallel, we proposed to extend the bilingual Latent Dirichlet Allocation and allow for different topics distributions for the documents of each language. After systematic evaluation of the proposed models, we showed that extending bilingual topic models and adapting them for bilingual collections improves the topical coherence of the learned topics, the generalization performance of the models as well as the performance of the documents representations learned in the task of cross-lingual document retrieval.

Word Embeddings for Text Mining Applications In Chapter 6 we investigated how text mining applications can benefit from word embeddings. Motivated by previous research suggesting that such word representations capture semantic and syntactic word properties we proposed models and algorithms for polylingual classification, multi-task classification and cross-lingual document retrieval. Our results confirmed that the tasks at hand can strongly benefit from rich text representations and efficient models. Our observations complemented seminal results on the fields of representation learning and deep learning for natural language processing tasks about the potential of word embeddings and neural network architectures.

7.2 Future Directions

In this section, we discuss future research directions for the topics covered in the manuscript as well as more broad topics of interest in the areas of text mining and natural language processing in general.

Incorporating Rich Text Structure to Topic Models For the models of Chapter 4 we considered segments to be contiguous words that are observed in

the text. However, one can discover far more structure in documents: from documents to paragraphs, from paragraphs to sentences and then from sentences to parse trees, one may imagine various ways to represent documents as complex structures like trees [29] of graphs [152]. Extending topic models to account for this complex structure is an interesting problem. For instance, copulas can be extended by nesting; nested copulas in turn can model tree-like dependencies which can provide the means for further extending topic models. Another issue, which is rarely touched, is the scalability of such complex models. Addressing the questions "How complexity impacts the obtained results ?" as well as "How scalable are these algorithms for truly big data ?" would provide further useful insights.

Beyond Comparable Corpora with Explicit Alignments The setting we considered in Chapter 5 required input corpora to be in the form of pairs of documents that should be topically aligned. Our motivation was that such corpora are more common that parallel. However, in the most general case one has only access to comparable corpora without any type of alignments. Can we apply the findings of this thesis as well as those of related work to address the challenges of this exciting setting? We believe that by combining the models of Chapter 5 with the cross-lingual document retrieval process outlined in Chapter 6 is a promising research direction.

Word Embeddings for Text Mining Applications In Chapter 6 we explored three different text mining applications. For the case of polylingual classification we validated the hypothesis that translations of a document result in document representations that improve the performance in the task of text classification. The approach used here motivated experiments with more and different languages. Is the hypothesis valid for languages that are very dissimilar like English and Japanese for instance? Are there any benefits when the number of more languages used to learn the representation increases?

We also discussed multi-task learning with neural networks. We showed that by jointly learning two sentiment classification tasks for tweets improved the performance of the fine-grained text classification task. Multi-task learning is a very promising area of research as it allows for knowledge transfer between different fields. For the case of sentiment analysis, it would be interesting to see if similar results can be obtained by varying the type of text.For instance, instead of using only tweets one could also use reviews of hotels or products. The hope is that by incorporating more data in the process of training one can arrive at selecting a

better hypothesis that would improve the final performance. The direction of investigating different types of tasks is also interesting: instead of focusing only on sentiment classification tasks, one may want to include other NLP tasks like Part-of-Speech tagging of Named-Entity-Recognition that would add a different type of inductive bias in the process.

The last application we investigated concerned cross-lingual documents retrieval. We used the solvers for the problem of optimal transport with ground distances estimated by the cross-lingual embeddings. An important aspect of this approach is the computational cost. Although the distance estimation between documents is straightforward to parallelize, it is still computationally expensive. Therefore, aceclerating the solution by aggressive sampling techniques so that less problems can be solved is another promising area. Also, in the community of compute vision area, earth movers distance has been shown to perform well when used as a kernel [63, 42] for classification tasks. We believe that building on these findings may lead to similar findings for the Word Movers Distance presented in this chapter.

Bibliography

[1] Judit Acs, Katalin Pajkossy, and Andras Kornai. Building basic vocabulary across 40 languages. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 52–58, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[2] Héctor Martínez Alonso and Barbara Plank. Multitask learning for semantic sequence prediction under varying data conditions. *arXiv:1612.02251*, 2016.

[3] Hesam Amoualian, Marianne Clausel, Eric Gaussier, and Massih-Reza Amini. Streaming-lda: A copula-based approach to modeling topic dependencies in document streams. In *SIGKDD*, 2016.

[4] S. Arlot and M. Lerasle. Why V=5 is enough in V-fold cross-validation. *ArXiv e-prints*, 2012.

[5] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[6] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *UAI*, pages 27–34, 2009.

[7] Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Massih-reza Amini. Reranking approach to classification in large-scale power-law distributed category systems. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, 2014.

[8] Narayanaswamy Balakrishnan and Valery B Nevzorov. *A primer on statistical distributions*. John Wiley & Sons, 2004.

[9] Georgios Balikas and Massih-Reza Amini. Multi-label, multi-class classification using polylingual embeddings. In *Advances in Information Retrieval*

- *38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 723–728. Springer, 2016.

[10] Georgios Balikas and Massih-Reza Amini. Twise at semeval-2016 task 4: Twitter sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 85–91, 2016.

[11] Georgios Balikas, Massih-Reza Amini, and Marianne Clausel. On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 921–924, 2016.

[12] Georgios Balikas, Hesam Amoualian, Marianne Clausel, Éric Gaussier, and Massih-Reza Amini. Modeling topic dependencies in semantically coherent text spans with copulas. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1767–1776, 2016.

[13] Georgios Balikas, Simon Moura, and Massih-Reza Amini. Multitask learning for fine-grained twitter sentiment analysis. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Tokyo, Japan, August 7-11, 2017*, 2017.

[14] Georgios Balikas, Ioannis Partalas, Eric Gaussier, Rohit Babbar, and Massih-Reza Amini. Efficient model selection for regularized classification by exploiting unlabeled data. In *International Symposium on Intelligent Data Analysis*, pages 25–36. Springer, 2015.

[15] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247, 2014.

[16] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

[17] Antonio Bella, Cesar Ferri, José Hernández-Orallo, and Maria Jose Ramirez-Quintana. Quantification via probability estimators. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 737–742. IEEE, 2010.

[18] Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.

[19] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[20] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[21] William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *EMNLP-CoNLL*, pages 546–556, 2012.

[22] Roi Blanco and Christina Lioma. Random walk term weighting for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 829–830. ACM, 2007.

[23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[24] D.M. Blei and J.D. McAuliffe. Supervised Topic Models. In *Advances in Neural Information Processing Systems 20 NIPS*, pages 121–128. Curran Associates, Inc., 2008.

[25] Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 203–208, 1999.

[26] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357, 2016.

[27] Jordan Boyd-Graber and David M Blei. Multilingual topic models for unaligned text. In *UAI*, pages 75–82, 2009.

[28] Jordan Boyd-Graber and Philip Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. *EMNLP*, pages 45–55, 2010.

[29] Jordan L Boyd-Graber and David M Blei. Syntactic topic models. In *NIPS*, 2009.

[30] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[31] Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, pages 288–296, 2009.

[32] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[33] Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan, and Zheng Chen. Contextual text understanding in distributional semantic space. In *CIKM*, pages 133–142, 2015.

[34] François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[35] Yaacov Choueka. Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In *RIAO 88:(Recherche d'Information Assistée par Ordinateur). Conference*, pages 609–623, 1988.

[36] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Approximate fisher kernels of non-iid image models for image categorization. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1084–1098, 2016.

[37] Trevor Cohen and Dominic Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of biomedical informatics*, 42(2):390–405, 2009.

[38] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, 2008.

[39] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[40] Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. Trans-gram, fast cross-lingual word-embeddings. *arXiv preprint arXiv:1601.02502*, 2016.

[41] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001.

[42] Mohammad Reza Daliri. Kernel earth mover's distance for eeg classification. *Clinical EEG and neuroscience*, 44(3):182–187, 2013.

[43] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *ACL*, pages 795–804, 2015.

[44] Wim De Smet and Marie-Francine Moens. Cross-language linking of news stories on the web using interlingual topic modelling. In *Proceedings of the 2nd ACM workshop on Social web search and mining*, pages 57–64. ACM, 2009.

[45] Wim De Smet, Jie Tang, and Marie-Francine Moens. Knowledge transfer across multilingual corpora via latent topics. In *Advances in Knowledge Discovery and Data Mining*. Springer, 2011.

[46] Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *SemEval@ NAACL-HLT*, pages 1124–1128, 2016.

[47] L. Du, W. Buntine, and H. Jin. A Segmented Topic Model Based on the Two-parameter Poisson-Dirichlet Process. *Journal of Machine learning*, 81(1):5–19, 2010.

[48] Lan Du, Wray L Buntine, and Mark Johnson. Topic segmentation with a structured topic model. In *HLT-NAACL*, 2013.

[49] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL (2)*, pages 845–850, 2015.

[50] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. *arXiv preprint arXiv:1606.09403*, 2016.

[51] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In *ACL*, pages 334–343, 2015.

[52] G. Elidan. Copula Network Classifiers (CNCs). In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 346–354, 2012.

[53] G. Elidan. Copulas in Machine Learning. In *Advances in Copulae in mathematical and quantitative finance*, pages 39–60. Springer, 2013.

[54] Andrea Esuli and Fabrizio Sebastiani. Optimizing text quantifiers for multivariate loss functions. Technical report, Technical Report 2013-TR-005, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 2013.

[55] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9, 2008.

[56] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1606–1615, 2015.

[57] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. Association for Computational Linguistics, 2014.

[58] John Rupert Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

[59] George Forman. Counting positives accurately despite inaccurate classification. In *Machine Learning: ECML 2005*, pages 564–575. Springer, 2005.

[60] George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.

[61] Kosuke Fukumasu, Koji Eguchi, and Eric P Xing. Symmetric correspondence topic models for multilingual text analysis. In *Advances in Neural Information Processing Systems*, 2012.

[62] Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. Learning continuous phrase representations for translation modeling. *Proc. of ACL. Association for Computational Linguistics, June*, 2014.

[63] Andrew Gardner, Christian A Duncan, Jinko Kanno, and Rastko R Selmic. Earth mover's distance yields positive definite kernels for certain ground distances. *arXiv preprint arXiv:1510.02833*, 2015.

[64] Matt Gardner, Kejun Huang, Evangelos Papalexakis, Xiao Fu, Partha Talukdar, Christos Faloutsos, Nicholas Sidiropoulos, and Tom Mitchell. Translation invariant word embeddings. EMNLP, 2015.

[65] Anastasia Giachanou and Fabio Crestani. Like it or not: A survey of twitter sentiment analysis methods. *ACM Comput. Surv.*, 49(2):28:1–28:41, 2016.

[66] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[67] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[68] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[69] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[70] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[71] Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *ICML*, pages 748–756, 2015.

[72] Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. In *HLT-NAACL*, pages 1386–1390, 2015.

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

[74] T.L. Griffiths, D.M. Steyvers, M.and Blei, and J.B. Tenenbaum. Integrating Topics and Syntax. In *Proceedings of Neural Information Processing Systems 17 NIPS*, volume 4, pages 537–544, 2004.

[75] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, page 6, 2010.

[76] John M Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. 1971.

[77] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[78] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[79] Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. In *ICLR*, 2013.

[80] Sepp Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. PhD thesis, diploma thesis, institut für informatik, lehrstuhl prof. brauer, technische universität münchen, 1991.

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

[82] M. Hofert. Efficiently Sampling Nested Archimedean Copulas. *Journal of Computational Statistics & Data Analysis*, 55(1):57–70, 2011.

[83] M. Hofert, M. Mächler, et al. Nested Archimedean Copulas Meet R: The nacopula Package. *Journal of Statistical Software*, 39(9):1–20, 2011.

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

[85] Liangjie Hong. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1212.3900*, 2012.

[86] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[87] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[88] Jagadeesh Jagarlamudi and Hal Daumé III. Extracting multilingual topics from unaligned comparable corpora. In *Advances in Information Retrieval*, pages 444–456. Springer, 2010.

[89] Mark Johnson. Pcfgs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157. Association for Computational Linguistics, 2010.

[90] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[91] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[92] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 2016.

[93] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *JAIR*, pages 723–762, 2014.

[94] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[95] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. 2012.

[96] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. *MT summit*, pages 79–86, 2005.

[97] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI'95, 1995.

[98] Kriste Krstovski and David A Smith. Online polylingual topic models for fast document translation detection. In *8th Workshop on Statistical Machine Translation*, pages 252–261, 2013.

[99] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[100] Matt J Kusner, Yu Sun, Nicholas I Kolkin, Kilian Q Weinberger, et al. From word embeddings to document distances. In *ICML*, volume 15, pages 957–966, 2015.

[101] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[102] Jey Han Lau and Timothy Baldwin. The sensitivity of topic coherence evaluation to topic cardinality. In *NAACL*, pages 483–487, 2016.

[103] Jey Han Lau, Timothy Baldwin, and David Newman. On collocations and topic models. *TSLP*, 2013.

[104] Stanislas Lauly, Alex Boulanger, and Hugo Larochelle. Learning multilingual word representations using a bag-of-words autoencoder. *arXiv preprint arXiv:1401.1803*, 2014.

[105] Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

[106] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL (1)*, pages 270–280, 2015.

[107] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[108] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185, 2014.

[109] Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *IJCAI*, pages 3650–3656, 2015.

[110] Bing Liu, Lin Liu, Anna Tsykin, Gregory J Goodall, Jeffrey E Green, Min Zhu, Chang Hee Kim, and Jiuyong Li. Identifying functional mirna–mrna regulatory modules with correspondence latent dirichlet allocation. *Bioinformatics*, 26(24):3105–3111, 2010.

[111] H. Liu, J. Lafferty, and L. Wasserman. The Nonparanormal: Semiparametric Estimation of High Dimensional UndirectedGraphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009.

[112] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[113] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[114] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Deep multi-task learning with shared memory for text classification. In *EMNLP*, pages 118–127, 2016.

[115] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In *IJCAI*, pages 2873–2879, 2016.

[116] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *AAAI*, pages 2418–2424, 2015.

[117] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[118] Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.

[119] A.W. Marshall and I. Olkin. Families of Multivariate Distributions. *Journal of the American Statistical Association*, 83(403):834–841, 1988.

[120] Giovanni Da San Martino, Wei Gao, and Fabrizio Sebastiani. Ordinal text quantification. In *SIGIR*, pages 937–940, 2016.

[121] Peter McCullagh. Generalized linear models. *European Journal of Operational Research*, 16(3):285–292, 1984.

[122] A. M. McEnery and R. Z. Xiao. *Parallel and comparable corpora: What are they up to?* Translating Europe. Multilingual Matters, 2007.

[123] A.J. McNeil. Sampling Nested Archimedean Copulas. *Journal of Statistical Computation and Simulation*, 78(6):567–581, 2008.

[124] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[125] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[126] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[127] David Mimno, Hanna M Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. Polylingual topic models. In *EMNLP*, pages 880–889, 2009.

[128] David M. Mimno, Hanna M. Wallach, Edmund M. Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, pages 262–272, 2011.

[129] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

[130] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

[131] Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics, 2010.

[132] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[133] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. Semeval-2016 task 4: Sentiment analysis in twitter. In *SemEval@NAACL-HLT*, pages 1–18, 2016.

[134] Roger B Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

[135] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *NAACL*, pages 100–108, 2010.

[136] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving topic models with latent feature word representations. *TACL*, 3:299–313, 2015.

[137] Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Mining multilingual topics from wikipedia. In *WWW*, pages 1155–1156, 2009.

[138] Michael A Nielsen. Neural networks and deep learning. *URL: http://neuralnetworksanddeeplearning. com/.(visited: 01.11. 2014)*, 2015.

[139] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[140] Arvid Österlund, David Ödling, and Magnus Sahlgren. Factorization of latent variables in distributional semantic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 227–231, 2015.

[141] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. LSHTC: A benchmark for large-scale text classification. *CoRR*, abs/1503.08581, march 2015.

[142] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[143] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[144] Ofir Pele and Michael Werman. A linear time histogram metric for improved sift matching. In *Computer Vision–ECCV 2008*, pages 495–508. Springer, October 2008.

[145] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, September 2009.

[146] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[147] Jean-François Pessiot, Young-Min Kim, Massih R Amini, and Patrick Gallinari. Improving document clustering in a learned concept space. *Information processing & management*, 46(2):180–192, 2010.

[148] Barbara Plank. Keystroke dynamics as signal for shallow syntactic parsing. In *COLING*, pages 609–619, 2016.

[149] John C Platt, Kristina Toutanova, and Wen-tau Yih. Translingual document representations from discriminative projections. In *EMNLP*, 2010.

[150] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[151] François Rousseau and Michalis Vazirgiannis. Composition of tf normalizations: new insights on scoring functions for ad hoc ir. In *Proceedings of*

the 36th international ACM SIGIR conference on Research and development in information retrieval, pages 917–920. ACM, 2013.

[152] François Rousseau and Michalis Vazirgiannis. Graph-of-word and tw-idf: new approach to ad hoc ir. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 59–68. ACM, 2013.

[153] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[154] Sebastian Ruder. A survey of cross-lingual embedding models. Technical report, http://sebastianruder.com/cross-lingual-embeddings, 2016.

[155] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[156] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[157] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[158] Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 298–307, 2015.

[159] M. Shafiei and E. Milios. Latent Dirichlet Co-clustering. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 542–551, Washington, DC, USA, 2006. IEEE Computer Society.

[160] Tianze Shi, Zhiyuan Liu, Yang Liu, and Maosong Sun. Learning cross-lingual word embeddings via matrix co-factorization. In *ACL (2)*, pages 567–572, 2015.

[161] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[162] M. Sklar. *Fonctions de Répartition à n Dimensions et Leurs Marges*. Université Paris 8, 1959.

[163] Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*, 2015.

[164] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[165] Robert Speer and Joshua Chin. An ensemble method to produce high-quality word embeddings. *CoRR*, abs/1604.01692, 2016.

[166] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4444–4451, 2017.

[167] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[168] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

[169] Yik-Cheung Tam and Tanja Schultz. Bilingual lsa-based translation lexicon adaptation for spoken language translation. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[170] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1555–1565, 2014.

[171] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In

Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 173–180. Association for Computational Linguistics, 2003.

[172] D. Tran, D.M. Blei, and E.M. Airoldi. Copula Variational Inference. In *Proceedings of Neural Information Processing Systems 28 NIPS*, pages 3564–3572, 2015.

[173] P.K. Trivedi and D.M. Zimmer. *Copula Modeling: An Introduction for Practitioners*. Now Publishers Inc, 2007.

[174] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1, 2015.

[175] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[176] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[177] Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora. *Information Retrieval*, 2013.

[178] Ivan Vulić, Wim De Smet, Jie Tang, and Marie-Francine Moens. Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications. *Information Processing & Management*, 51(1):111–147, 2015.

[179] Ivan Vulić and Marie-Francine Moens. Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, 55:953–994, 2016.

[180] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[181] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300. Association for Computational Linguistics, 2009.

[182] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM 2007*, pages 697–702, 2007.

[183] Y Wang. Distributed gibbs sampling of latent topic models: The gritty details. Technical report, 2008.

[184] Y. Wang, E. Agichtein, and M. Benzi. TM-LDA: Efficient Online Modeling of Latent Topic Transitions in Social Media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 123–131, New York, NY, USA, 2012. ACM.

[185] Yu-Chun Wang, Chun-Kai Wu, and Richard Tzong-Han Tsai. Cross-language article linking with different knowledge bases using bilingual topic model and translation features. *Knowledge-Based Systems*, 111:228–236, 2016.

[186] A.G. Wilson and Z. Ghahramani. Copula Processes. In *Advances in Neural Information Processing Systems 23 NIPS*, pages 2460–2468. Curran Associates, Inc., 2010.

[187] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[188] Haiyang Wu, Daxiang Dong, Xiaoguang Hu, Dianhai Yu, Wei He, Hua Wu, Haifeng Wang, and Ting Liu. Improve statistical machine translation with context-sensitive bilingual semantic embedding model. In *EMNLP*, pages 142–146, 2014.

[189] Min Xiao and Yuhong Guo. Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*, pages 119–129, 2014.

[190] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *HLT-NAACL*, pages 1006–1011, 2015.

[191] Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. *arXiv preprint arXiv:1606.04038*, 2016.

[192] Duo Zhang, Qiaozhu Mei, and ChengXiang Zhai. Cross-lingual latent topic extraction. In *ACL*, pages 1128–1137, 2010.

[193] Heng Zhang and Guoqiang Zhong. Improving short text classification by learning vector representations of both words and hidden topics. *Knowl.-Based Syst.*, 102:76–86, 2016.

[194] Tao Zhang, Kang Liu, Jun Zhao, et al. Cross lingual entity linking with bilingual topic model. In *IJCAI*, 2013.

[195] Xiang Zhang and Yann LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.

[196] Bing Zhao and Eric P Xing. Bitam: Bilingual topic admixture models for word alignment. In *COLING*, pages 969–976, 2006.

[197] Xiaojin Zhu, David Blei, and John Lafferty. Taglda: Bringing document structure knowledge into topic models. Technical report, Technical Report TR-1553, University of Wisconsin, 2006.

[198] Judit Ács. Pivot-based multilingual dictionary building using wiktionary. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

Appendix A

Efficient Model Selection for Regularized Classification by Exploiting Unlabeled Data

In the main part of the manuscript we proposed extensions of topics models and we explored applications of word embeddings for various challenging tasks. In this appendix we present a contribution that concerns model selection.[1] Differently from the presented models, that learn word representations by modeling how words co-occur within documents, we move to a higher level and explore how categories occur in a collection of documents. By observing the distribution output by a classifier such as Logistic Regression, we propose algorithms for efficient model selection.

Model selection is an essential step in the pipeline of data analysis tasks. Having decided on the algorithm to be used, one should proceed to parameter selection that is the process of selecting a value for the model's hyper-parameter(s) expected to obtain the optimal performance on unseen examples. For instance, when using Support Vector Machines (SVM) or Logistic Regression (LR) in a classification task, one has to tune the regularization parameter λ which controls the complexity of the model.

The fundamental idea of parameter estimation methods is to validate the model's performance in fractions of the training data. In several learning scenarios however, except few labeled data, a larger set of unlabeled data may be available (for example in text classification) as the cost of assigning labels is high. This is the case for example of the transductive learning framework [32], where the data to

[1] Chronologically, this was the first contribution in the framework of the author's thesis.

be classified are available beforehand and can be leveraged during the training or inference procedures.

The situation we are investigating in this paper is when unlabeled data are available during the step of parameter selection in a classification problem. The challenge is to come up with a method that is able to leverage the information in the unlabeled data, instead of ignoring them as traditional model selection strategies such as k-fold cross validation (k-CV) do. To tackle this problem, we incorporate quantification techniques in order to infer the distribution of the examples on unlabeled data, which in turn is used to calculate upper bounds (Section 3) on the performance of a model that motivate an efficient model selection scheme (Section 4).

We place ourselves in the supervised learning paradigm where the i.i.d. assumption holds. Note that unlike semi-supervised and transductive learning that make use of the unlabeled data in the training process to improve the performance, we use the unlabeled data for *hyper-parameter selection* and, hence, the obtained performance in the test set depends on the amount of the available labeled data. Our method, which is an alternative to k-CV, motivates the selection of the optimal value for the model's hyper-parameter(s) from a finite set that in turn results in the optimal performance (again from a finite set of possible performances). In this work, we propose a hyper-parameter selection method that (i) benefits from unlabeled data, (ii) performs on par with k-CV but it is k times faster and (iii) has the same complexity as hold-out estimation but performs better due to the use of unlabeled data. We demonstrate the efficiency and the effectiveness of the proposed method in Section 5 where we present multi-class text classification results on several datasets with a large number of classes.

A.1 Related Work

Several approaches have been proposed for selecting the hyper-parameters of learning algorithms. The goal is always to select the hypothesis that minimizes the generalization error, which is approximated by the estimated error [132]. A popular method to calculate the estimated error is the hold-out procedure that splits the data in a training and a validation set; the estimated error is calculated on the latter.

The k-CV technique repeats k times the hold-out procedure: in each round the available training data are partitioned into two complementary subsets, one for

training and one for validation. To reduce variability, multiple rounds of cross-validation are performed using different partitions and the validation results are averaged over the rounds. At the end, an hypothesis is selected e.g. by retraining the classifier on all data using the best values found for the hyper-parameters, or by averaging the hypotheses [18]. A variant of this method is proposed by Blum et al. [25] with a progressive cross-validation procedure that begins by splitting the data in training and test. At each step, it tests an example which in the next round is used for training, resulting in as many hypotheses as the available test examples. To label an example, a hypothesis is randomly selected. This method has the advantage of using more examples for training than the hold-out and was shown to select a better hypothesis. In addition, the study in [97] reviews accuracy estimation and model selection methods based on cross-validation and bootstrap. The former is shown to be better than the latter in different datasets, especially in terms of accuracy estimation (for which a stratified approach may be preferred).

The hold-out estimation and the k-CV when k is small are known to have large variance, a problem that can be partially compensated in k-CV by selecting high values for k (like 5 or 10) [5, 4]. However, k-CV and its variants are computationally expensive and may be intractable in practice if one wants to search for the appropriate values in large-scale scenarios.

We propose here a different method that can select an appropriate model on unlabeled datasets. The advantages compared to the above-mentioned methods concern its efficiency and its ability to be applied when few labeled examples are available. It dispenses with the use of validation sets which can be cumbersome to produce in unbalanced or small datasets. It is, however, intended for model selection only, whereas cross-validation and hold-out estimation can be used for performance evaluation as well.

A.2 Accuracy and Macro-F1 Quantification Bounds

In this section, we propose an upper bound on several performance measures (accuracy and macro-F1) of a given classifier C on a dataset S which doesn't need to be labeled. We then use this bound, which is based on the class distribution induced by C on S, to perform model selection.

We considered mono-label multi-class classification problems, where observations $\mathbf{x}$ lie in an input space $\mathcal{X} \subseteq \mathbb{R}^d$. Each observation $\mathbf{x}$ is associated with a label $y \in \mathcal{Y}$, where $|\mathcal{Y}| > 2$. We suppose that examples consist of pairs of $(\mathbf{x}, y)$ identically

and independently distributed (i.i.d) according to a fixed, but unknown probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{D}_{\mathcal{X}}$ will denote the marginal probability for $\mathbf{x}$ in $\mathcal{X}$). In the context of text classification, $\mathbf{x}^{(i)} \in \mathcal{X}$ denotes the vector representation of document i and its label $y^{(i)} \in \mathcal{Y}$ represents the category associated with $\mathbf{x}^{(i)}$. We further assume to have access to a training set $S_{train} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ also generated i.i.d with respect to $\mathcal{D}$.

Quantification. As explained below, our analysis makes use of $M_y^{C(S)}$, the number of documents in the unlabeled set S assigned by classifier C to class y. Many classifiers do not directly assign a category to documents, but rather produce scores (probabilistic or not) for each category, from which a categorization decision can be made. The task of determining the number of instances of each target category in a set S is called *quantification* and was first proposed by Forman et al. [59, 60]. Contrary to classification that identifies in which target categories an observation belongs, quantification is solely concerned with the estimation of the number of observations belonging to a target category (the positive examples). Note that a good quantifier is not necessarily a good classifier, and vice versa. For example, in a binary problem with 40 observations, a learner that outputs 20 False Positives and 20 False Negatives is a perfect quantifier but a really bad classifier.

Given a set of instances in S, quantifiers output, for each target category y of S, a number denoting the prediction of the relative frequency of category y in S. Quantification methods using general purpose learners are usually split ([54]) in *aggregative* and *non aggregative* methods based on whether the quantification step requires the classification of the individual instances as a basic step or not. Quantification has been mainly used to estimate distribution drifts. We make a different use of it here, in the context of model selection, and rely on two popular quantification methods, namely: a) *Classify and Count (CC)* and b) *Probabilistic Classify and Count (PCC)* [54]. In *CC*, given a classifier C trained on a set S_{train}, the relative frequency of a class y in a set S, denoted by $\overline{p}_y^{C(S)}$, is obtained by counting the instances of S that classifier C assigns the target category y, that is $\overline{p}_y^{C(S)} = \frac{M_y^{C(S)}}{|S|}$, where $|S|$ denotes the size of S. *PCC* extends *CC* using the posterior probabilities of an instance belonging to a category, leading to $\overline{p}_y^{C(S)} = \frac{1}{|S|} \sum_{\mathbf{x} \in S} p(y|\mathbf{x})$, where $p(y|\mathbf{x})$ is the posterior probability that an instance $\mathbf{x}$ of S belongs to y. We do not consider the adjusted version of those two approaches proposed in [17] because they require the expensive k-fold cross-validation in the training set which is undesirable in large scale settings. Lastly, having a trained classifier, the computational complexity of quantification reduces to the prediction cost of a trained classifier.

Quantification-based Bounds. We now present our main result which consists of quantification-based upper bounds on the accuracy (denoted $A^{C(S)}$), the macro-precision (denoted $MaP^{C(S)}$), the macro-recall (denoted $MaR^{C(S)}$) and the macro-F1 (denoted $MaF^{C(S)}$) of a classifier C on a dataset S which does not need to be labeled.

Theorem A.2.1. *Let $S = \{(\boldsymbol{x}^{(j)})\}_{j=1}^{M}$ be a set generated i.i.d. with respect to $\mathcal{D}_{\mathcal{X}}$, p_y the true prior probability for category $y \in \mathcal{Y}$ and $\frac{N_y}{N} \triangleq \hat{p}_y$ its empirical estimate obtained on S_{train}. We consider here a classifier C trained on S_{train} and we assume that the quantification method used is accurate in the sense that:*

$$\exists \epsilon, \epsilon \ll \min\{p_y, \hat{p}_y, \overline{p}_y^{C(S)}\}, \forall y \in \mathcal{Y} : |\overline{p}_y^{C(S)} - \frac{M_y^{C(S)}}{|S|}| \quad \leq \epsilon$$

Let $B_A^{C(S)}$, $B_{MaP}^{C(S)}(\epsilon)$ and $B_{MaR}^{C(S)}(\epsilon)$ be defined as:

$$\frac{\sum_{y \in \mathcal{Y}} \min\{\hat{p}_y \times |S|, \overline{p}_y^{C(S)} \times |S|\}}{|S|} \triangleq B_A^{C(S)}$$

$$\frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{\min\{\hat{p}_y \times |S|, \overline{p}_y^{C(S)} \times |S|\} + |S|\epsilon}{\overline{p}_y^{C(S)} \times |S| + |S|\epsilon} \triangleq B_{MaP}^{C(S)}(\epsilon)$$

$$\frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{\min\{\hat{p}_y \times |S|, \overline{p}_y^{C(S)} \times |S|\} + |S|\epsilon}{\hat{p}_y^{C(S)} \times |S| + |S|\epsilon} \triangleq B_{MaR}^{C(S)}(\epsilon)$$

Then for any $\delta \in]0, 1]$, with probability at least $(1 - \delta)$:

$$A^{C(S)} \leq B_A^{C(S)} + |\mathcal{Y}|(\sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}} + \epsilon) \tag{A.1}$$

$$MaP^{C(S)} \leq B_{MaP}^{C(S)}(\epsilon) + \sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}},\ MaR^{C(S)} \leq B_{MaR}^{C(S)}(\epsilon) + \sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}} \tag{A.2}$$

$$MaF^{C(S)} \leq \frac{2B_{MaP}^{C(S)}(\epsilon)B_{MaR}^{C(S)}(\epsilon)}{B_{MaP}^{C(S)}(\epsilon) + B_{MaR}^{C(S)}(\epsilon)} + \sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}} \tag{A.3}$$

Proof. (SKETCH) We first consider the case where $S \neq S_{train}$. Using Hoeffding's inequality for random variables bounded in the interval $[0, 1]$, we have the standard result that, for any $\delta \in]0, 1]$, with probability at least $(1 - \delta)$:

$$\forall y \in \mathcal{Y}, p_y \leq \hat{p}_y + \sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}}$$

The $\log|\mathcal{Y}|$ factor is a result of the fact that the bound should hold simultaneously for all categories. This implies, using the quantification assumption, that, for any $\delta \in]0,1]$, with probability at least $(1-\delta)$, $\forall y \in \mathcal{Y}$:

$$|\min\{p_y \times |S|, M_y^{C(S)}\} - \min\{\hat{p}_y \times |S|, \overline{p}_y^{C(S)} \times |S|\}| < |S|(\sqrt{\frac{\log|\mathcal{Y}| + \log\frac{1}{\delta}}{2N}} + \epsilon) \quad \text{(A.4)}$$

$\min\{p_y \times |S|, M_y^{C(S)}\}$ corresponds to an upper bound on the number of documents of S correctly classified by C in class y. Hence, the accuracy of C on S is upper bounded by:

$$\frac{\sum_{y\in\mathcal{Y}} \min\{p_y \times |S|, M_y^{C(S)}\}}{|S|}$$

which leads, using Inequality A.4, to Inequality A.1. The other bounds can be derived in the same way. □ □

The above theorem is inspired by a previous result we have developed in the context of multi-class classification [7]. We have generalized and extended it here through the consideration of macro measures and quantification. Even though this extension renders the developments more complex, it is crucial for model selection using unlabeled datasets.

When the *Classify and Count* (*CC*) quantification method is used, then, by definition, $\overline{p}_y^{C(S)} = \frac{M_y^{C(S)}}{|S|}$, and ϵ can be set to 0. This leads to stricter bounds for all the measures. Furthermore, the condition $\epsilon \ll \min\{p_y, \hat{p}_y, \overline{p}_y^{C(S)}\}$ in the quantification assumption implies that the term $|S|\epsilon$ is negligible compared to $|S| \times \hat{p}_y$ or $|S| \times \overline{p}_y^{C(S)}$, so that $B_{MaP}^{C(S)}(\epsilon)$ and $B_{MaR}^{C(S)}(\epsilon)$ are close to $B_{MaP}^{C(S)}(0)$ and $B_{MaR}^{C(S)}(0)$. Lastly, it can be noted that the quality of the bound is better for the macro measures than for the accuracy as the multiplying $|\mathcal{Y}|$ factor is dropped.

Theorem A.2.1 states that the accuracy, macro-precision, macro-recall and macro-F1 of a classifier can be upper-bounded by quantities that are related to the behavior of the classifier on an unlabeled dataset, and that the quality of the bound depends on the number of classes, the size of the training set, the quality of the quantification method and the precision desired. These bounds represent necessary conditions for a classifier C to have high accuracy/macro-F1[2]. They can nev-

[2]They do not provide a sufficient condition since it is possible, in an adversarial setup, to achieve an upper bound of 1 on the accuracy by simply assigning instances to categories in the same proportion as in the training set.

Algorithm 4: Model selection using the proposed bounds

Require: Training data $S_{train} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, $S = \{(\mathbf{x}^{(j)})\}_{j=1}^{M}$, and learning algorithm (SVM, Logistic Regression, ...)

for each value of λ (typically $\lambda \in \{10^{-4}, 10^{-3}, \ldots, 10^{2}, 10^{3}\}$) **do**

Train a classifier C_λ using S_{train}

Perform quantification of C_λ on S using method M_q (typically *CC* or *PCC*)

If $M_q = CC$, set $\epsilon = 0$

If $M_q \neq CC$, set $\epsilon = \max_{y \in \mathcal{Y}} \min\{\hat{p}_y, \overline{p}_y^{C(S)}\} - |\overline{p}_y^{C(S)} - \frac{M_y^{C(S)}}{|S|}|$

If $\epsilon < 0$, go back to step 4 with $M_q = CC$

Compute the accuracy bound using Inequality A.1

Compute the macro-F1 bound ($\frac{2B_{MaP}^{C_\lambda(S)}(\epsilon) B_{MaR}^{C_\lambda(S)}(\epsilon)}{B_{MaP}^{C_\lambda(S)}(\epsilon) + B_{MaR}^{C_\lambda(S)}(\epsilon)}$) using Inequality A.3

end

Select C_λ with the highest accuracy/macro-F1 bound

ertheless be exploited, within a given family of classifiers obtained through *e.g.* different regularization parameters, to select good classifiers.

Model Selection Using Quantification Bounds. We consider here a standard regularization setting in which one aims at minimizing a combination of the empirical error and the model complexity using the following template of the objective function:

$$\hat{w} = \arg\min R_{emp}(w) + \lambda Reg(w)$$

where $Reg(w)$ is the regularization term to avoid overfitting and $R_{emp}(.)$ represents the empirical error.

The parameter λ controls the trade-off between the empirical error and the regularization term. As mentioned before, λ is typically estimated through hold-out estimation or k-fold cross-validation. We propose here to estimate it on the basis of the upper bounds presented in Theorem A.2.1, as described in Algorithm 4. As one can note, for each value of λ, a classifier is trained and quantified on the unlabeled set S. If the quantification assumption of Theorem A.2.1 is not valid, then one falls back on the *Classification and Count* method for quantification. The bounds, as computed by Inequalities A.1 and A.3 are used to select the "best" classifier. Tuning the hyper-parameter is, therefore, reduced to the problem of finding a classifier which yields the highest value of the bounds on a given set. In contrast with other selection methods, the set used to select the classifier can be an unlabeled set from the same distribution (unlabeled data is usually readily available, contrary to labeled data) or the test set in a transductive-like scenario.

In terms of complexity, the quantification cost is reduced to the prediction for the already trained classifier, which is linear in the cardinality of the set S on which quantification is performed. The computational cost for Algorithm 4 is thus the same as 1-fold cross-validation. Additionally, as only one hypothesis is generated for each parameter value by training to the whole set of labeled data one has just to select the hypothesis with the highest bound without the need of retraining the model in contrast to hold-out or k-fold cross-validation. More precisely, the complexity of our approach for m values of λ is O([Tr(N) + Pr(M)]×m), which is k times lower than the complexity of k-CV with re-training the learner for the selected λ value, given by O([Tr(($\frac{k-1}{k}$)×N)+Pr($\frac{1}{k}$×N)]×k×m+T(N)), where Tr(N), Pr(N) are the training and predicting costs for N examples.

A.3 Experimental Framework

To empirically evaluate the model selection method presented above we use the publicly available datasets of the LSHTC 2011 (*Large Scale Hierarchical Text Classification*) challenge [141]. Specifically, we make use of the Dmoz and Wikipedia datasets containing 27,875 and 36,504 categories respectively. The datasets are provided in a pre-processed format using stop-word removal and stemming while we transformed the term-frequency vectors to the tf*idf representation. For each of the datasets we randomly draw several datasets with increasing number of classes.

Table A.1 presents the important statistics of the different datasets. As one can note, the number of categories in our datasets ranges from 250 to 2,500, and the number of features from 26,000 to 212,000. An interesting property of the instances of those datasets is the fit to the power law distribution. As a result, there are several under-represented classes having a few labelled examples. Thus, model selection approaches using only a fraction of the labeled instances, such as hold out, may lead to sub-optimal decisions.

The classification problems defined from our datasets are multi-class, and we adopt a standard one-vs-rest approach to address them (the large datasets considered prevents one from using more complex multi-class approaches). The Dmoz dataset is single-labeled, *i.e.* each training/test instance is associated to a single target category. On the other hand, the Wikipedia dataset is multi-labeled with the average labels per instance in the training set being 1.85. We transformed the

multi-label problem to single label, both in the training and the test phase, by replicating the multi-labeled instances according to the number of their labels.

In order to empirically measure the effectiveness of model selection, we compare the following three methods: (i) ***k*-CV**, using $k = 5$ folds, (ii) **hold-out** estimation with a split of 70% and 30% for the training and the validation sets, and (iii) our method using as quantification set i) an unlabeled set denoted "quantification set" in Table A.1, and ii) the test set which may be available during training in a transductive alike scenario. The corresponding methods are called $\textbf{Bound}_{\text{UN}}$ and $\textbf{Bound}_{\text{Test}}$ respectively.

Dataset	#Training inst.	#Quantification inst.	#Test inst.	#Features	# Parameters
dmoz_{250}	1,542	2,401	1,023	55,610	13,902,500
dmoz_{500}	2,137	3,042	1,356	77,274	38,637,000
dmoz_{1000}	6,806	10,785	4,510	138,879	138,879,000
dmoz_{1500}	9,039	14,002	5,958	170,828	256,242,000
dmoz_{2500}	12,832	19,188	8,342	212,073	530,182,500
wiki_{250}	1,917	3,095	1,003	26,699	6,674,750
wiki_{500}	4,912	8,190	2,391	46,556	23,278,000
wiki_{1000}	7,887	12,790	4,067	60,788	60,788,000
wiki_{1500}	12,156	19,776	6,160	79,973	110,959,500
wiki_{2500}	22,642	37,398	11,171	109,694	274,235,000

Table A.1: The properties of the datasets we used. The dataset name denotes the collection we sampled it from; its subscript denotes the number of categories.

Evaluation of the quantification methods. We first discuss the performance of the quantification methods presented above (*CC* and *PCC*), prior to comparing the results obtained by the different model selection methods (k-fold cross-validation, hold-out estimation, Bound_{UN} and $\text{Bound}_{\text{Test}}$). Recall that Theorem A.2.1 is based on the assumption that the quantity $\text{Max}_\epsilon = \max_{y\in\mathcal{Y}} |\overline{p}_y^{C(S)} - \frac{M_y^{C(S)}}{|S|}|$ is small. As mentioned above, this quantity is null for the quantification method *CC*, which thus agrees with our theoretical developments. The other quantification method considered, *PCC*, is based on the probabilities that an instance belongs to a class. When using LR, those probabilities are directly produced by the model. For SVMs, however, one needs to transform the confidence scores into probabilities, which can be done in several ways, as using a logistic function, a multivariate logistic regression function or neural networks based on logistic activation functions and without hidden layers (the latter two settings can be seen as generalizations of Platt's scaling for the multi-class problem). We obtained the best results with a

simple logistic function defined as $\frac{1}{1+e^{-\sigma t}}$, varying σ from 1 to 10. Table A.2 displays the values of Max_ϵ obtained for *PCC* for each of the dataset and for each classifier (the default hyper-parameter values of the classifiers are used), using the value of σ leading to the lowest value of Max_ϵ. As one can note, although the values obtained are small in most cases (except for Dmoz_{1000} and Dmoz_{1500}), there are not negligible compared to the class prior probabilities, which are in the range of 1 divided by the number of classes. Thus, the quantification method *PCC* does not fully agree with our theoretical development. It turns out that it also performs worse than *CC* in practice. We thus rely on this latter method for the rest of our experiments.

	dmz_{250}	dmz_{500}	dmz_{1000}	dmz_{1500}	dmz_{2500}	wiki_{250}	wiki_{500}	wiki_{1000}	wiki_{1500}	wiki_{2500}
SVM	.0728	.0967	.1067	.1125	.0345	.0287	.0754	.0310	.0425	.0365
LR	.0942	.0674	.0889	.1111	.0530	.0219	.0517	.0481	.0310	.0294

Table A.2: Evaluation of the assumption of Theorem A.2.1 concerning the quantification step. For each dataset, we present Max_ϵ for the *PCC* quantification method.

Model Selection Evaluation. We evaluate model selection methods for two families of classifiers: (i) SVMs, and (ii) LR which are among the best performing models in text classification. We explore for both classifiers the value for the regularization parameter $\lambda \in \{10^{-3}, 10^{-2}, \ldots, 10^{4}\}$. We used the implementations in Python's scikit-learn [142] that are wrappers of the LibLinear package [55].

We report the scores obtained in Accuracy and Macro-F (MaF) measure when a classifier is applied on the test set. In particular, for each dataset of Table A.1 the model selection methods are used only for selecting the regularization parameter λ when optimizing for the repsective measure. After the selection of λ, the classifier is retrained on the entire training set, and we report its performance in the test set. This last step of retraining is not required for our method since the classifier is trained in the overall labeled set from the beginning. Also, as hold-out estimation may be sensitive to the initial split, we perform 10 different random splits training/validation and report the mean and the standard deviation of the scores obtained for both evaluation measures.

Figure A.1 illustrates the model selection decisions for the different methods using an SVM on the Wikipedia dataset with 1,500 classes for the MaF measure. The curve MaF corresponds to the actual MaF on the test set. Although each parameter estimation method selects the value for λ that seems to maximize the performance, the goal in this example, ultimately, is to select the value that maximizes the performance of MaF. For instance, *hold-out*, by selecting $\lambda = 10^{-1}$, fails

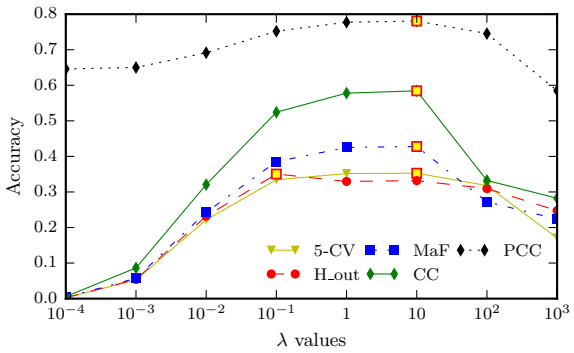


Figure A.1: Model selection process for SVM on the wiki_{1500} for MaF. The squares denote the best performance for each method.

to select the optimal λ value, while all other the methods succeed. Here, the 5-CV approach requires 1310 sec., whereas the bound approach only requires 302 sec. (the computations are performed on a standard desktop machine, using parallelized implementations on 4-cores). The bound approach is thus 4.33 times faster, a result consistent over all experiments and in agreement with the complexity of each approach (Section 3). Lastly, we notice that the curve for Bound_{UN} with the quantification method *CC* follows the MaF curve more strictly than the curve with the quantification method *PCC*.

Table A.3 presents the evaluation of the three model selection methods using as classifiers SVM and LR respectively. As one can note, the performance of the method proposed here is equivalent to the one of cross-validation, for all datasets, and for both classifiers and performance measures (accuracy and MaF). The performance of SVM is furthermore higher than the one of LR on all datasets, and for both evaluation measures, the difference being more important for the MaF. The performance of cross-validation however comes with the cost of extra processing time, as our method achieves a k speed-up compared to cross-validation. If both methods can easily be parallelized (at least on the basis of the number of values of the hyper-parameter to be tested), k-fold cross validation requires k times more computing resources than our method.

Unlike cross-validation, hold-out estimation fails to provide a good model in many instances. This is particularly true for SVMs and the MaF measure, for which the model provided by hold-out estimation lies way behind the ones provided by Bound_{UN} and $\text{Bound}_{\text{Test}}$ on several collections as Dmoz_{1500} and Dmoz_{2500}. The difference is less important for LR, but the final results in that case are not as good as in the SVM case.

		Bound_{Un}		$\text{Bound}_{\text{Test}}$		Hold-out		5-CV	
	Dataset	Acc	MaF	Acc	MaF	Acc	MaF	Acc	MaF
SVM	wiki_{250}	**.7747**	.5889	**.7747**	**.5927**	$.7663_{\pm.0158}$	$.5746_{\pm.0183}$	**.7747**	**.5927**
	wiki_{500}	.7445	**.5257**	**.7449**	.5254	$.7440_{\pm.0006}$	$.5228_{\pm.0031}$	.7445	.5254
	wiki_{1000}	**.7000**	**.4737**	.6993	.4732	$.6996_{\pm.0009}$	$.4584_{\pm.0274}$	**.7000**	**.4737**
	wiki_{1500}	**.6360**	.4278	.6354	**.4283**	$.6343_{\pm.0049}$	$.4230_{\pm.0126}$	**.6360**	.4278
	wiki_{2500}	.5808	**.3763**	.5811	.3762	$.5822_{\pm.0023}$	$.3759_{\pm.0004}$	**.5832**	**.3763**
	dmoz_{250}	.8260	.6242	**.8270**	**.6243**	$.8260_{\pm.0000}$	$.6242_{\pm.0000}$	.8260	.6242
	dmoz_{500}	**.7227**	**.5584**	**.7227**	**.5584**	$.7221_{\pm.0005}$	$.5558_{\pm.0022}$	.7220	.5562
	dmoz_{1000}	**.7302**	.4883	**.7302**	**.4892**	$.7301_{\pm.0001}$	$.4835_{\pm.0155}$	.7299	.4883
	dmoz_{1500}	**.7132**	**.4715**	**.7132**	**.4715**	$.6958_{\pm.0457}$	$.4065_{\pm.0998}$	**.7132**	**.4715**
	dmoz_{2500}	**.6352**	.4301	.6350	**.4306**	$.6350_{\pm.0001}$	$.3949_{\pm.0686}$	**.6352**	.4301
Logistic Regression	wiki_{250}	**.7527**	**.5423**	**.7527**	**.5423**	$.7464_{\pm.0078}$	$.5335_{\pm.0134}$	**.7527**	**.5423**
	wiki_{500}	**.7302**	**.4709**	**.7302**	**.4709**	$.7266_{\pm.0056}$	$.4633_{\pm.0116}$	**.7302**	**.4709**
	wiki_{1000}	**.6836**	**.4354**	**.6836**	**.4354**	$\mathbf{.6836}_{\pm.0000}$	$\mathbf{.4354}_{\pm.0000}$	**.6836**	**.4354**
	wiki_{1500}	**.6166**	**.3801**	**.6166**	**.3801**	$\mathbf{.6166}_{\pm.0000}$	$\mathbf{.3801}_{\pm.0000}$	**.6166**	**.3801**
	wiki_{2500}	**.5802**	**.3506**	**.5802**	**.3506**	$\mathbf{.5802}_{\pm.0000}$	$\mathbf{.3506}_{\pm.0000}$	**.5802**	**.3506**
	dmoz_{250}	**.7742**	**.4724**	**.7742**	**.4724**	$.7718_{\pm.0047}$	$.4692_{\pm.0096}$	**.7742**	**.4724**
	dmoz_{500}	**.6608**	**.4513**	**.6608**	**.4513**	$.6586_{\pm.0064}$	$.4488_{\pm.0076}$	**.6608**	**.4513**
	dmoz_{1000}	**.6845**	**.3681**	**.6845**	**.3681**	$\mathbf{.6845}_{\pm.0000}$	$\mathbf{.3681}_{\pm.0000}$	**.6845**	**.3681**
	dmoz_{1500}	**.6678**	**.3616**	**.6678**	**.3616**	$\mathbf{.6678}_{\pm.0000}$	$\mathbf{.3616}_{\pm.0000}$	**.6678**	**.3616**
	dmoz_{2500}	**.5959**	**.3351**	**.5959**	**.3351**	$\mathbf{.5959}_{\pm.0000}$	$\mathbf{.3351}_{\pm.0000}$	**.5959**	**.3351**

Table A.3: The performance of the model selection methods for SVM and Logistic Regression on the test set. For held out, we report the mean and in parenthesis the standard deviation of 10 rounds of the method.

A.4 Summary

We presented in this work a new method for model selection that is able to exploit unlabeled data (this is in contrast with current model selection methods). To do so, we have introduced quantification-based bounds for accuracy and macro performance measures. We have then shown how to apply this bound in practice, in the case where unlabeled data is available in conjunction with labeled data, and in a transductive-like setting where the instances to be classified are known in advance. The experimental results, obtained on 10 datasets with different number of classes ranging from 250 to 2,500, show that the method proposed here is equivalent, in terms of the quality of the model selected, to k-fold cross-validation, while being k times faster. It furthermore consistently outperforms hold-out estimation for SVM classification, for both accuracy and macro-F1, the difference being more important for macro-F1. Furthermore, and contrary to hold-out estimation, our

method needs neither a validation/train splitting procedure nor a retraining procedure.

In our future work we plan to investigate the application of a generalized version of the proposed model selection approach in cases where more than one hyper-parameters have to be tuned. In this framework, we also plan to research the extension of the theoretical and experimental findings to multi-label classification problems i.e., multi-class classification problems where each instance can be given more than one categories at once.