EFFICIENT INFERENCE, SEARCH AND EVALUATION FOR LATENT VARIABLE MODELS OF TEXT WITH APPLICATIONS TO INFORMATION RETRIEVAL AND MACHINE TRANSLATION

A Dissertation Presented

by

KRISTE KRSTOVSKI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2016

College of Information and Computer Sciences

EFFICIENT INFERENCE, SEARCH AND EVALUATION FOR LATENT VARIABLE MODELS OF TEXT WITH APPLICATIONS TO INFORMATION RETRIEVAL AND MACHINE TRANSLATION

A Dissertation Presented

by

KRISTE KRSTOVSKI

Approved as to style and content by:

David A. Smith, Chair

James Allan, Member

W. Bruce Croft, Member

Bruce Desmarais, Member

Michael J. Kurtz, Member

James Allan, Chair
College of Information and Computer Sciences

To my mother and father, Blagorodna Mirčevska and Mirko Krstovski

ACKNOWLEDGMENTS

First and most of all I would like to thank my advisor David Smith for his constant support, patience and guidance throughout my doctoral studies. David, thank you for being an invaluable source of wisdom and ideas and more importantly for being a role model of a researcher and a teacher.

I am thankful to my committee members W. Bruce Croft, James Allan, Michael J. Kurtz and Bruce Desmarais for their helpful and insightful comments and opinions that made this dissertation better. I'm grateful to Bruce and James for the valuable discussions that we had and for making time to talk with me even when I was working as a predoctoral fellow at the Center for Astrophysics (CfA).

I would like to mention those who have made the journey through my doctoral studies an exciting and enjoyable experience. I would like to start by thanking my lab mates at the Center for Intelligent Information Retrieval (CIIR) for their support whether that may be the time spent in listening to my ideas to sharing their opinions. Many thanks for my time spent at CIIR goes to: Elif Aktogla, Anton Bakalov, Michael Bendersky, Ethem Can, Marc-Allen Cartright, Jeffrey Dalton, Van Dang, Laura Dietz, Henry Feild, Shiri Dori-Hacohen, Weize Kong, Chia Jung Lee, David Mimno, Jae Hyun Park, Sebastian Riedel, Xiaobing Xue, I. Zeki Yalniz and Xing Yi.

Special thanks goes to the CIIR support staff, especially to Jean Joyce, Kate Moruzzi – for perfectly orchestrating all the CIIR logistics, and Dan Parker for making sure that all software related issues are addressed promptly. As a graduate program manager Leeanne Leclerc deserves special thanks for being an invaluable resource for all the doctoral program requirements and for always making sure that students don't miss any deadlines.

I spent half of my PhD studies as a predoctoral fellow at the Harvard-Smithsonian CfA. While being part of the Astrophysics Data System (ADS) effort I had the great opportunity to be advised by Michael J. Kurtz from whom I've learned a great deal of computer science perspectives. I own a great gratitude to Alberto Accomazzi for always finding time to talk to me, share his opinions, and explain to me in details the various aspects of ADS. I would like to specially thank Roman Chyla and Edwin Henneken for helping me with my various experiments and for their willingness to steer away from their regular daily schedules in order to do that.

Lastly and most importantly I would like to thank my parents, Blagorodna and Mirko for their continuous encouragement, teaching me how to persevere through rough periods and for always being there for me. I can't imagine being able to complete my doctoral studies without their unconditional support.

This work was supported in part by the Harvard-Smithsonian CfA predoctoral fellowship, in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

ABSTRACT

EFFICIENT INFERENCE, SEARCH AND EVALUATION FOR LATENT VARIABLE MODELS OF TEXT WITH APPLICATIONS TO INFORMATION RETRIEVAL AND MACHINE TRANSLATION

MAY 2016

KRISTE KRSTOVSKI

B.Sc., UNIVERSITY OF NEW HAMPSHIRE

M.Sc., UNIVERSITY OF NEW HAMPSHIRE

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor David A. Smith

Latent variable models of text, such as topic models, have been explored in many areas of natural language processing, information retrieval and machine translation to aid tasks such as exploratory data analysis, automated topic clustering and finding similar documents in mono- and multilingual collections. Many additional applications of these models, however, could be enabled by more efficient techniques for processing large datasets.

In this thesis, we introduce novel methods that offer efficient inference, search and evaluation for latent variable models of text. We present efficient, online inference for representing documents in several languages in a common topic space and fast approximations for finding near neighbors in the probability simplex representation of mono- and multilingual document collections. Empirical evaluations show that these methods are as accurate as–and significantly faster than–Gibbs sampling and brute-force all-pairs search respectively. In addition, we present a new extrinsic evaluation metric that achieves very high correlation with common performance metrics while being more efficient to compute. We showcase the efficacy and efficiency of our new approaches on the problems of modeling and finding similar documents in a retrieval system for scientific papers, detecting document translation pairs, and extracting parallel sentences from large comparable corpora. This last task, in turn, allows us to efficiently train a translation model from comparable corpora that outperforms a model trained on parallel data.

Lastly, we improve the latent variable model representation of large documents in mono- and multilingual collections by introducing online inference for topic models with hierarchical Dirichlet prior structure over textual regions such as document sections. Modeling variations across textual regions using online inference offers a more effective and efficient document representation, beyond a bag of words, which is usually a handicap for the performance of these models on large documents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Many applications in natural language processing (NLP), information retrieval (IR) and machine translation (MT) use latent variable models of text to model documents in mono- and multilingual collections. Topic models, which are some of the most widely used such models, have been extensively studied and have had their potential showcased on numerous tasks including exploratory data analysis and visualization [14, 46, 126, e.g.], retrieving document translation pairs [85] and modeling lexical variation across geographical regions [35], to name a few.

The most commonly used topic model is latent Dirichlet allocation (LDA) [16]. Figure 1.1 shows LDA in a graphical model "plate" notation for dynamic Bayes networks, which we will explain in Chapter 2. LDA models the generative process of a document collection by assuming that words w in a document are drawn from a document specific mixture of topics θ where each topic T is defined as a multinomial distribution over words in the collection φ, drawn from a Dirichlet prior β. A document D is generated through a three step process by first drawing a document specific topic distribution θ from a collection specific Dirichlet prior α (1). For each word in the document a topic assignment z is first drawn from θ (2) which specifies the topic-word distribution φ that will be used to draw the actual word (3). Steps (2) and (3) are repeated for each of the N words in the document. In most applications of LDA the main inferential problems are estimating the posterior document-topic distributions θ and topic-word distributions φ. This thesis focuses on retrieving documents based on their varying topical content. Therefore most of the inference problems here will involve document-topic distributions θ.

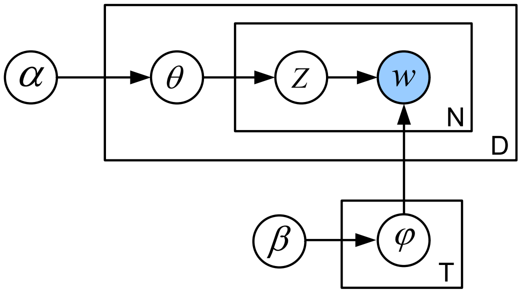


Figure 1.1: Graphical representation of the latent Dirichlet allocation (LDA) model using plate notation (§ 2.2.2.1).

One of the greatest advantages of using topic models is their ability to represent documents as probability distributions θ into a low-dimensional latent space—the T-dimensional probability simplex. Representing documents as points in a shared latent space abstracts away from the specific words used in a document, thereby facilitating the analysis of relationships between documents written using different vocabularies. This type of document representation makes it appealing for the IR and MT communities especially on tasks that involve document similarity search.

Utilizing latent variable models of text for document similarity search involves the following steps:

- Defining a model to represent documents in the latent space
- Inference about latent parameters, such as θ and φ
- Measuring similarity and performing search across inferred probability distributions
- Model evaluation

While various examples show their potential, topic models have not been fully utilized due to the constraints imposed in the outlined steps when processing large document collections. In this work we address the problems of efficient modeling, inference, search and evaluation for latent variable models of text and introduce novel approaches that streamline the use of these models on large document collections and large number of topics. We showcase the efficiency of our approaches on tasks of modeling large scientific articles, retrieving consecutively downloaded documents in a large scholarly IR system and extracting parallel sentences from comparable corpora.

In order for the applications of topic models to become practical, especially on large document collections, they need to be efficient. In this dissertation the governing question of efficiency, across the above outlined steps, will be discussed in terms of empirical tradeoffs between running time and accuracy. Rather than focusing on asymptotic efficiency or statistical efficiency we will discuss algorithmic efficiency from different points of view when dealing with large document collections and quantify it using measures such as the absolute running time of the approach, number of passes through the collection and constraints on memory use. In our efficiency analysis we will not perform comparison with approaches that utilize parallel architectures and in all cases we will assume that we are processing the data in a single stream.

In this thesis we will make the following contributions on each of the above steps:

- Define topic models that better model topic variations across textual regions in mono- and multilingual collections (**Chapter 7**). The inference time of these models will be used to analyze and quantify their efficiency. Across evaluation sets of mono- and multilingual documents these models achieve $\sim$2 times better representation compared to LDA and its multilingual variant – the polylingual topic model (PLTM) – as measured by perplexity which is an intrinsic evaluation metric commonly used to evaluate topic models.

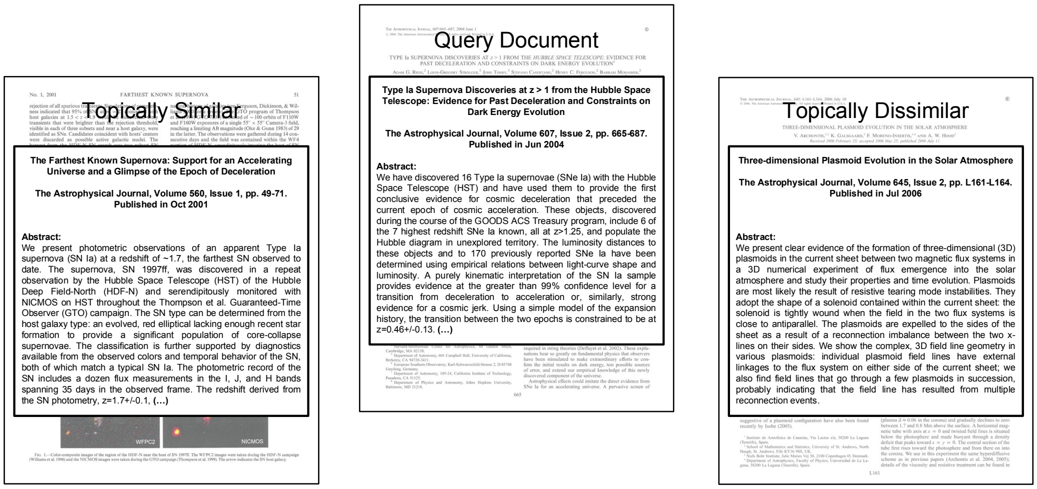
- Introduce fast inference for PLTM (**Chapter 4**) and topic models with multi-level hyperpriors (**Chapter 7**). The efficiency of our fast inference approach will be measured based on the running time, both in absolute time and in the number of passes through the collection. On a cross-language information retrieval (CLIR) task of finding document translation pairs our proposed inference approach is more than 3 times faster than the original PLTM inference while preserving similar accuracy.

- Present efficient nearest-neighbor (NN) search approach for inferred probability distributions (**Chapter 3**). We will quantify efficiency based on relative speed comparisons with regular similarity search baseline. On a patent retrieval task, an implementation of this approach is on average $\sim$105 times faster than regular similarity search in the probability simplex. On our CLIR task this approach achieved an average speed improvement of $\sim$30 times across different PLTM topic configurations. With our fast inference for PLTM and efficient NN search technique we were able to extract sentences from comparable corpora and train a MT system that outperforms a baseline system trained on parallel collection (**Chapter 6**). The efficiency aspect on this task is measured based on the amount and type of bilingual resources required to train a MT system.
- Introduce two efficient evaluation measures for topic models (**Chapter 5**). The efficiency aspect of the evaluation measures will be analyzed based on their computation time. On our two document similarity tasks of retrieving related patents and finding document translation pairs these metrics are more than 4 times faster to compute compared to two existing IR metrics and 25 times faster than perplexity. Across these two tasks and across ad-hoc web retrieval tasks we show that one of our proposed metrics achieves high correlation with two existing IR metrics. The same correlation was found by analyzing ten Text REtrieval Conference (TREC) tasks from the past ten years.

Detailed below are vignettes of these contributions in the order in which they are presented in this thesis along with a discuss on how we measure efficiency.

Efficient Nearest-Neighbor Search in the Probability Simplex

Many applications of latent variable models of text include comparing pairs of probability distributions to find close matches. For example, topic models, which represent documents as probability distributions over a fixed set of topics, are often used for finding



The Farthest Known Supernova: Support for an Accelerating Universe and a Glimpse of the Epoch of Deceleration

The Astrophysical Journal, Volume 560, Issue 1, pp. 49-71. Published in Oct 2001

Abstract:
We present photometric observations of an apparent Type Ia supernova (SN Ia) at a redshift of ~1.7, the farthest SN observed to date. The supernova, SN 1997ff, was discovered in a repeat observation by the Hubble Space Telescope (HST) of the Hubble Deep Field-North (HDF-N) and serendipitously monitored with NICMOS on HST throughout the Thompson et al. Guaranteed-Time Observer (GTO) campaign. The SN type can be determined from the host galaxy type: an evolved, red elliptical lacking enough recent star formation to provide a significant population of core-collapse supernovae. The classification is further supported by diagnostics available from the observed colors and temporal behavior of the SN, both of which match a typical SN Ia. The photometric record of the SN includes a dozen flux measurements in the I, J, and H bands spanning 35 days in the observed frame. The redshift derived from the SN photometry, z=1.7+/-0.1, **(…)**

Type Ia Supernova Discoveries at z > 1 from the Hubble Space Telescope: Evidence for Past Deceleration and Constraints on Dark Energy Evolution

The Astrophysical Journal, Volume 607, Issue 2, pp. 665-687. Published in Jun 2004

Abstract:
We have discovered 16 Type Ia supernovae (SNe Ia) with the Hubble Space Telescope (HST) and have used them to provide the first conclusive evidence for cosmic deceleration that preceded the current epoch of cosmic acceleration. These objects, discovered during the course of the GOODS ACS Treasury program, include 6 of the 7 highest redshift SNe Ia known, all at z>1.25, and populate the Hubble diagram in unexplored territory. The luminosity distances to these objects and to 170 previously reported SNe Ia have been determined using empirical relations between light-curve shape and luminosity. A purely kinematic interpretation of the SN Ia sample provides evidence at the greater than 99% confidence level for a transition from deceleration to acceleration or, similarly, strong evidence for a cosmic jerk. Using a simple model of the expansion history, the transition between the two epochs is constrained to be at z=0.46+/-0.13. **(…)**

Three-dimensional Plasmoid Evolution in the Solar Atmosphere

The Astrophysical Journal, Volume 645, Issue 2, pp. L161-L164. Published in Jul 2006

Abstract:
We present clear evidence of the formation of three-dimensional (3D) plasmoids in the current sheet between two magnetic flux systems in a 3D numerical experiment of flux emergence into the solar atmosphere and study their properties and time evolution. Plasmoids are most likely the result of resistive tearing mode instabilities. They adopt the shape of a solenoid contained within the current sheet: the solenoid is tightly wound when the field in the two flux systems is close to antiparallel. The plasmoids are expelled to the sides of the sheet as a result of a reconnection imbalance between the two x-lines on their sides. We show the complex, 3D field line geometry in various plasmoids: individual plasmoid field lines have external linkages to the flux system on either side of the current sheet; we also find field lines that go through a few plasmoids in succession, probably indicating that the field line has resulted from multiple reconnection events.

Figure 1.2: Finding topically similar articles in the Astrophysical Journal (ApJ). Scientific papers were represented as points in a shared topic space using LDA with 2000 dimensions. Shown in the center is an abstract of a query article which describes the preliminary results on the discovery of the expansion of the universe using Type Ia supernovae measurements. To the left of it is the abstract of the topically most similar retrieved article which also covers the topic of universe expansion. To its right is an abstract of topically unrelated article that talks about the magnetohydrodynamics of the sun.

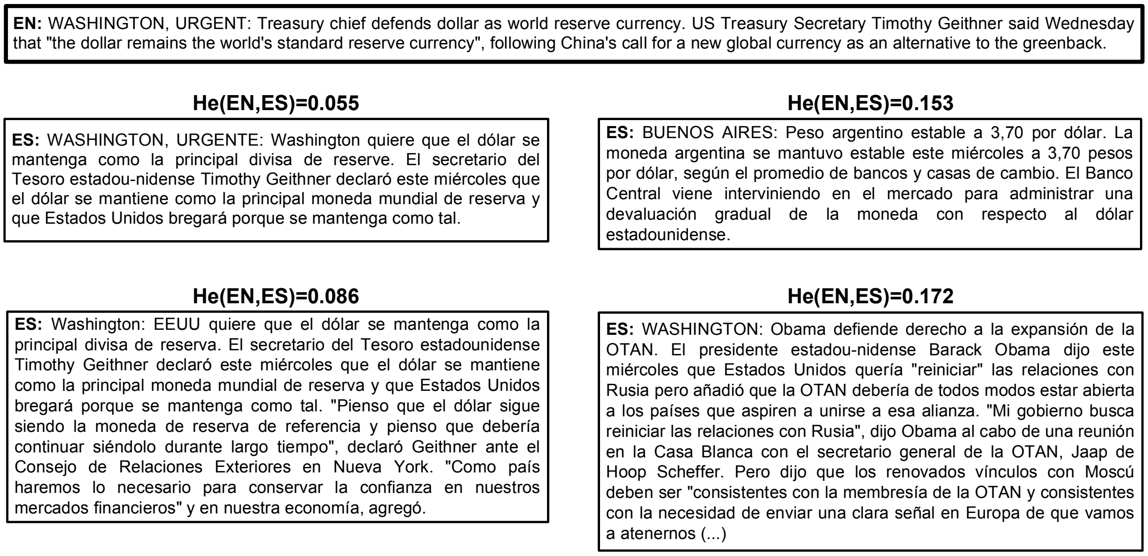
documents that are topically similar to each other. Example tasks of this nature include retrieving most topically relevant documents for a given query document, document clustering and finding document translation pairs in a multilingual collection. Figure 1.2 illustrates the application of LDA in finding topically similar scientific publications in the Astrophysical Journal (ApJ) – a peer-reviewed journal of astrophysics. Shown in this figure are the abstracts of a query article (center) and its most topically similar (left) and dissimilar (right) retrieved articles.

Unlike a metric space where two vectors are compared using metric based similarity measurements, such as Euclidean (Eu) or Cosine (Cos) distance, in the probability simplex similarity is computed using information-theoretic measurements such as the Kullback-Leibler (KL) and Jensen-Shannon (JS) divergence and Hellinger (He) distance. Topic distributions are continuous and therefore similarity computations for most topic distributions require $O(N^2)$ for near-neighbor detection tasks or $O(kN)$ computations when k queries are compared against a data set of N documents. As a result, the question of whether topic spaces would yield advantages to various document similarity tasks, especially on big, real-world data sets, has not really been explored.

To perform efficient similarity search on large collections in the probability simplex, in **Chapter 3** we frame the computation as an approximate NN search problem. More specifically, we present novel analysis and applications of the reduction of He divergence to Eu distance computations which allows us to exploit fast approximate NN techniques, such as locality-sensitive hashing (LSH) and approximate search in k-dimensional (k-d) trees in the probability simplex. For example, when using k-d trees we achieved orders of magnitude faster NN search compared to regular similarity search in the probability simplex across the following three different tasks: (1) when clustering scientific articles using LDA our speedup was more than 1,400; (2) using LDA to aid in prior-art retrieval for patents the average speedup across different topic configurations was more than 105; and (3) using PLTM to find document translation pairs in a large bilingual corpus we achieved an average speedup of more than 30 times across different topic configurations. On the patent retrieval task we also show that LDA could help achieve better retrieval performance by doing system combination with an existing IR model. Better retrieval model performance is also obtained when applying approximate NN computations in topic space on the task of speeding up a relevance model.

Online Polylingual Topic Model

A subset of extensions of LDA, such as PLTM, offer the ability to model multilingual collections. They assume that documents that are translations of each other, while written in a different language, cover the same set of topics. Mapping multilingual documents into a common latent topic space provide means to analyze the relationship between documents written in a different language. Applications of PLTM include: finding or detecting document translation pairs, creating translation lexicons, aligning passages and modeling



EN: WASHINGTON, URGENT: Treasury chief defends dollar as world reserve currency. US Treasury Secretary Timothy Geithner said Wednesday that "the dollar remains the world's standard reserve currency", following China's call for a new global currency as an alternative to the greenback.

He(EN,ES)=0.055

ES: WASHINGTON, URGENTE: Washington quiere que el dólar se mantenga como la principal divisa de reserve. El secretario del Tesoro estadou-nidense Timothy Geithner declaró este miércoles que el dólar se mantiene como la principal moneda mundial de reserva y que Estados Unidos bregará porque se mantenga como tal.

He(EN,ES)=0.153

ES: BUENOS AIRES: Peso argentino estable a 3,70 por dólar. La moneda argentina se mantuvo estable este miércoles a 3,70 pesos por dólar, según el promedio de bancos y casas de cambio. El Banco Central viene interviniendo en el mercado para administrar una devaluación gradual de la moneda con respecto al dólar estadounidense.

He(EN,ES)=0.086

ES: Washington: EEUU quiere que el dólar se mantenga como la principal divisa de reserva. El secretario del Tesoro estadounidense Timothy Geithner declaró este miércoles que el dólar se mantiene como la principal moneda mundial de reserva y que Estados Unidos bregará porque se mantenga como tal. "Pienso que el dólar sigue siendo la moneda de reserva de referencia y pienso que debería continuar siéndolo durante largo tiempo", declaró Geithner ante el Consejo de Relaciones Exteriores en Nueva York. "Como país haremos lo necesario para conservar la confianza en nuestros mercados financieros" y en nuestra economía, agregó.

He(EN,ES)=0.172

ES: WASHINGTON: Obama defiende derecho a la expansión de la OTAN. El presidente estadou-nidense Barack Obama dijo este miércoles que Estados Unidos quería "reiniciar" las relaciones con Rusia pero añadió que la OTAN debería de todos modos estar abierta a los países que aspiren a unirse a esa alianza. "Mi gobierno busca reiniciar las relaciones con Rusia", dijo Obama al cabo de una reunión en la Casa Blanca con el secretario general de la OTAN, Jaap de Hoop Scheffer. Pero dijo que los renovados vínculos con Moscú deben ser "consistentes con la membresía de la OTAN y consistentes con la necesidad de enviar una clara señal en Europa de que vamos a atenernos (...)

Figure 1.3: English news story with its topically most similar (left) and dissimilar (right) Spanish news stories discovered within a Hellinger (He) distance range of $He \in [0.0\ ,\ 0.2]$. News stories were extracted from the Gigaword collection using PLTM with 30 topics (Chapter 6).

comparable corpora. Figure 1.3 shows the result of applying PLTM on the task of detecting document translations in a comparable corpus of English-Spanish news stories. For the given English news story (top), PLTM discovered the two Spanish news stories on the left as the topically most similar and ranked the two news stories on the right as topically less similar. Similarity was computed using He distance between their representations in the topic space. We give a detailed explanation of this task in Chapter 6.

The process of discovering relationships between multilingual documents using PLTM starts with representing documents in a shared latent space by inferring their topic distributions. Almost all inference approaches for multilingual topic models, including the original PLTM, rely on Gibbs sampling to approximate posterior distributions. While more straightforward to implement, this inference approach requires iterating over the collection multiple times in order to obtain good topic estimates. As the size of the document collection grows, this type of an inference makes PLTM infeasible to use, especially for large multilingual collections with large number of topics.

In **Chapter 4** we introduce a new method for performing inference in PLTM. Our approach, which is based on online variational Bayes (VB) inference, provides a means for the efficient use of PLTM in large multilingual collections. Unlike Gibbs sampling it does not require multiple passes over the whole collection and could further be used on streams of data. For example, when modeling English and Spanish documents on the CLIR task of finding document translation pairs across different topic configurations our proposed inference approach is on average more than 3 times faster than, and as accurate as, PLTM.

Evaluating Topic Models through Histogram Analysis

Topic models are evaluated in one of two different ways - intrinsically and extrinsically. Intrinsic evaluations, such as perplexity, are typically used to determine the right number of topics and the optimal values of other model parameters. While useful for relative comparisons, perplexity is weakly correlated to the models' performance on a particular task.

Extrinsic evaluations, on the other hand, involve measuring the performance of the model on a specific task. For example, mean average precision (MAP) is used in many IR tasks. While extrinsic evaluations give us better insight into the models' ability to represent the document collection in context of a real world task, they are difficult to perform due to their dependence on an annotated collection of topically similar documents which is a scarce resource for large collections.

Computing perplexity requires traversing over each word in the test collection and integrating over all possible topic mixtures while MAP requires traversing over ranked lists. For large test collections and especially large numbers of topics (e.g. several thousand), the time that it takes to compute these metrics grows proportionally with the size of the test collection.

In **Chapter 5** we introduce two new extrinsic evaluation measures: Distributional Overlap (DO) that exploits the clustering of distances in topic space and Histogram Slope Analysis (HSA) that compares histograms, computed over the models' similarity metric, between query relevant and non-relevant documents in log space. Unlike perplexity, HSA achieves very high correlation with IR performance metrics, such as MAP and precision of the top rank (P@1), while being more efficient to compute. When used to evaluate LDA models with different topic configurations on a prior-art search task, both evaluation measures were on average more than 4 times faster to compute than MAP and more than 25 times faster than perplexity. When used to evaluate PLTM models with different topic configurations, these metrics were on average more than 5 times faster to compute than MAP.

In this chapter we also present a variant of HSA called random HSA (rHSA) which helps automate the process of evaluating retrieval models in large document collections such as a collection of scholarly publications. rHSA analyzes the histograms of scores computed over consecutively downloaded (CD) and randomly generated (RG) document pairs and rather than using human annotated evaluation sets it relies on pseudo-relevant sets of similar documents which are automatically generated from download logs. Across two families of retrieval models we show that rHSA achieves very high linear and rank correlation with MAP.

Bootstrapping Translation Detection and Sentence Extraction from Comparable Corpora

In **Chapter 6** we showcase the efficiency of our approaches for performing inference and NN search on the problem of detecting document translation pairs and extracting parallel sentences from comparable corpora. We utilize our fast, online based, inference for PLTM to represent multilingual documents in a shared topic space. Unlike typical multilingual topic models that are trained on parallel data, we generate the training set from comparable corpora using a minimally supervised approach for detecting and ranking document translation pairs that we previously developed [67]. Compared to previous approaches for extracting parallel sentences, this approach does not depend on linguistic resources such as parallel documents or translation dictionaries but only on observing documents published on similar dates and the co-occurrence of a small number of identical tokens across languages.

Additionally, with our approximate NN search techniques we achieve fast translation detection for documents represented in the probability simplex and the common metric space thus making our approach efficient for large comparable corpora. On the task of translation detection, we demonstrate that our approach achieves the same performance as a PLTM trained on parallel text. More importantly, using only sentences extracted from comparable corpora, we are able to train a MT system that outperforms a baseline system trained on a parallel collection.

Topic Models with Multi-Level Dirichlet Priors for Modeling Topical Variations across Textual Regions

Many latent variable models of text use the bag of words assumption when representing documents. This assumption simplifies the modeling process but at the same time discards document structure information such as word, paragraph and section ordering. Topic models infer document-topic distributions using document level word co-occurrence statistics that don't capture variations in topics found across large documents such as scientific articles, news stories published in a single day, parliamentary proceedings, etc. For example, scientific articles contain multiple sections whose topical relatedness is often not uniform. The introduction section could be topically less related to the conclusion then to other document sections. On tasks such as retrieving similar documents it is often the case that users may be more interested in discovering a specific document section that is topically related to the query rather than the whole document. Current topic models fail to address this issue and are inefficient in directly modeling sub-document level variations in topics.

In **Chapter 7** we propose to alleviate this problem by introducing two new topic models which we call, multi-level hyperpiors LDA (mlhLDA) and its multilingual variant – multi-level hyperpiors PLTM (mlhPLTM). Both models capture the topical variation across textual regions, such as document sections, in mono- and multilingual collections by assigning topic distributions to individual sections and modeling their relationship with the document-topic distribution using hierarchical structure over the Dirichlet priors. Developing mlhLDA and mlhPLTM allows us to more efficiently model large documents with prior structure information in mono- and multilingual collections. For example, on an evaluation set of ApJ articles, mlhLDA achieved 2.12 times better perplexity compared to LDA. While with mlhPLTM we achieved 2.42 and 1.87 times better perplexity over English and Spanish documents respectively in a collection of document translation pairs. On a CLIR task, over the same multilingual collection, mlhPLTM achieved 1.8 times better P@1 performance compared to our online implementation of PLTM.

CHAPTER 2

TOPIC MODELS

2.1 Introduction

Assume we are running a digital library which contains scientific articles form various Astrophysical journals[1]. A user has just downloaded a journal article that covers a recent results on analyzing the chemical composition of the sun and she is interested in finding articles that investigates the same or similar topic.

A common approach for solving this task would be to perform lexical and syntactical analysis of the articles in the collection, extract article features and use them to represent articles in a shared feature space. Solutions of this type utilize the information retrieval (IR) framework where article features are indexed and the document on the chemical composition of the sun is the query. While the scoring function depends on the nature of the features, in the IR framework most scoring functions compute the similarity between the features of the query and the documents in the collection and return a list of documents ranked based on the similarity score. Features generated based on lexical and syntactical analysis range from the basic, such as the ones that indicate whether words are present in the document, to the term and document frequency counts to features which are grounded on syntactic theories such as dependency grammar, to name a few.

Representing documents in a shared feature space goes beyond using the actual words in the document to find similar documents. In the past this type of document representation has been proven to yield good results across various document similarity tasks. Therefore

[1] http://ads.harvard.edu

using one such approach in our digital library would most certainly generate a list of similar articles that will be useful to the user.

However, features of this nature are strongly anchored with the words present in the document and they suffer from the general IR problem - vocabulary mismatch: while two documents may be presenting work on the chemical composition of the sun or, more generally speaking, may be covering the same topic or a set of topics, they may be using different words. Certainly, two journal articles will most likely have a large pool of overlapping words and most of the existing document similarity approaches would retrieve a reasonably well set of similar documents but it would most certainly fail to retrieve similar documents that differ a lot in their vocabularies. In addition, the dimensionality of the shared feature space is fairly big compared to the actual size of the documents in the collection, which can lead to inefficiency in large document collections.

To alleviate these problems researchers have looked into automatically representing documents by modeling their generative process. The underlying assumption in these unsupervised modeling approaches is that authors generate words in the document based on a topic or a set of topics that they are trying to convey to the reader. Regardless of the words used by the author, the reader ultimately learns and extracts the topics expressed by the author and therefore representing documents through a set of topics can provide a better semantic representation that also offers a more compact representation space.

Family of models that use this generative assumption are known as topic models and they are the central theme of this thesis.

Topic models fall under a much broader category of models known as latent variable models of text. In the past they have been showcased as effective tools for discovering hidden structure in document collections [46, e.g.]. In most cases they are being used for exploratory data analysis for analyzing dominant term collocations across documents in the collection or for analyzing term and topic preference over time [14, 126]. Wei & Croft [128] showed improvements on an IR task.

Their ability to represent documents as probability distributions in a low-dimensional latent space, that abstracts away from the specific words used in each document, allows for deeper semantic analysis of documents written using different vocabularies. For example, topic models have been used to identify scientific communities working on related problems in different disciplines, e.g., work on cancer funded by multiple Institutes within the National Institutes of Health (NIH) [115].

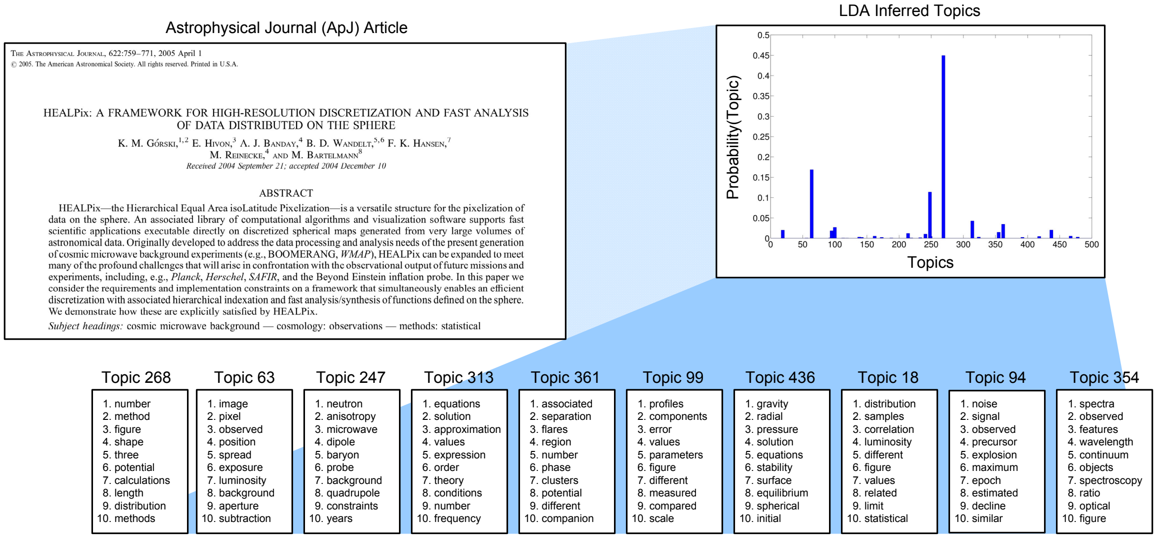
An exemplar topic model is latent Dirichlet allocation (LDA). With LDA, documents are represented as a mixture of topics where each topic is defined as a multinomial distribution over a collection wide vocabulary. The number of topics is determined a priori as well as the collection wide vocabulary which is typically constructed using term and document frequency statistics. In the next section we give a detailed technical overview of the model components and the steps involved in utilizing LDA.

Figure 2.1 showcases an example of representing one of the most cited Astrophysical Journal (ApJ) article (across all articles there were published in the past 10 years) using LDA configured with 500 topics. LDA represents documents as multinomial probability distributions by inferring the posterior document-topic distribution.

When we represent all the ApJ articles in our digital library as multinomial distributions over topics, the task of finding articles that investigate the same topic or set of topics as the user's downloaded article is recast to finding similar probability distributions.

2.2 Technical Background

Latent variable models of text supplement the observed textual units, such as words, using hidden or latent variables. Modeling both, the observed and latent variables, allows for the complex distribution over the observed variables to be represented through more constrained distributions over the hidden variables. Probabilistic topic models are an exemplar latent variable models of text where the hidden variables are distributions over a fixed set of topics. They use the simple assumption that documents in the collection are a



THE ASTROPHYSICAL JOURNAL, 622:759–771, 2005 April 1
© 2005. The American Astronomical Society. All rights reserved. Printed in U.S.A.

HEALPix: A FRAMEWORK FOR HIGH-RESOLUTION DISCRETIZATION AND FAST ANALYSIS OF DATA DISTRIBUTED ON THE SPHERE

K. M. GÓRSKI,[1,2] E. HIVON,[3] A. J. BANDAY,[4] B. D. WANDELT,[5,6] F. K. HANSEN,[7]
M. REINECKE,[4] AND M. BARTELMANN[8]

Received 2004 September 21; accepted 2004 December 10

ABSTRACT

HEALPix—the Hierarchical Equal Area isoLatitude Pixelization—is a versatile structure for the pixelization of data on the sphere. An associated library of computational algorithms and visualization software supports fast scientific applications executable directly on discretized spherical maps generated from very large volumes of astronomical data. Originally developed to address the data processing and analysis needs of the present generation of cosmic microwave background experiments (e.g., BOOMERANG, *WMAP*), HEALPix can be expanded to meet many of the profound challenges that will arise in confrontation with the observational output of future missions and experiments, including, e.g., *Planck*, *Herschel*, *SAFIR*, and the Beyond Einstein inflation probe. In this paper we consider the requirements and implementation constraints on a framework that simultaneously enables an efficient discretization with associated hierarchical indexation and fast analysis/synthesis of functions defined on the sphere. We demonstrate how these are explicitly satisfied by HEALPix.

Subject headings: cosmic microwave background — cosmology: observations — methods: statistical

Figure 2.1: Representing an ApJ article using LDA with 500 topics. In the upper right corner we show the inferred article topics which is the document-topic multinomial distribution generated by the model. Shown on the bottom are the topic-word distributions for each of the ten most probable topics in the article. For each topic-word distribution we present a ranked list of the top ten most probable words for that topic.

mixture of topics θ and jointly model the process of generating the words in a document through a hierarchical process that involves three steps:

- Draw a topic distribution from the fixed set of topics and assign it to the document
- Go over each word position in the document and draw a topic from the assigned set of document topics
- Based on the drawn topic, choose the actual word from the topic specific distribution over words

As we have mentioned, the most widely used probabilistic topic model is LDA. In this chapter we give a detailed presentation of LDA and its multilingual variant the polylingual topic model (PLTM). LDA is normally considered as an unsupervised topic model. One easy way to understand LDA is to first explain naive Bayes (NB) classifier which is a supervised classification technique that, from Bayesian inference perspective, is very similar to LDA. Understanding LDA, PLTM and even NB requires explaining probabilistic graphical models and the simplifying assumption for modeling text using multinomial distributions. Bayesian inference with multinomial distributions requires good prior distributions over them such as the Dirichlet distribution. We will present the Dirichlet distribution and its hyperparameter α and the concept of conjugacy which simplifies the relationship between prior and posterior distributions. We start by introducing basic concepts from the probability theory that are necessary for understanding these concepts.

We will also present two major inference procedures used by LDA and PLTM. We conclude the chapter by presenting the important steps that are required to preprocess a document collection.

2.2.1 Probability Distributions

Given a probability variable X and a space Ω of all possible values i.e. outcomes, such that $X \in \Omega$, the probability distribution P is a function of X, $P : F_X$ that assigns probability, i.e. a real value number $\Re^+$ in a range of $[0, 1]$, to each outcome: $P : F_X \rightarrow [0, 1]$. Probability variables could be either discrete or continuous. Discrete probability variables could take on a finite set of values compared to continuous variables where the set is infinite. To give an example, let's assume that we are analyzing the grades in a math class that has 30 students. Since grades could take on a finite set of values (e.g. A, B, C, D and F) the probability of a student achieving a particular grade would be a discrete probability variable. On the other hand if we were to measure the width or the height of every student we would be dealing with continuous probability variable since its range of values would be continuous.

The probability distribution is defined based on the nature of the probability variable. When dealing with discrete variables, the probability distribution is characterized by the probability mass function (pmf) as it assigns values to all possible outcomes. For a given discrete probability variable X with a finite set of n outcomes $x_i = x_1, x_2, x_3, ..., x_n$ where $x_i \in \Omega$ the probability distribution is a vector whose dimensionality is equal to the number of outcomes and its values are the probabilities across the outcomes such that $\sum_{i=1}^{n} P(X = x_i) = 1$.

Since continuous probability variables could take on an infinite number of values the probability of assuming any one particular value would be zero. Therefore they are characterized using the probability density function (pdf). Given a continuous probability variable X, the pdf assigns the probability of the variable assuming a value in an interval $[a, b]$: $P[a \leq X \leq b] = \int_a^b F(x)\, dx$. The pdf assigns density of the probability mass in the given interval and therefore it is the equivalent to the pmf for continuous variables.

2.2.1.1 Sum and Product Rules

The sum and product rules, along with the Bayes' rule, that we will present in the next section, are the most important rules of probability needed to understand the relationship between variables in probabilistic topic models. Given two discrete random variables X and Y where X consists of set of N discrete values or outcomes $X = x_1, x_2, x_3, ..., x_n$ and similarly Y consists of M such values $Y = y_1, y_2, y_3, ..., y_m$ their joint probability is defined as $p(X = x_i, Y = y_j)$ such that $\sum_{i=1}^{N} \sum_{j=1}^{M} p(X = x_i, Y = y_j) = 1$.

When we are given a joint probability over two probability variables and we are interested in observing only one of them we would need to consider all values of the other variable and sum them up. This process is referred to as marginalizing and is computed using the sum rule:

$$p(X = x_i) \quad = \quad \sum_{j=1}^{M} p(X = x_i, Y = y_j) \tag{2.1}$$

When we are given a joint probability distribution and we constrain ourselves to observing or considering only a particular value for the Y variable (e.g. $Y = y_3$) then we transition from a joint probability to a conditional probability: $p(X = x_i | Y = y_j)$. The relationship between the joint and the conditional probabilities is defined by the product rule:

$$p(X = x_i, Y = y_j) \;=\; p(X = x_i|Y = y_j)p(Y = y_j) \tag{2.2}$$

2.2.1.2 Bayes' Theorem

Combining the sum and the product rules defines one of the most important theorems in the probability theory - Bayes' theorem or often referred to as Bayes' rule:

$$p(X = x_i|Y = y_j) \;=\; \frac{p(Y = y_j|X = x_i)p(X = x_i)}{p(Y = y_j)} \tag{2.3}$$

With the sum rule we could represent the denominator in the Bayes' theorem as $p(Y = y_j) = \sum_{i=1}^{N} p(Y = y_j|X = x_i)p(X = x_i)$:

$$p(X = x_i|Y = y_j) \;=\; \frac{p(Y = y_j|X = x_i)p(X = x_i)}{\sum_{i=1}^{N} p(Y = y_j|X = x_i)p(X = x_i)} \tag{2.4}$$

Bayes' rule is often used for interpreting probability from the perspective of quantifying the uncertainty or the degree of belief, i.e. the Bayesian interpretation. In this interpretation we start by quantifying the known uncertainty through the marginal probability P(X) which is our prior belief of the process that we are modeling.

When Y outcomes are observed from the process we deal with a degree of belief or uncertainty which is conditioned on our previous belief $P(Y|X)$. From Bayes' interpretation of probability this defines our likelihood. In other words it quantifies our belief of how likely the given data is given the prior. When multiplied with the prior and normalized with the denominator the product gives us the posterior belief or uncertainty $P(X|Y)$.

2.2.1.3 Multinomial Distribution

Probability distributions could also vary depending on their parameter/s. In probability theory there is a well known set of probability distributions that are commonly used to describe various generative processes. For example, let's assume that we would like to model the coin toss process where we assign the probability of the coin coming up "heads" $x_1 = 0.5$ and probability of coming up "tails" $x_2 = 0.5$. If we are dealing with a single coin toss then a well known distribution that models this process is the Bernoulli distribution. If we observe multiple coin tosses, say n rather than a single one, and we would like to model the probability of getting exactly k heads then we would use the Binomial distribution.

In this section we describe the multinomial distribution which is a generalization of the Binomial distribution where rather than having two possible outcomes (e.g. "heads" or "tails") we have K outcomes. The multinomial distribution is very important in understanding the modeling of the generative process used by various latent variable models of text.

If we are given a set of K possible outcomes $x_i = x_1, x_2, x_3, ..., x_k$ with a set of probability values for each outcome $p_i = p_1, p_2, p_3, ..., p_k$ such that $P(X = x_i) = p_i$ and an instance of N observations where we have $c_i = c_1, c_2, c_3, ..., c_k$ representing the number of instances of the outcome x_i being observed in the N observations, the multinomial distribution defines the probability of any combinations of outcomes across the set of K outcomes. Its pmf has the following functional form:

$$f\left(c_1, c_2, c_3, ..., c_k | p_1, p_2, p_3, ..., p_k\right) = \frac{(\sum_{i=1}^{k} c_i)!}{\prod_{i=1}^{k} c_i!} \prod_{i=1}^{k} p_i^{c_i} \tag{2.5}$$

In the above equation the fraction $\frac{(\sum_{i=1}^{k} c_i)!}{\prod_{i=1}^{k} c_i!}$ is the multinomial coefficient which quantifies the number of ways that we could divide the set of observations N into subsets of size x_1 to x_k. This coefficient could be represented using the Gamma function $\Gamma(x)$:

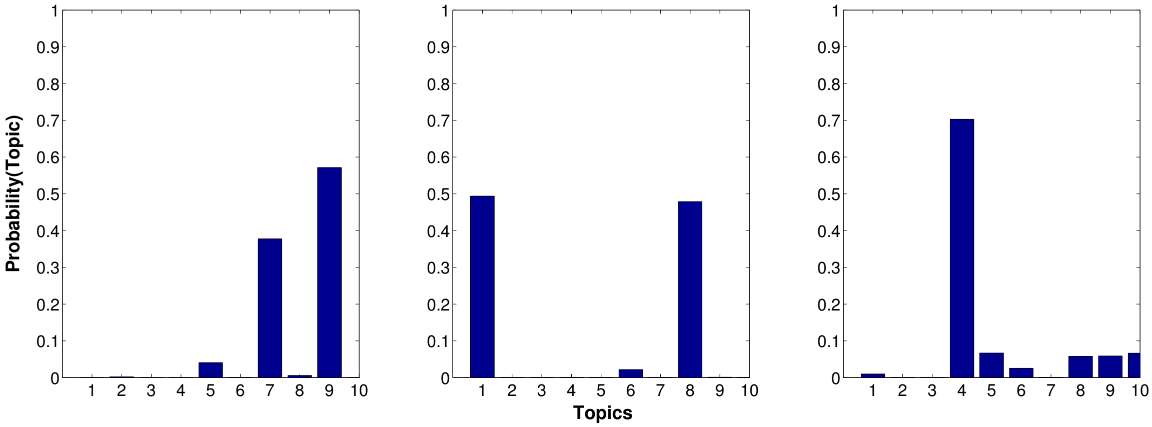


Figure 2.2: Examples of three multinomial distributions over ten topics. The y-axis represents the probability of the topic being present in the document.

$$\frac{(\sum_{i=1}^{k} c_i)!}{\prod_{i=1}^{k} c_i!} = \frac{\Gamma\left(\sum_{i=1}^{K} c_i + 1\right)}{\prod_{i=1}^{K} \Gamma\left(c_i + 1\right)} \tag{2.6}$$

Figure 2.2 shows examples of three multinomial distributions over ten possible outcomes which in this case are equivalent to a set of ten possible topics that have been assigned to three documents.

2.2.1.4 Dirichlet Distribution

The Dirichlet distribution is a continuous distribution over a family of multinomial distributions. Often abbreviated as “Dir” this distribution is parametrized by a vector α: $Dir(\alpha)$, referred as the hyperparameter, that controls the sparsity of the family of multinomial distributions. In a $K-1$ dimensional probability simplex the pdf of the Dirichlet distribution is defined as:

$$p(x_1, x_2, x_3, ..., x_n | \alpha_1, \alpha_2, \alpha_3, ..., \alpha_n) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma\left(\alpha_i\right)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{2.7}$$

The hyperparameter α is a K dimensional vector $[\alpha_1, \alpha_2, ..., \alpha_K]$ where α_j is interpreted as the number of times topic j has been seen in a document as if the document has already been observed.

It is often written as a product of a scalar (i.e. the concentration parameter α) and a vector of base measures u, $\hat{\alpha} = \alpha u$. Typically the Dirichlet prior is symmetric which means that the concentration parameter is fixed and the base measure across all topics is uniform. Values of symmetric hyperparameters are usually given in terms of the concentration parameter which implies that the base measure is a vector of ones. Figure 2.3 shows five multinomial distributions drawn from five different Dirichlet distributions with different symmetric hyperparameters $\alpha \in [0.01 - 100.0]$. Dirichlet distributions are defined over a 50 dimensional probability simplex which in this example are interpreted as topics. With smaller concentration parameters the family of multinomials contains sparse distributions. As the concentration parameter increases the probability mass tends to spread out evenly over the possible outcomes (i.e. the topics) and with very high values ($\alpha \geq 100$) the family of multinomials consists of almost uniform distributions.

The values of the symmetric hyperparameter (i.e. the concentration parameter) are typically set using certain heuristics which are based on the document collection. In case of LDA however it is often the case that the values of $\alpha = \frac{50}{T}$ and $\beta = 0.01$ are being used for the document-topic and the topic-word distributions. These values, which are de-facto being used as default values, were originally introduced by Steyvers and Griffiths [111]. Another way to set the values of the Dirichlet hyperparameter is through estimation such as the maximum likelihood (ML) estimation proposed by Minka [86].

An alternative to the symmetric prior is the asymmetric prior where the base measures are non-uniform. Recently, Wallach et al. [124] showed that in the LDA model, using asymmetric Dirichlet priors over document-topic distributions offer modeling advantage over symmetric priors as measured by perplexity. This approach treats the base measures vector as a hidden variable and assigns a symmetric Dirichlet prior to it which in turn

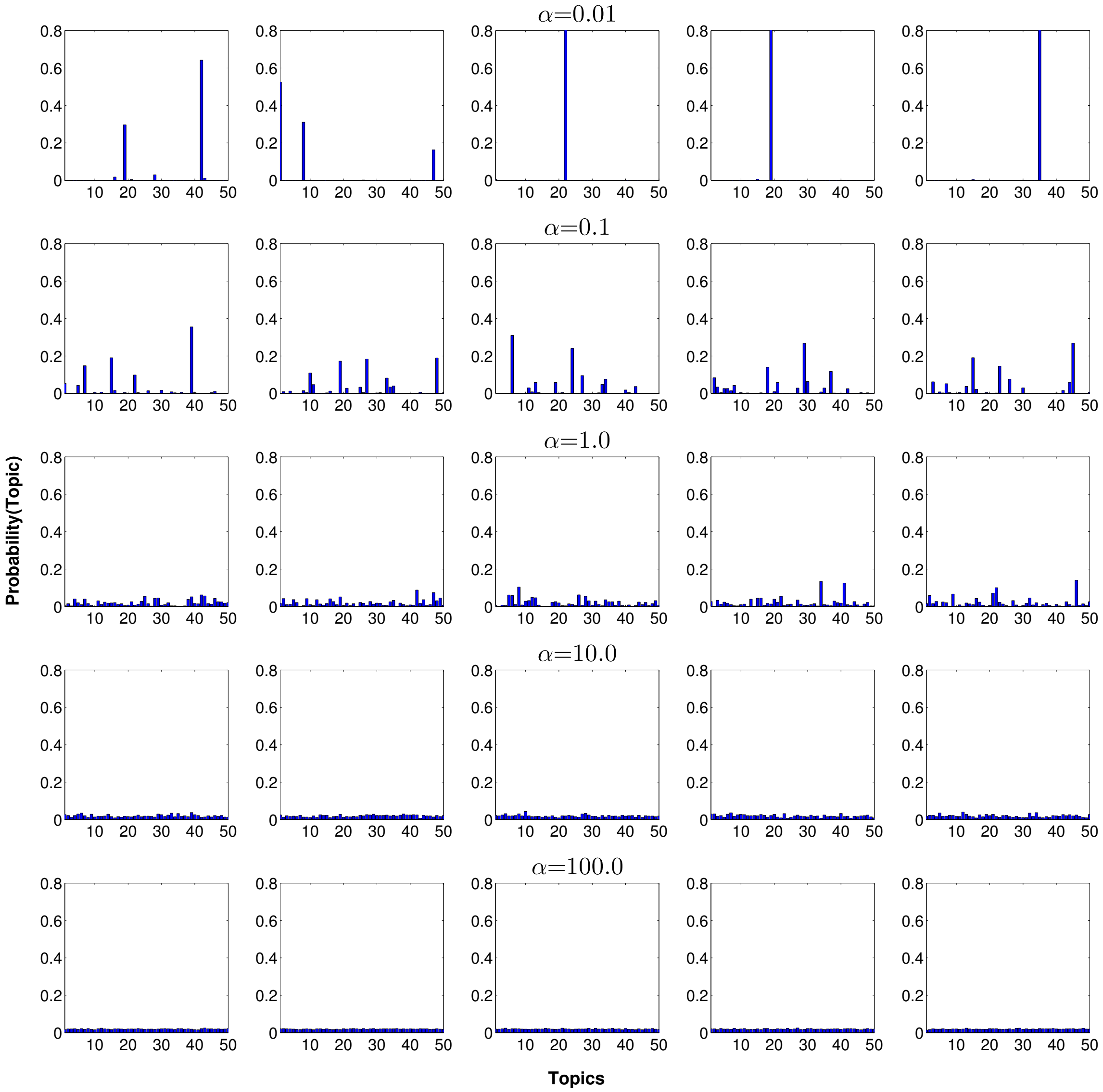


Figure 2.3: Example of multinomial distribution samples drawn from Dirichlet distributions with different symmetric hyperparameters (i.e. concentration parameters).

creates a hierarchical Dirichlet prior structure over all document-topic distributions in the collection.

2.2.1.5 Conjugacy

An important concept for simplifying and understanding the relationship between the prior and posterior distributions in the modeling process of various latent variable models of text is conjugacy. Let's assume that we are given a data set D and a parameter θ that models the data. Based on the product rule the joint probability of observing the data and the parameter would be equal to the likelihood multiplied by the prior $P(D|\theta)P(\theta)$. The concept states that a probability distribution P_1 is a conjugate to a probability distribution P_2 if when assigned as a prior distribution $P_1(\theta)$ and multiplied with the likelihood distribution $P_2(D|\theta)$ we obtain a posterior distribution of the same functional form as the likelihood - P_2. More general, if the posterior distribution is from the same family of distributions as the prior, i.e. they have the same functional forms, then the prior and the likelihood distributions are conjugate pairs. Across different generative models it is more convenient to use conjugacy pairs as they simplify the description of the generative process and provide mathematical convenience when deriving the computations for the posterior estimations in the model.

In probability theory there exists a set of well known conjugate pairs such as: Beta-Binomial, Gamma-Poisson, Gamma-Exponential, etc. We illustrate the conjugacy using the Dirichlet-Multinomial conjugate pair which is also used in the LDA inference.

Let's assume that we have a multinomial distribution which is parametrized by θ, a point in a k-dimensional probability simplex, and that the distribution is drawn from a Dirichlet distribution with a symmetric parameter α (i.e. the concentration parameter). Let's also assume that we sample an outcome x_i from θ:

$$\theta \sim Dir(\alpha) \tag{2.8}$$

$$x_i \sim Miltinomial(\theta) \tag{2.9}$$

Following the product rule, the joint probability of $p(\theta, x)$ for N outcomes is defined as:

$$p(\theta, x) = p(\theta|\alpha) \prod_{i=1}^{N} p(x_i|\theta) \tag{2.10}$$

The posterior probability is defined using Bayes' rule:

$$p(\theta|x) = \frac{p(\theta, x)}{p(x)} \tag{2.11}$$

Putting the functional forms of the multinomial (Equation 2.5) and the Dirichlet distributions (Equation 2.7) together we get:

$$\begin{aligned} p(\theta|x) \sim & \frac{\Gamma\left(\sum_{i=1}^{K} c_i + 1\right)}{\prod_{i=1}^{K} \Gamma\left(c_i + 1\right)} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma\left(\alpha_i\right)} \prod_{i=1}^{K} x_i^{c_i} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \\ &= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i + c_i + 1\right)}{\prod_{i=1}^{K} \Gamma\left(\alpha_i + c_i + 1\right)} \prod_{i=1}^{K} x_i^{\alpha_i - 1 + c_i} \end{aligned} \tag{2.12}$$

From the above equations we clearly see that the functional form of the posterior distribution is the same as the prior distribution as they both define a Dirichlet distribution and the parameters of that distribution combine α with counts c.

2.2.2 Probabilistic Graphical Models

The sum and product rules along with the Bayes' theorem are one of the core concepts required for understanding the relationship between probability distributions and explicitly representing the joint distribution across different, model specific, probability variables. In

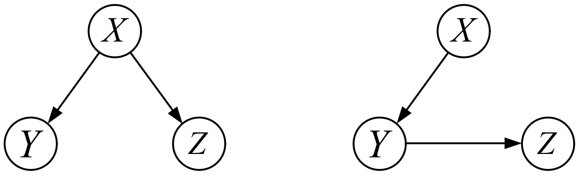


Figure 2.4: Simple directed graphical model representations of two joint probability distributions.

instances where we are dealing with a small set of probability variables it is straightforward to comprehend and represent the inter-variable relationship in the joint distribution. However, many real-world processes require probability models with large set of variables which in turn makes the definition and algebraic representation of their joint distributions more complex. Probabilistic graphical models represent constraints on the joint distribution using edges between component random variables. These models depict probability variables as nodes or vertices in the graph while the edges or arcs represent the probabilistic relationship between variables. This representation type helps visualize the relationship between probability variables. At the same time they represent complex relationships across different probability distributions in a compact way. In this section we give overview of the basic elements of graphical models that are required for understanding the graphical representation of latent variable models of text. As we will show in Section 2.2.4 the structure of topic models is best described and represented using probabilistic graphical models.

There are two types of graphical models - directed and undirected. Directed graphical models are often referred to as Bayesian or belief networks and sometimes as causal networks. These models use directed acyclic graphs (DAGs) to represent the causal relationship between variables. In other words, edges between nodes in these graphs have directions which represent the causal relationship between the variables. Figure 2.4 shows examples of two directed graphical models between three probability variables X, Y and Z which depict the following two joint probability distributions (from left to right): $P(X, Y, Z) = P(X)P(Y|X)P(Z|X)$ and $P(X, Y, Z) = P(X)P(Y|X)P(Z|Y)$.

Undirected graphical model are also referred to as Markov random field (MRF). Unlike directed graphical models where the conditional dependence is modeled explicitly by the direction of the edge, they don't model the causal relationship and are more suited for modeling processes where the dependence assumption is not natural such as in modeling spatial data. For example modeling the dependence amongst pixels in an image is more natural to be achieved using undirected graphical models. Regardless of the modeling process, in the graphical representation of the model, nodes of the observed variables are shaded in order to distinguish them from hidden or unobserved variables.

2.2.2.1 Plate Notation

Probabilistic graphical models offer a compact representation of the joint distribution across the probability variables in a model. In instances when the probabilistic model contains a large set of variables the number of nodes present in the graph could easily expand to a point where the model could become cluttered and difficult to interpret. In such graphs it is often the case that there exists subgraphs with repeated structure. This is certainly the case when variables are generated by independent draws from the same distribution. The plate notation avoids this clutter by drawing a rectangle i.e. "plate" around the repeated nodes and substituting them with a single node where the number of repeated nodes is shown in the lower right corner of the plate. As an example, in Figure 2.5 we show the graphical model of the joint probability $P(X,Y) = P(X)\prod_{i=1}^{N} P(Y_i|X)$ and its graphical representation using plate notation. Representing a graphical model with all its nodes from its plate notation is referred to as an unrolled model.

In some graphical models it is common for a repeated structure to occur on multiple levels. For example consider the joint probability distribution of the three variables shown in Figure 2.6. We have $P(X,Y,Z) = P(X)\prod_{j=1}^{2}\sum_{i=1}^{N} P(Z_{i,j}|Y_i)P(Y_i|X)$. In this case the graphical model could be made more compact by introducing a second plate which is referred to as nested plate notation.

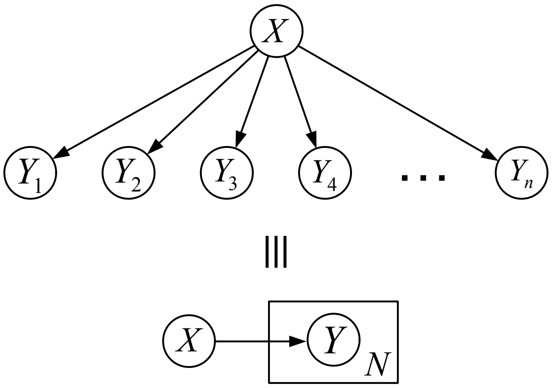


Figure 2.5: Example graphical model representation of a joint probability distribution $P(X, Y)$ and its plate notation equivalent.

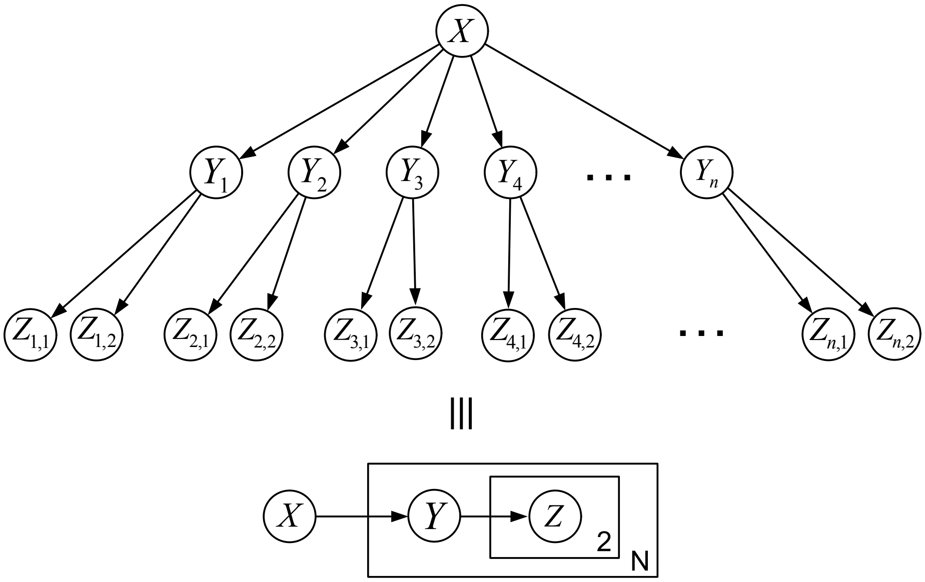


Figure 2.6: Nested plate notation and its unrolled graphical model equivalent.

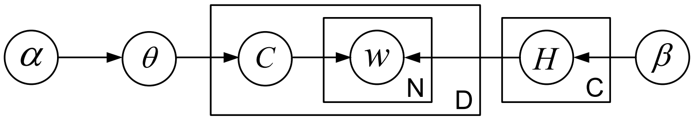


Figure 2.7: Graphical representation of the Bayesian Naive Bayes (BNB) model.

2.2.3 Bayesian Inference for Naive Bayes Text Classification

A simple example of a latent variable model of text that uses the Bayesian conjugacy and models the generative process of documents in a collection is the Bayesian Naive Bayes (BNB) classifier [90]. Similar to the NB [34] classifier, BNB has a "naive" assumption that when assigning a class C to a document in the collection one should expect that words in that document are generated interdependently given the class. However, unlike NB which uses ML estimates of the parameters, in BNB we do full Bayesian inference. With the BNB model the generative process of a document d in a collection is modeled by first assigning a class to a document. The class prior is drawn from a Dirichlet distribution. A class in BNB is an actual multinomial distribution over all the words in the collection and the naive assumption is that given a class, i.e. a multinomial distribution, words drawn from the distribution are independent of each other. The graphical model representation of the BNB is shown in Figure 2.7.

BNB assumes that the following process generates each document D in a collection with vocabulary V:

- Draw a multinomial distribution θ over classes C from a collection wide Dirichlet distribution with hyperparameter α: $\theta \sim Dir(\alpha)$.
- For each class C draw a multinomial distribution H_c over the collection vocabulary V. H_c is drawn from a collection wide Dirichlet distribution with hyperparameter β: $H_c \sim Dir(\beta)$.
- For each document $d = 1, 2, ..., D$ in the collection assign a category C_d by drawing from θ: $C_d \sim Multinomial(\theta)$.

- Go over each word position $n = 1, 2, 3, ..., N_d$ in the document and draw the actual word w_n from the category specific distribution over words: $w_n \sim Multinomial(H_c)$.

Due to the Dirichlet-multinomial conjugacy and the naive independence assumption, the posterior distributions of θ and H_c have exact solutions provided by Equation 2.12 which for a training set of D documents is:

$$p(\theta|D) \sim \frac{\Gamma(\sum_{i=1}^{C} \alpha_i + \delta(c_i) + 1)}{\prod_{i=1}^{C} \Gamma(\alpha_i + \delta(c_i) + 1)} \prod_{i=1}^{C} x_i^{\alpha_i - 1 + \delta(c_i)} \tag{2.13}$$

Where δ is an indicator function of the number of times category c_i was assigned to documents in the training collection. Equivalently we have the same exact solution for the posteriors over H_c. Given hyperparameters α and β the joint probability distribution in BNB is given by the following equation:

$$P(\theta, H, C_d, w|\alpha, \beta) = p(\theta|\alpha) \prod_{c=1}^{C} p(H_c|\beta) p(C_d|\theta) \prod_{n=1}^{N_d} p(w_n|H_c) \tag{2.14}$$

2.2.4 Latent Dirichlet Allocation (LDA)

The BNB model suffers from two problems: having a single class assigned to a document and more importantly has a “naive” independence assumption about the words generated in the documents. LDA on the other hand assigns a hidden variable over the topic mixtures. Since in the BNB model there is an exclusive OR over the possible categories for each document, it is also referred to as a “mixture” model. While LDA is referred to as an “admixture” model since each document has a distribution over topics. For a collection of D documents with a collection wide vocabulary of V words, the model assumes that its only observable variable are the actual words in the document and that the number of topics in the collection is a priori set to T. For each topic $t = 1, 2, ..., T$ in the collection, the model first draws a V dimensional multinomial distribution φ_t from a prior Dirichlet distribution with hyperparameter β: $\varphi_t \sim Dir(\beta)$. For each document $d = 1, 2, ..., D$ in the collection, LDA's generative process includes the following steps:

- Draw a multinomial distribution θ_d from a collection wide Dirichlet distribution with hyperparameter α: $\theta_d \sim Dir(\alpha)$.
- Go over each word position $n = 1, 2, 3, ..., N_d$ in document d and assign a topic indicator z_n by drawing topic from θ_d: $z_n \sim Multinomial(\theta_d)$.
- Based on the drawn topic assignment $z_n = t$, draw the actual word w_n from the topic specific distribution over words: $w_n \sim Multinomial(\varphi_{z_n})$.

The above process is then repeated for every document in the collection. In Figure 2.8 we show the unrolled graphical model of LDA and its equivalent model with plate notation.

Given hyperparameters α and β the joint probability distribution in LDA is given by the following equation:

$$P(\theta_d, \varphi, z, w|\alpha, \beta) = p(\theta_d|\alpha) \prod_{t=1}^{T} p(\varphi_t|\beta) \prod_{n=1}^{N_d} p(z_n|\theta_d) p(w_n|z_n, \varphi) \tag{2.15}$$

In LDA we are dealing with two Dirichlet distributions which are used as prior distributions for the document-topic distribution θ_d:

$$p(\theta_d|\alpha) = \frac{\Gamma\left(\sum_{t=1}^{T} \alpha_t\right)}{\prod_{t=1}^{T} \Gamma\left(\alpha_t\right)} \prod_{t=1}^{T} \theta_t^{\alpha_t - 1} \tag{2.16}$$

And the topic-word distributions φ_t:

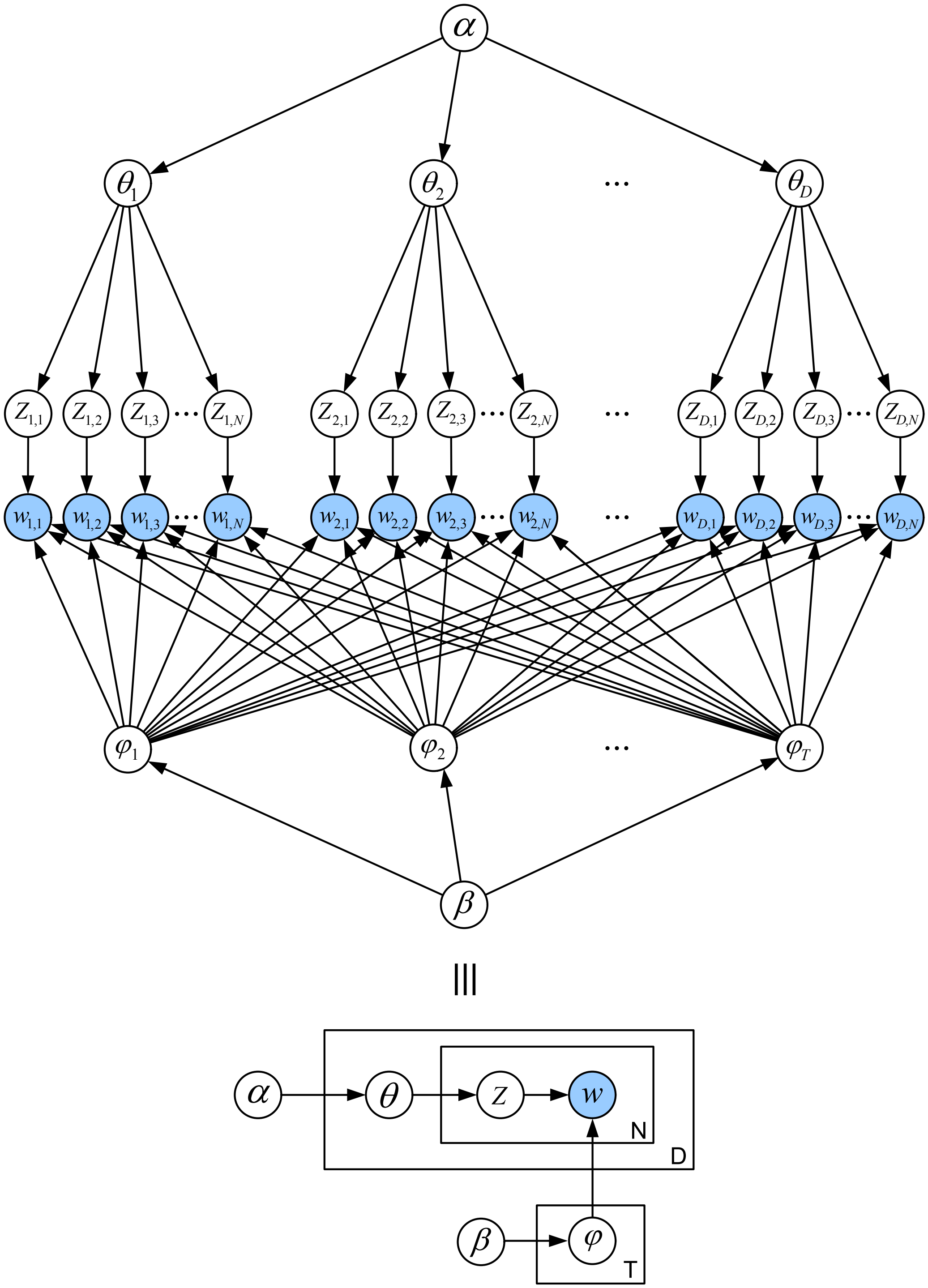


Figure 2.8: Unrolled graphical model representation of LDA.

$$p(\varphi_t|\beta) = \frac{\Gamma\left(\sum_{v=1}^{V}\beta_v\right)}{\prod_{v=1}^{V}\Gamma\left(\beta_v\right)}\prod_{v=1}^{V}\varphi_v^{\beta_v-1} \tag{2.17}$$

The probability of document d in the collection which consists of w words is computed by marginalizing out the document-topics distribution θ_d and the topics assignment variable z:

$$P(w|\alpha,\beta) = p(\varphi_t|\beta)\frac{\Gamma\left(\sum_{t=1}^{T}\alpha_t\right)}{\prod_{t=1}^{T}\Gamma\left(\alpha_t\right)}\int\left(\prod_{t=1}^{T}\theta_t^{\alpha_t-1}\right)\left(\prod_{n=1}^{N_d}\sum_{t=1}^{T}\prod_{v=1}^{V}\left(\theta_t\varphi_{tv}\right)^{w_n^v}\right)d\theta \tag{2.18}$$

For a given document d with a set of words w the main goal of LDA is to infer mixture of topics θ_d over the document and to infer topic assignments z to words w: $p(\theta, z|w, \alpha, \varphi)$. Using Bayes' theorem we have the following algebraic layout:

$$P(\theta, z|w,\alpha,\varphi) = \frac{P(\theta, z, w|\alpha,\varphi)}{p(w|\alpha,\varphi)} \tag{2.19}$$

In many latent variable models of text computing the posterior probability distribution over latent variables is often intractable. The same holds for LDA. Due to the product between the two multinomial distributions θ and φ (Eq. 2.15) the exact computation of the posterior distribution in the above equation is not computationally feasible and therefore approximate inference methods have been proposed.

2.2.5 Inference in LDA

When inferring posterior distributions with approximate methods researchers often refer to three approximate approaches: Gibbs sampling, variational Bayes (VB) and expectation-propagation. Most widely used of the three approaches is the Gibbs sampling approach followed by VB. These two approaches formulate the approximate inference as a sampling (Gibbs) and optimization (VB) tasks. In this section we layout both approaches and give an overview of their algorithms.

2.2.5.1 Gibbs Sampling

Gibbs sampling is a variant of the Markov chain Monte Carlo (MCMC) method which constructs a Markov chain whose states are parameter settings and whose stationary distribution is the true posterior over those parameters. Rather than estimating the posterior document-topic θ_d and the topic-word φ_t distributions, Gibbs sampling first estimates the posterior topic assignments z which are then used to approximate θ_d and φ_t. The Gibbs sampling directly utilizes the Dirichlet-multinomial conjugacy. For a collection of D documents, the algorithm iterates over each word in the collection and assigns a topic t to the current word w_n ($z_n = t$) conditioned on the topic assignments of all the other words in the collection which is often denoted as z_{-n}. This defines the conditional probability $p(z_n = t|z_{-n}, w_n, d_n, \cdot)$. In this section we present the Gibbs sampling algorithm using the syntax and naming convention adopted by Griffiths and Steyvers in their original work that introduced Gibbs sampling for LDA [43]. In this work the "$\cdot$" symbol is used to represent all other hidden and observed variables in the LDA model including the hyperparameters. The underlying data structure that the algorithm requires are basic word and topic assignment counts on a document and collection level. Griffiths and Steyvers use matrices for this purpose. More specifically they use the matrix C^{VT}_{wt} to store the number of times that word w is assigned to topic t discarding the current word topic assignment and C^{DT}_{dt} stores the number of times topic t is assigned to any word in document d also discarding the current word topic assignment. With the two matrices in place Griffiths and Steyvers present the following approximation for the conditional posterior word-topic assignment:

$$p(z_n = t|z_{-n}, w_n, d_n, \cdot) \sim \frac{C^{VT}_{w_n t} + \beta}{\sum_{i=1}^{V} C^{WT}_{w_i t} + W\beta} \frac{C^{DT}_{d_n t} + \alpha}{\sum_{j=1}^{T} C^{DT}_{d_n t_i} + T\alpha} \tag{2.20}$$

Where d_n is the document that the current word w_n belongs to. The symbol "$\sim$" in the above equation tells us that the conditional probability estimate is not normalized. In order to convert it to a true probability value within the [0,1] range, the above equation needs to be divided using the marginal probability over the topics.

Using the above conditional estimates of the topic assignments it is straightforward to approximate the document-topic $\hat{\theta}$ and the topic-word $\hat{\varphi}$ distributions:

$$\hat{\theta}_{dt} = \frac{C^{DT}_{dt} + \alpha}{\sum_{j=1}^{T} C^{DT}_{dt_j} + T\alpha} \tag{2.21}$$

$$\hat{\varphi}_{tw} = \frac{C^{WT}_{wt} + \beta}{\sum_{i=1}^{W} C^{WT}_{w_i t} + W\beta} \tag{2.22}$$

As in the case with a typical MCMC approximation, in the Gibbs sampler the first set of samples are discarded since they don't provide good estimates of the posterior. This is referred to as the "burn-in" period. It is typically the case that at least the first 1,000 samples are discarded. It is also the case that $\hat{\theta}$ and $\hat{\varphi}$ are updated at a certain sample interval (e.g. every 10, 30 or 100 samples).

2.2.5.2 Variational Bayes

The VB approach [59] defines the problem of approximating the posterior distributions as an optimization task. VB uses a family of probability distributions with variational parameters that simplifies the complex dependence of the latent variables θ, z and φ in the LDA model. Their original dependence is broken down into dependencies of the individual model variables over the variational distributions γ, ϕ and λ respectively. Figure 2.9 shows

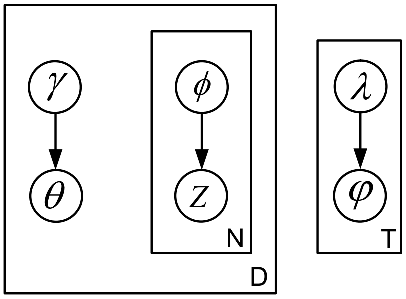


Figure 2.9: Graphical model representation of the free variational parameters for the VB approximation of the LDA posterior.

the graphical representation of the variational distributions used for approximating the LDA posteriors. The approximation takes on the following functional form:

$$q(\theta, z, \varphi|\gamma, \phi, \lambda) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \prod_{t=1}^{T} q(\varphi_t|\lambda_t) \tag{2.23}$$

The variational parameters approximate the true posterior through an optimization process whose goal is to maximize the evidence lower bound (ELBO) on the marginal log likelihood probability $logp(w|\alpha, \beta)$. This lower bound is derived using the Jensen inequality:

$$log\, p(w|\alpha, \beta) \geq E_q \left[log\, p(\theta, z, w, \varphi|\alpha, \beta)\right] - E_q \left[log\, q(\theta, z, \varphi)\right] \tag{2.24}$$

Where the inequality, i.e the difference between the two sides of the above equation, is due to the Kullback-Leibler (KL) divergence between the true posterior and variational posterior probability. Maximizing the lower bound translates to minimizing the KL divergence which is the optimization objective:

$$
\begin{aligned}
log\, p(w|\alpha,\beta) = E_q\left[log\, p(\theta,z,w,\varphi|\alpha,\beta)\right] - E_q\left[log\, q(\theta,z,\varphi)\right] \\
+KL(q(\theta_d,z,\varphi|\gamma,\phi,\lambda)\,||\,p(\theta_d,z,\varphi|w,\alpha,\beta))
\end{aligned} \tag{2.25}
$$

Minimizing the KL divergence is performed by setting the derivatives of the variational parameters to zero. From an algorithmic perspective, VB follows the Expectation-Maximization (EM) procedure where for a given document, the E-step updates the per document variational parameters γ_d and ϕ_d while holding the words-topic distribution parameter λ fixed. It then updates the variational parameter λ using the sufficient statistics computed in the E-step. For each of the variational parameters we have the following per document updates in the E and M-steps of the algorithm:

$$\phi^d_{wt} \propto \exp\left\{E_q\left[\log\theta_{dt}\right] + E_q\left[\log\varphi_{tw}\right]\right\} \tag{2.26}$$

$$\gamma_{dt} = \alpha + \sum_{w=1}^{W} \phi^d_{wt}\, n^d_w \tag{2.27}$$

$$\lambda_{tw} = \beta + \sum_{d=1}^{D} n^d_w \phi^d_{wt} \tag{2.28}$$

The expectation values of the true posterior distributions θ and φ given the optimized values of the variational parameters γ and λ are derived using the exponential form of the two distributions:

$$E_q[\log(\theta_{dt}|\gamma_{dt})] = \Psi(\gamma_{dt}) - \Psi(\sum_{t=1}^{T}\gamma_{dt}) \tag{2.29}$$

$$E_q[\log(\varphi_{tw}|\lambda_{tw})] = \Psi(\lambda_{tw}) - \Psi(\sum_{w=1}^{W}\lambda_{tw}) \tag{2.30}$$

Where $\Psi(x)$ is the digamma function and it is the first derivative of the logarithm of the Gamma function $\Gamma(x)$.

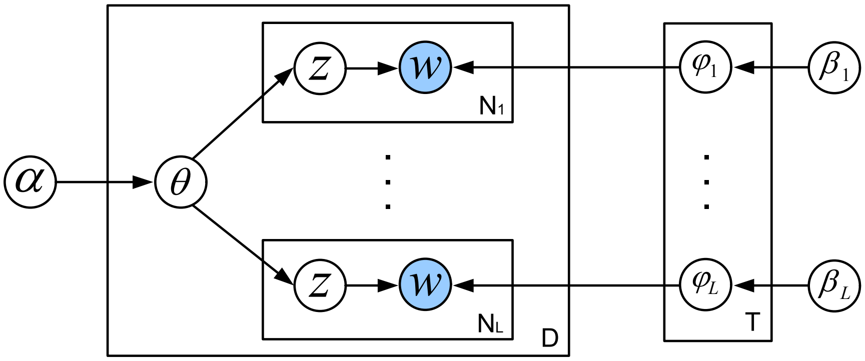


Figure 2.10: Graphical representation of the polylingual topic model (PLTM).

2.2.6 Polylingual Topic Model (PLTM)

As we pointed out at the beginning of this section, one of the greatest advantages of using topic models is their ability to represent documents in a low-dimensional shared space that abstracts away from the actual words used in the documents. This representation facilitates analysis of documents written using different vocabularies. For example representing scientific article in the domain of physics and computer science using topic models may discover that the two articles share the same set of topics despite the vocabulary mismatch. The vocabulary mismatch is more emphasized when documents are written in a different language. For example, documents that are translations of each other while written in a different language most certainly share the same set of topics. Detecting document pairs that are translations of each other could be easily facilitated if the documents are first represented in a language independent topic space. This could be achieved using PLTM [85].

Given a collection of document tuples d such that each tuple consists of one or many documents that are topically similar in different languages $l = 1, 2, ..., L$, d = (doc_1, doc_2, doc_3, ..., doc_L), PLTM assumes that documents in the tuple, while written in a different observed language L, cover the same set of topics θ_d. The graphical model representation of PLTM is shown in Figure 2.10.

The model also assumes that each language has its own language specific topic-word distribution φ^l over the words in the language vocabulary V_l. The generative process of

PLTM is similar to LDA with only difference that words are drawn from a language specific topic distributions φ^l: $w^l \sim p(w^l|z^l, \varphi^l)$ and we are now dealing with tuple-topic distribution θ_d which is used to draw the topic assignments over words: $z^l \sim p(z^l|\theta_d)$. Aside from the benefits that it offers on the task of detecting document translation pairs, PLTM, as we will demonstrate in Chapter 6, provides a means for extracting parallel sentences from comparable corpora.

2.2.7 Inference in PLTM

2.2.7.1 Gibbs Sampling

Incorporating the polylingual concept in the Gibbs sampling inference approach is very straightforward. Approximating the tuple-topic distribution $\hat{\theta}_d$ is performed using the counts of the number of times topic t was assigned in all documents of tuple d: C^{DT}_{dt}. Across the different languages in the collection we approximate the topic-word distributions using language specific matrices of counts $C^{W_lT}_{v_lt}$. The estimates of the two types of distributions are shown below:

$$\hat{\theta}_{dt} = \frac{C^{DT}_{dt} + \alpha}{\sum_{j=1}^{T} C^{DT}_{dt_j} + T\alpha} \tag{2.31}$$

$$\hat{\varphi}^l_{tw} = \frac{C^{W^lT}_{w^lt} + \beta^l}{\sum_{i=1}^{W_l} C^{W^lT}_{w^l_i t} + W^l\beta^l} \tag{2.32}$$

2.2.7.2 Variational Bayes

In VB inference the update steps for the variational parameters take into account the language l count statistics as well as the topic-word distributions:

$$\gamma_{dt} = \alpha + \sum_{l=1}^{L} \sum_{w=1}^{W^l} \phi_{wt}^{dl} \, n_w^{dl} \tag{2.33}$$

$$\phi_{wt}^{dl} \propto \exp\left\{ E_q \left[\log \theta_{dt}\right] + E_q \left[\log \varphi_{tw}^{dl}\right] \right\} \tag{2.34}$$

$$\lambda_{tw}^{l} = \beta^{l} + \sum_{d=1}^{D} n_w^{dl} \phi_{wt}^{dl} \tag{2.35}$$

2.2.8 Inferring Topics in Document Collections using LDA and PLTM

Topic models are unsupervised generative models of text. Unlike supervised models, where parameters are first trained on an annotated collection, in unsupervised models there is no clear boundary in defining the training and test steps. It is often the case that the models' topic-word distributions and the Dirichlet hyperparameters are inferred by first running the model on a training set of documents. These parameters are then used to infer document-topic distributions on a separate set of unseen documents which are considered as the test set.

Document collections require certain processing prior to running topic models. The most important preprocessing step involves deciding on the vocabulary that will be used by the model. From the words present in the collection, the model vocabulary is constructed by removing stop words which are usually generic and predefined for each language. For the given collection the top most frequent words (e.g. top 25, 50, 100, etc.) are considered as collection specific stop words and are also removed. In addition, words whose collection wide frequency, i.e. term frequency (tf), is very low (e.g. hapax words) are also removed.

To help streamline the inference process documents are often represented using only the words from the models' vocabulary along with their document specific tf. Words are also represented as integer numbers so that the document representation becomes a sequence of tuples (word, document tf). Since topic models treat documents as bag of words, any sequence information on the words occurrence in the document is discarded in this process.

CHAPTER 3

EFFICIENT NEAREST-NEIGHBOR SEARCH IN THE PROBABILITY SIMPLEX

3.1 Introduction

Document similarity comparisons are part of many tasks in information retrieval (IR) and natural language processing (NLP). In addition, other fields in computer science also utilize similarity comparisons. For example in computer vision similarity comparisons are usually performed over discrete vector representations of images which act as approximations of their original continuous representations. Document similarity comparisons tasks include clustering of documents, retrieving similar documents, finding document translation pairs in a large multilingual collection, to name a few. Typical and well established solutions to these tasks involve two steps:

- Representation: Documents are represented in a shared feature space. While pairwise similarity functions could be directly computed over the original representation of documents, representing them in a shared feature space, amongst many other benefits, allows for efficient comparisons. Depending on the task, a decision needs to be made on the appropriate space and most suitable document representation. While there are various choices for what features to use to represent documents, when deciding on the shared space i.e. the feature space[1], the choice is two fold: documents could be represented as features in the (1) metric space or the (2) probability simplex.

[1] Throughout this chapter we will be using the terms “shared” and “feature” space interchangeably.

- Comparison: Once documents are represented in a shared space a comparison is performed. Depending on the task the comparison could be all-pairs (e.g. document clustering) or could be limited to comparing one or k documents in a given query with a whole collection of N documents.

In this chapter we focus our attention on document representations in the probability simplex, such as the representation offered by topic models. We first give an overview of the potential and benefits of using topic models to represent documents in a shared space. We then highlight the issues of performing efficient similarity comparison across topic distributions in large collections. Unlike the metric space where similarity between vector representations is computed using distance metrics, such as Euclidean (Eu) or Cosine (Cos), in the probability simplex similarity between distributions is computed using measures of probability divergence. More specifically, information-theoretic measurements such as: Kullback-Liebler (KL) or its symmetric form Jensen-Shannon (JS) divergence and Hellinger (He) distance. In the metric space, viewing the problem as an approximate nearest-neighbor (NN) search problem offers solution for performing efficient similarity search. Unfortunately such solutions don't exist for the topic space where distributions are continuous. As a consequence, questions whether representations of documents in the topic space offer advantages over existing representations, especially in large collections, have not been exploited. In this chapter we present novel approaches that allow us to perform efficient similarity search in the probability simplex. We achieve this by performing novel analysis of the reduction of He to Eu distance computations. Furthermore we analyze the constant factor relationship between He and JS which allows us to approximate JS with He. With these methods we are able to utilize fast NN search techniques over information-theoretic measurements and therefore perform efficient similarity comparisons in the probability simplex. We showcase the effectiveness and efficiency of these methods by utilizing two widely used fast approximate NN techniques – locality-sensitive hashing (LSH) and approximate search in k-dimensional (k-d) trees on four different tasks with two families of topic models: (1) using latent Dirichlet allocation (LDA) to cluster scientific articles; (2) using LDA to aid in prior-art retrieval for patents; (3) applying approximate NN computations in topic space to the task of speeding up a relevance model; and (4) using the polylingual topic model (PLTM) to find document translation pairs in a large bilingual corpus. Empirical evaluations on these tasks show that these methods are as accurate as and significantly faster than brute-force all-pairs search. While both example approximate NN search techniques perform well when the task searches for the single NN, as in the translation detection task, when, as in the patent task, a larger set of similar documents is to be retrieved, the kd-tree approach is more effective and efficient.

3.2 Document Representation

Representing documents in a shared space is a natural first step in solving the task of finding similar documents. More specifically, for the purpose of obtaining similarity scores across the collection, all documents must be represented in a document independent, shared space. This type of representation abstracts beyond the specific sequence of words used in each document. For example, the "bag of words" representation provides a shared space that ignores the order of the words in the document. This type of representation can also facilitate the analysis of relationships between documents written using different vocabularies. For instance, comparing documents written in a highly domain-specific language, such as patents and scientific papers, can be difficult due to the vocabulary differences between the domains. As a concrete example, determining the novelty of a patent application involves comparing the invention and idea presented by the applicant with previously granted patents and scientific and technical publications. This task poses significant challenges due to the esoteric, and even obfuscated, use of language in patent applications. Similarly, identifying academic communities working on related scientific topics can involve comparing the divergent terminologies in different subfields [115]. A more extreme form of vocabulary-mismatch occurs when documents are written in different languages.

Mapping documents in different languages into a common shared space can therefore be an effective method of finding documents or passages that are translations of each other [85, 97]. Feature space selection is therefore a crucial first step in solving the document similarity task.

Across many IR tasks documents are represented in the metric space with features that are based on the actual words and are either constructed using single words or n-gram combinations of them. In most cases feature values are based on document specific word or n-gram counts or collection wide statistics such as term frequency (tf) or inverse document frequency (idf). An exemplar of this type of representation is the vector space model that represents documents in the metric space. Shared spaces constructed with word based features are high dimensional – large document collections contains thousands of words and n-grams. At the same time such shared spaces are sparse. Distribution of words in the collection follows the Zipf's law [98] which predicts that exactly half of the words in a collection will be singletons. More generally, it predicts that words with frequency n will have $\frac{1}{n(n+1)}$ predicted proportion of occurrences. These characteristics of the vector space model are explored by the inverted index representation of the document collection which provides efficient architecture for representing and performing similarity comparisons across collections of different lengths.

Since the first introduction of the vector space model for performing document similarity comparisons, variations that offer more semantic representation have been proposed such as the Latent Semantic Indexing (LSI) [33]. With the introduction of statistical modeling techniques LSI has been recast as a generative model of text with probabilistic latent semantic indexing (PLSI). In more recent years, latent variable models of text with prior structure over latent variables are given more attention as they alleviate the constrains found in the PLSI model. More specifically, statistical topic models, such as LDA, have proven to be highly effective at discovering hidden structure in document collections (e.g. [46]).

Unlike the vector space model and other models that use word based representation of documents, topic models represent documents as probability distributions over a small number of topics which are themselves distributions over words, just as in (Dirichlet smoothed) unigram language models. On the other hand, unlike the vector space features which are sparse and discrete, topic distributions are continuous. Their ability to map documents into a low-dimensional probability simplex allows for the reduction of the dimensionality of the representation of documents while discarding noise and retaining salient information. In the past feasibility and effectiveness of topic models have been explored by the IR community. For example, Wei & Croft [128] inferred topics over words using LDA to improve document smoothing and ad-hoc retrieval. In a subsequent study, Yi & Allan [131] discovered that it is hard for low-dimensional models such as LDA to significantly help ad-hoc retrieval tasks with short queries and tasks where precision is highly valued. In this work we give attention to recall oriented tasks but as we will show in § 3.7.3 LDA could also be used to create faster relevance models. More recently, Andrzejewski & Buttler [7] showed that LDA has the potential to improve on the task of query expansion for specialized domain collections with a small user base.

3.3 Measuring Similarity

Within the shared space, document similarity is interpreted based on a particular metric. In addition to Eu and Cos in the metric space, distance metrics such as Manhattan, Jaccard, Product and Dice are used as well. Table 3.1 shows the algebraic expressions of these metrics along with their range of values.

When documents are mapped into the probability simplex, their similarity comparison is performed by measuring the difference between two probability distributions. As such, distance metrics are not appropriate in the probability simplex and information-theoretic measurements are used instead: KL and JS divergence and He distance. JS divergence and He distance are f-divergences [31] as they both measure the difference between two prob-

Similarity Metric	Algebraic Expression	Min. Value	Max. Value
Manhattan (L_1 norm)	$\sum_{i=1}^{n} \lvert p_i - q_i \rvert$	0	∞
Euclidean (L_2 norm)	$\sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$	0	∞
General L_k metric	$\sqrt[k]{\sum_{i=1}^{n} (p_i - q_i)^k}$	0	∞
Cosine	$\frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2 \sum_{i=1}^{n} q_i^2}}$	0	1
Jaccard	$\frac{\lvert p \cap q \rvert}{\lvert p \cup q}$	0	1
Product	$\sum_{i=1}^{n} p_i q_i$	0	∞
Dice	$\frac{2\lvert p \cap q \rvert}{\lvert p \rvert + \lvert q \rvert}$	0	∞

Table 3.1: Common similarity measurements used in the metric space.

ability distributions. JS divergence was originally derived from KL divergence as its symmetric version [75, 100]. KL divergence is also know as relative entropy. Query likelihood IR models use KL divergence, which, being asymmetric, is less suitable for ranking document pairs. He distance is also symmetric and is used along with JS divergence in various fields where a comparison between two probability distributions is required [15, 18, 46].

Unlike distance metrics, these measurements don't satisfy the triangle inequality and therefore could not be used to measure distances. On the other hand distance metrics are not well behaved for probability distributions. Table 3.2 shows the algebraic expressions of the three most frequently used similarity measurements in the probability simplex along with their range of values.

KL divergence could have a value of infinity [27] while JS divergence is bounded by one [75]. Throughout the scientific literature He distance could be found in various algebraic variations where two dominant variations differ in the constant of $\frac{1}{2}$ placed in front of the sum as well as the square root of the whole equation. He distance is bounded by 1, 2, or $\sqrt{2}$ depending on the version of the formula.

Similarity Metric	Algebraic Expression	Min. Value	Max. Value
Kullback-Liebler (KL)	$\sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)}$	0	∞
Jensen-Shannon (JS)	$\frac{1}{2}\text{KL}\left(p, \frac{p+q}{2}\right) + \frac{1}{2}\text{KL}\left(q, \frac{p+q}{2}\right)$	0	1
Hellinger (He)	$\sum_{i=1}^{n}\left(\sqrt{p(x_i)} - \sqrt{q(x_i)}\right)^2$	0	2

Table 3.2: Common similarity measurements used in the probability simplex.

3.4 Document Similarity in Large Collections

Time complexity of the comparison, regardless of the type, grows as the collection size grows. For large collections computing similarity practically becomes infeasible. Depending on the task, document similarity could be performed across all documents in the collection (i.e. all-pairs comparison) or the l most similar documents. For example, in a collection of N documents, exact similarity comparison for clustering tasks involves computing similarity across all pairs which requires $O(N^2)$ comparisons. As another example, given a set of k query documents which are compared against a data set of N documents requires $O(kN)$ computations.

For document similarity tasks to be performed efficiently on large collections the time complexity ought to be reduced. Framing the document similarity computation as an approximate NN search problem allows for approximate computation methods to be introduced which reduce the time complexity. NN search is an optimization problem that deals with the task of finding nearest neighbors of a given query q in a metric space of N points. Once the similarity task is formulated as a NN search problem, we are able to use different data structures and approximate algorithms that deal with this problem in an optimal way while trading off speed and accuracy.

In the past, this type of formulation for the document similarity comparison problem has been proven to yield good results in the metric space due to the fact that NN search problem has been designed to handle the most common distance metrics (e.g. Eu, Cos, Manhattan, etc.) and therefore could be applied directly.

Compared to the inverted index approach, approximate NN search approaches are more advantageous when dealing with dense representations, such as low-dimensional topic representations. On the other hand when dealing with high-dimensional shared spaces, such as the ones constructed using word based features, which are also sparse, approximate NN techniques may fail and the inverted index approach is more suitable due to the fact that it utilizes the sparse nature of the representation.

The LSH approach approximates the NN search by hashing data points into “buckets” such that data points with close proximity fit into the same bucket with high probability. For example, LSH using Cos distance have been previously used for various NLP tasks such as noun clustering [101], first-story detection on Twitter [96], spoken term discovery in speech data [56], and ranking document pairs by overlapping words [67]. Another approach that provides approximate NN search for the most common distance metrics, which we will be using in this chapter, is k-d trees [11]. K-d trees partition and organize the K multidimensional space of data points using binary trees. In the past both approximate NN search techniques have found there use across different IR tasks. For example, LSH techniques have been used on tasks such as near-duplicate and sub-image retrieval [61], content-based music retrieval [130], cross-lingual pairwise similarity [119], to name a few. And K-d trees have been used on tasks such as video retrieval using human poses as queries [55], retrieving protein molecules with similar structures [82], performing ad-hoc search in sensor networks [3], to name a few.

While LSH schemes exist for both Cos and Eu distances [5, 25] and k-d trees and their variants work with the Minkowski metrics (L_1, L_2, etc.) i.e. general L_k norm [9, 38], these cannot be directly applied to measuring distances in the probability simplex. Therefore, performing document similarity in large datasets where documents are represented as points in the simplex cannot be addressed with the same NN search algorithmic instances as described in the previous paragraph. The inability to perform fast document similarity computations when documents are represented in the simplex has thus limited researchers' ability to explore the potential of simplex representations of documents on large scales.

In this chapter we introduce techniques for performing NN search that facilitate efficient document similarity computations across document representations in the probability simplex. In the next section, we show that He distance can be framed as Eu distance under a monotonic transformation of the probability space. We can thus quickly find approximate NN using He distance by inputting the transformed probability distributions into Eu LSH and k-d trees. Furthermore, we empirically show that JS divergence is bound by He distance thus enabling us to approximate JS divergence using He distance. With the formulation of the document similarity task as an NN search problem and utilizing approximate NN search techniques we are able to trade off time complexity and accuracy. We perform speed and accuracy analysis of our approaches on four different tasks that utilize document similarity: scientific community discovery, related patent retrieval, pseudo-relevance feedback for ad-hoc retrieval and finding document translation pairs. It is important to note that while the effective NN search techniques are showcased on document similarity tasks these techniques generalize well across different tasks that involve comparisons of probability distributions in large collections. For example, these techniques could be useful for comparing dense word and document embeddings in non-probability spaces such as the ones generated from deep belief network (DBN) models.

3.5 Fast NN Search

While there are many approximate NN search techniques, in this work we utilize two widely used approaches: LSH [53] and the multidimensional binary search tree or k-d tree concept introduced by Bentley [11]. More specifically we use the Exact Euclidean LSH (E2LSH) approach developed by Andoni et al. [5] which is implemented in the E2LSH package [6], and the k-d tree implementation in the ANN library [88]. Both of these approaches were originally introduced in the metric space.

3.5.1 Locality Sensitive Hashing (LSH)

LSH is a sub-linear solutions for retrieving the nearest neighbors of a query point. In their work Indyk and Motwani [53] defined the problem as follows. Given a query point q, return point p that is an ϵ-approximate NN of q such that $\forall p'$ nearest neighbors that satisfy the inequality: $Distance(q, p) \leq (1 + \epsilon) Distance(q, p')$ or more succinctly $(1 + \epsilon)r$ nearest neighbors, where r is the radius of data set points considered for each query point. In general this approach uses a hashing scheme where the original data points are hashed into separate buckets using a family of hash functions such that the probability of collision between query point q and points p in the dataset increases with the similarity between them. More formally, the function $p(t) = \Pr[h(q) = h(p) :\| q - p \|= t]$ is strictly decreasing in t. Once points are hashed into a bucket, the closest point(s) out of those already in the bucket is returned. Utilizing multiple hash functions improves the accuracy. Varying the radius r changes the number of data set points considered for each query point and therefore directly affects the accuracy of the results as well as the running time of the algorithm. Charikar et al. [25] expanded this approach and showed that it could be used to perform approximate Cos distance computation.

3.5.2 K-d Trees

Another widely used sub-linear solution for performing fast NN search in the metric space is k-d trees. This method utilizes coordinate information to come up with an optimized data structure that organize points in space by partitioning. With k-d trees points in metric space are stored in a partitioning data structure where data points are represented with nodes along with two pointers and a discriminator variable whose range of values is the dimensionality of the space. The two data pointers point to subtrees or take on the null value based on whether the value of the chosen point dimension (based on the discriminator value) is greater or smaller then the split value for that dimension. Unlike LSH, k-d trees don't give probabilistic guarantees but at the same time their efficient data structure is very practical to use. As in the case with LSH, the runtime of k-d trees depends on the number of pairs within distance r which is independent of the number of data points N. In general, the efficiency of k-d trees is heavily affected by the ratio between N and the dimensionality of the space d. In instances when these two number are close the performance is only slightly better than linear search because of the number of nodes in the tree that the algorithm needs to consider. In the next section we describe how common probability divergence measurements could be used with these sub-linear NN search approaches.

3.5.3 Transforming Divergences

Observing the algebraic expressions of the Eu distance in Table 3.1 and He distance in Table 3.2 one could immediately notice a close similarity between the two measurements which differs in how the square root function is applied. He distance could be transformed into the Eu distance if the square root of the Eu distance is applied to the individual distributions $\sqrt{p_i}$ and $\sqrt{q_i}$. Since the NN search problem involves ranking points based on their distance to the query point, this transformation doesn't affect the overall ranking as it is performed across all data points.

The above transformation allows us to compute He using Eu distance by first computing the square root of the distributions that we would like to compare. With approximate NN search techniques for Eu distance we are now able to efficiently perform NN search in the probability simplex.

The algebraic expression of the JS divergence doesn't provide similar ground for performing a simple transformations as in the case of the He distance. However, in the past this divergence has been theoretically shown that could be bounded by the He distance. More specifically, Topsøe [118] has shown that JS divergence (referred in his work as capacitory discrimination), behaves similarly with the triangle divergence (triangular discrimination):

$$\frac{1}{2} \triangle (p,q) \ \le JS(p,q) \ \le \ \ln(2) \triangle (p,q) \tag{3.1}$$

In his work, Topsøe [118] also outlines the close relationship between He distance and triangle divergence:

$$He(p,q) \ \le \ \triangle(p,q) \ \le \ 2He(p,q) \tag{3.2}$$

Without rehearsing the proofs combining these two expressions (as in the case with [45]), we observe that JS divergence is bound by He distance:

$$\frac{1}{2}He(p,q) \ \le \ \frac{1}{2} \triangle (p,q) \ \le \ JS(p,q) \ \le \ \ln(2) \triangle (p,q) \ \le \ 2\ln(2)He(p,q) \tag{3.3}$$

More specifically, there is a constant factor relationship between these two measurements which allows for the approximation of JS divergence with He distance:

$$\frac{1}{2}He(p,q) \ \le \ JS(p,q) \ \le \ 2\ln(2)He(p,q) \tag{3.4}$$

In the above analysis we showed that with approximate approaches for computing Eu distance we could compute He distance. Furthermore, by exploring ways to approximate JS with He we could achieve fast document similarity comparison with JS divergence.

3.6 Experiments with Synthetic Data

In order to evaluate the theoretical bounds across probability distributions of various dimensions and different sparsity, an experimental setup was constructed where we computed all-pairs similarity using JS and He measurements. We used a dataset which consisted of 100 multinomial distributions drawn from families of Dirichlet distributions with different dimensions D and hyperparameters α where each dataset consists of 10k values from both measurements. As it was discussed in Chapter 2, varying the hyperparameter α generates distributions with different sparsity where the lower the parameter the higher the sparsity of the generated distributions. More specifically we used five dimensions D =50, 100, 200, 500 and 1000 and six hyperparameter values α= 0.001, 0.01, 0.1, 1, 10 and 100 to generate 24 sets of 100 multinomial distributions. Figure 3.1 shows the relationship between JS divergence and He distance across dimensions D = 50, 100, 500 and 1000 and the six hyperparameter values. Lines across all plots represent the bounds in Equation 3.4. The lower bound is $He(p,q) = \frac{1}{2\ln(2)} JS(p,q)$ while the upper bound is $He(p,q) = 2JS(p,q)$.

While the theoretical bounds hold, in reality they are even tighter. This provides ample evidence that JS divergence could be approximated by He distance. There are several observations that are worth noting from these plots. Across all dimensions, the range of values for both measurements reduces as the sparsity decreases and when drawn multinomials are close to uniform ($\alpha = 100.0$), the range is the smallest. For very sparse families of distributions ($\alpha \leq 0.1$) the bulk of the mass of measurement values resides close to the upper bound. As we increase the dimensions of the distributions, this mass shifts towards the upper limit of both measurements.

3.6.1 Approximate vs. Regular Computation

Across the 24 datasets we setup IR evaluation experiments where we compared the performance of the E2LSH and k-d trees NN methods on the task of retrieving the top ten nearest points obtained using all-pairs JS divergence. Figures 3.2 and 3.3 show the performance comparison results. In Figure 3.2 we show comparison results over hyperparameter range of values α=0.001, 0.01 and 0.1 which generate sparse multinomial distributions. While in Figure 3.3 results are shown over more uniform distributions α=1, 10 and 100.

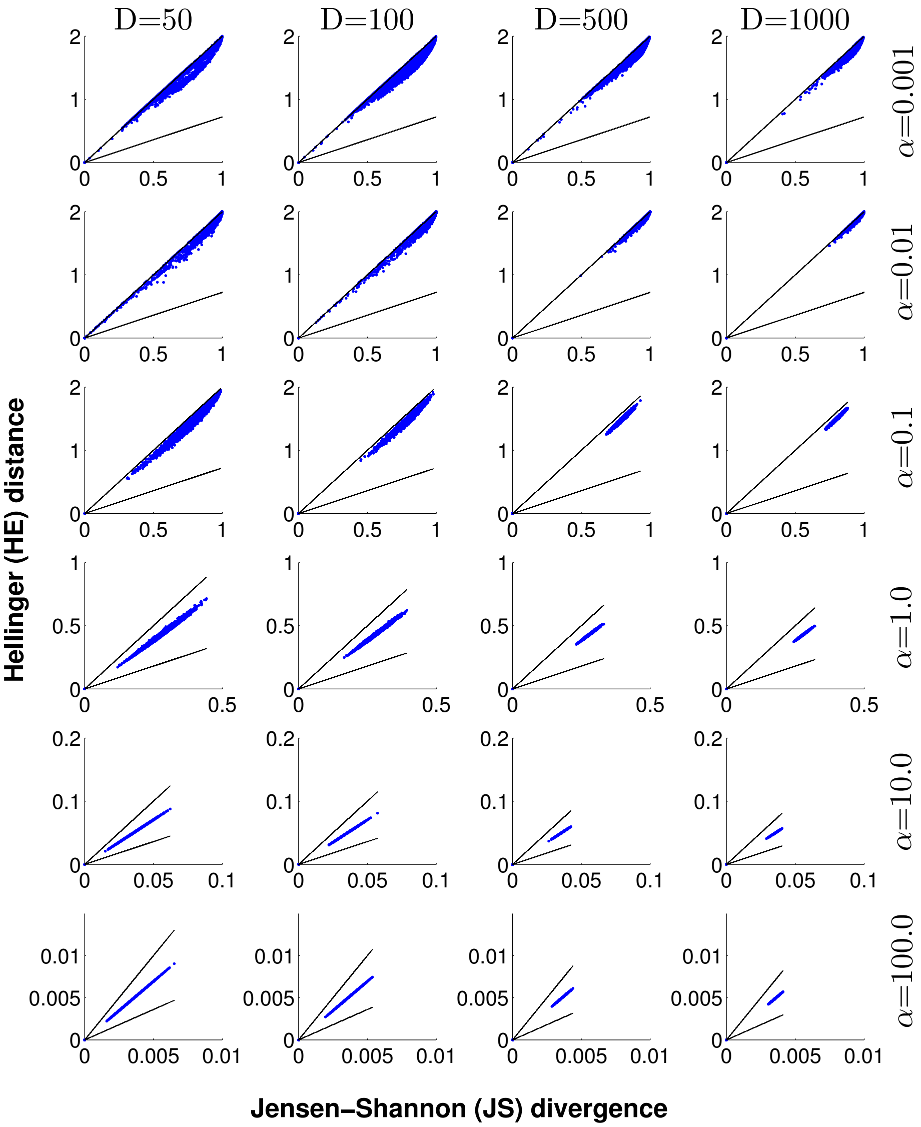


Figure 3.1: Empirical evidence of the relationship between JS divergence and He distance across probability distributions of different lengths and sparsity.

Both approximate methods were configured with their default parameter and E2LSH was set with the default radius value of $R = 0.6$. While across all datasets k-d trees outperforms E2LSH it is worth noting that in most instances both methods are able to retrieve the NN (which is at rank two) with high precision. The performance of E2LSH reduces with the increase of the sparsity while both methods give better performance on low dimensional distributions.

3.7 Efficient NN Search in the Probability Simplex

The goal of developing approximate NN search approaches for the probability simplex was to be able to utilize the performance of latent variable models of text in large mono- and multilingual collections. To demonstrate the effectiveness and efficiency of these approaches we used four different tasks which span in three different domains: document clustering, document retrieval and MT and use two different evaluation measures. All tasks involve document similarity by representing documents in a shared topic space and as such could be recast as NN search problems. In addition they offer a range of objectives for the NN search problem from 1-NN to k-NN. They include: (1) clustering scientific articles from the National Institutes of Health (NIH); (2) retrieving prior-art patents by representing patent applications as topic distributions in the probability simplex (in this task vocabulary mismatch may be part of the authors' strategy); (3) applying approximate NN computations in topic space to the task of speeding up a relevance model; (4) finally, we'll utilize our approximate NN search techniques to find document translation pairs in large collections in an efficient way without significant degradation in results.

Across all tasks we'll be comparing the performance of the approximate He distance computed through LSH and k-d trees with regular He distance. We'll also compare the approximation of JS divergence with He distance. With these comparisons we'll perform analysis of the performance loss and speed gain when using approximate approaches for computing similarity in the probability simplex.

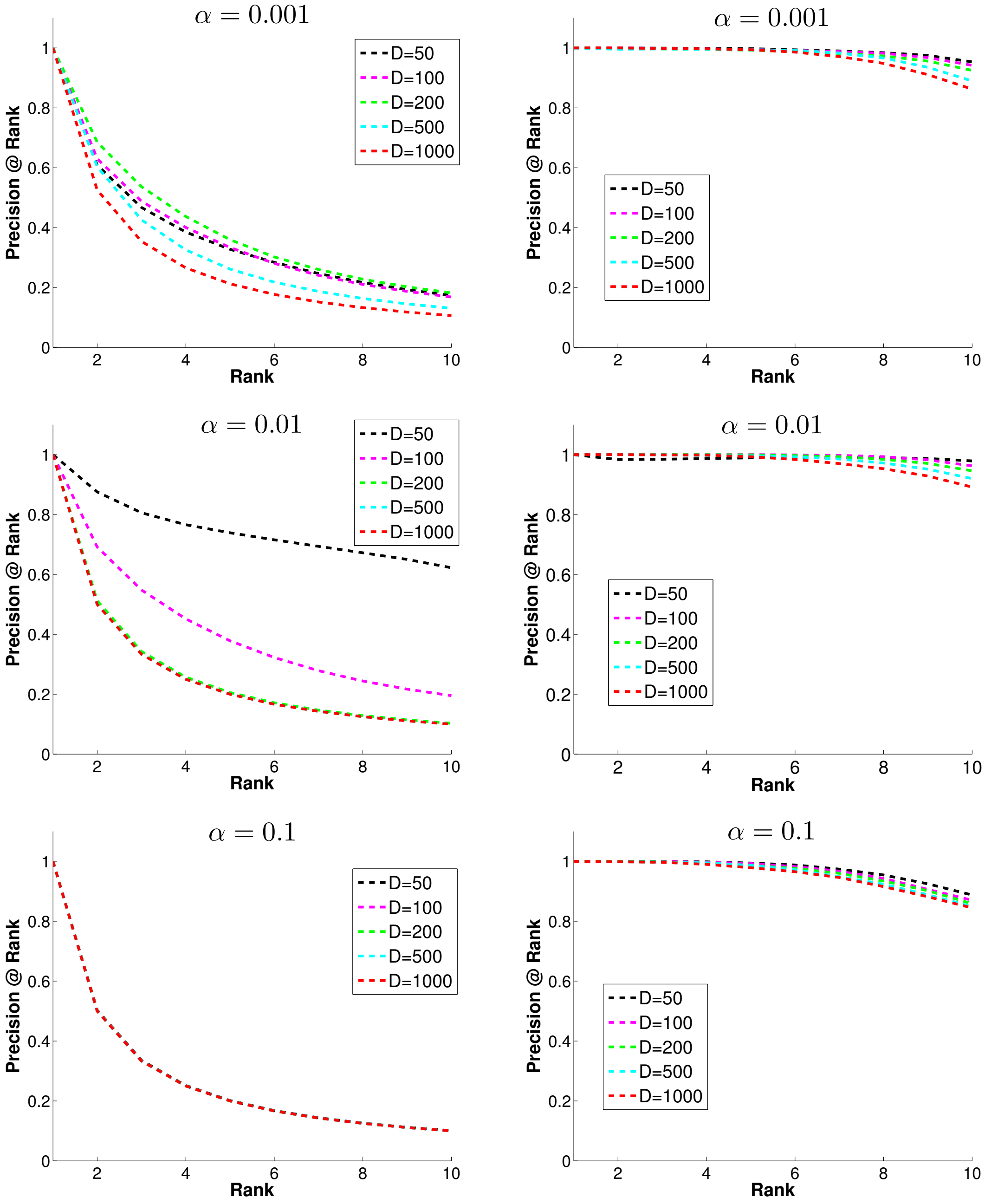


Figure 3.2: Comparison of approximate NN search techniques across families of distributions with different dimensions (D=50, 100, 200, 500 and 1000) and sparsity (α=0.001, 0.01 and 0.1).

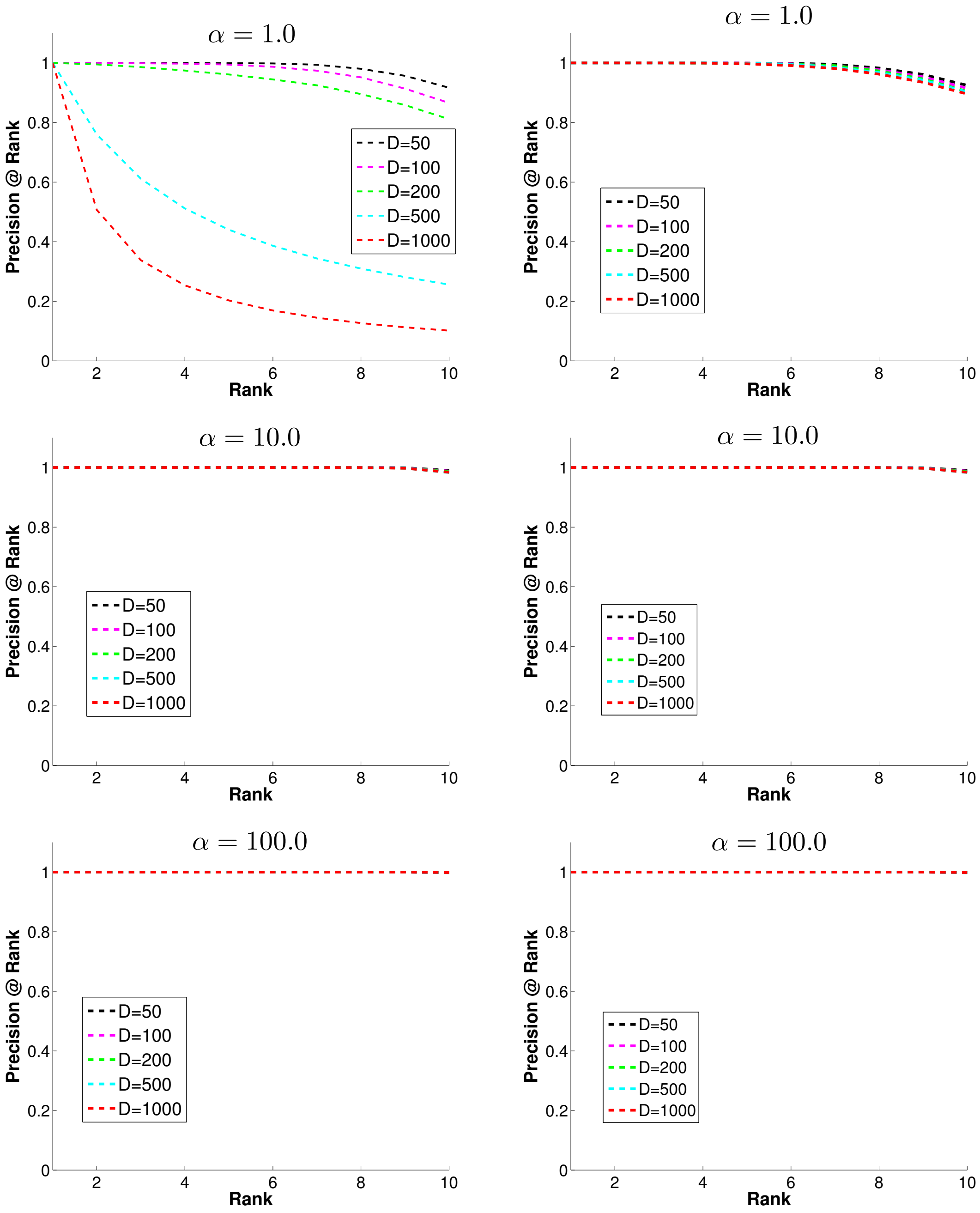


Figure 3.3: Comparison of approximate NN search techniques across families of distributions with different dimensions (D=50, 100, 200, 500 and 1000) and sparsity (α=1, 10 and 100).

While various latent variable models could be used to represent documents in the shared probability simplex in these experimental setups we focus on topic models and perform relative comparisons of models performance opposed to utilizing different families of latent variables or even comparing performance with models that represent documents in the metric space. An exception to this experimental setup will be made on the prior-art patent search task where we'll showcase how retrieved patents represented in the topic space could be combined with results from a vector space model and improve the precision of the retrieving patents.

3.7.1 Mapping NIH Funding

In an earlier work by Talley et al. [115] authors showcased that using LDA to map documents in a shared topic space facilitates the analysis of relationships across issued grants from NIH. This in turn helps discover funding grants that cover similar or same set of topics across various institutes and centers that are part of NIH. More specifically, in [115] authors constructed a graph-based layout of grants where grants were arranged based on a weighted sum of KL divergence computed on word probability distributions and topic distributions. Their goal was to provide tools for the centers and institutes across NIH to get an insight on how their funding grants are related. As these centers and institutes grow in their missions and increase the number of awarded grants it is more likely that their grants will overlap with each others goals. Therefore having such tools is very beneficial for future planning. In our case with this dataset we are trying to answer the question whether using latent variable model representations of documents could be efficiently used to cluster large document collections. We use this task to compare the performance of the approximate methods with the brute-force all-pairs comparison approach for computing He and JS on the task of clustering documents in a large collection. Similar to the experiments performed on the synthetic datasets, in this task we use LSH and k-d trees to retrieve the top ten relevant documents discovered using regular JS and He measurements.

3.7.1.1 Experimental Setup

In their work, Talley et al. [115] used LDA to represent a collection of $\sim$350k documents which contains abstracts and titles of grants from 2007–2010, MEDLINE journal articles published between 2007–2010 that cite NIH grants and intramural and sub-awards into a shared topic space. We used the same topic representations which were generated using Mallet's [83] implementation of LDA configured with 550 topics. The Mallet implementation of LDA utilizes Gibbs sampling to approximate the posterior document-topic and topic-word distributions which requires multiple iterations over the whole collection.

In our experimental setup we first computed regular JS divergence across all pairs of $\sim$350k documents and used the top ten most similar documents for each grant as a set of relevant grants. We then created a query set of 10k randomly chosen grants and use the LSH and k-d trees implementations to retrieve the top ten grants from the remaining database of $\sim$343k grants.The ANN implementation of k-d trees was configured with default parameters while for the E2LSH implementation of LSH we varied the radius R to values of R=0.4, 0.6 and 0.8 .

3.7.1.2 Evaluation Task and Results

Since the relevance sets are ten documents long we evaluated the performance of the approximate similarity methods by computing precision of the top 5 (P@5), recall of the top 5 (R@5), and mean average precision (MAP) over each query result. In Table 3.3 we show the results obtained along with the relative difference in time between all-pairs JS divergence (relative speed of one), the approximate LSH based He distance with different value of R and the approximate k-d trees based He distance. The code implementation of the similarity computation directly affects the time that it takes to measure similarity across the whole collection. Since we use all-pairs JS divergence as our baseline speed measurement in our implementation we made sure that the code is optimal. Due to the size

Divergence Type	MAP	P@5	R@5	Speedup
He LSH R=0.4	0.14	0.26	0.13	983.88
He LSH R=0.6	0.53	0.70	0.35	654.22
He LSH R=0.8	0.92	0.99	0.49	336.27
He k-d trees	0.92	0.99	0.49	1425.23

Table 3.3: Finding similar National Institutes of Health (NIH) grants: Performance comparison between all-pairs JS divergence, LSH, and approximate k-d trees using He distance. NIH grants were represented as topic distributions over 550 topics.

of the test collection, in our implementation, for each query document k we go over the list of n documents in the test collection.

When we configure LSH with its default radius value of R=0.6 we obtain MAP=0.53 which means that on average across all query documents we are able to retrieve half of the ten relevant documents for each query. With R=0.8 we achieve best LSH performance. In this case MAP=0.92, or on average 9.2 of the relevant documents are retrieved. The same MAP value is obtained with k-d trees, but unlike the LSH implementations whose speed improvement linearly decreases with the increase of the radius R, k-d trees gives us the best speed improvement. While many factors affect the speed of both approaches, such as the memory used and the size of the data set, in our comparison experiments we focused on using the readily available versions of both approaches assuming that they are already fairly optimized. In terms of precision (P@5) and recall (R@5), methods are characterized similarly to MAP. LSH with R=0.8 and k-d trees gives almost perfect scores on these two metrics. From the observed results we could reliably assume that the clustering task originally performed by Talley et al. [115] could have been performed with approximate similarity computation of the He distance with almost the same performance but with significant speed improvement.

3.7.2 Prior-Art Patent Search

Prior-art patent search is the process of searching and retrieving patents that are similar or relevant to the query patent. It is an important step in the process of reviewing patent applications as it helps the patent examiner in determining the extent of the proposed novelty by comparing it with existing granted patents and applications. Unlike typical IR tasks, retrieving related patents is recall oriented – from the patent examiner's perspective it is more important for the system to retrieve all related patents rather than having a small number of relevant patents at the top of the ranked list. Some of the challenges of prior-art patent search are discussed as part of the Text REtrieval Conference (TREC) Chemistry Track [79] and other conferences and workshops [114] as well as in recent books [78, 80].

Retrieving related patents poses many challenges mainly because of the vocabulary used in these documents but also because of the document structure. When writing patents authors use different strategies in order to protect their invention, extend it to different domains and make it appear different from previously granted patents. This includes using vague vocabulary, esoteric language and introducing new terminology. In addition, specific scientific domains by default use different vocabulary which creates another type of a challenge where the examiner needs to explore similar patents in completely different fields. Patent applications also contain different sections which differ in the writing style.

There are many benefits in using topic models for retrieving related and/or similar patents and to the best of our knowledge utilizing topic models on this task have not been explored before, at least on a level of large patent collections. Most importantly since patents are written with obfuscatory goals and large vocabulary differences, representing them in a shared topic space abstracts away from the actual vocabulary used across patents.

In this experimental setup we explore the feasibility of utilizing topic models on the task of retrieving related patents in large collections using approximate NN search techniques. As in the case on the previous experimental setup the goal is to asses the accuracy and speed of the approximate methods rather than comparing absolute performances. We showcase the benefits of casting the task of finding related patents as an approximate NN search problem in the probability simplex. In addition we present results of combining the ranked lists of topically related patents with the ranked lists obtained by Xue & Croft [129] using rank aggregation.

3.7.2.1 Experimental Setup

Our experimental setup consists of the United States Patent and Trademark Office (USPTO) collection of patents [94]. The collection contains $\sim$1.6M patents (with $\sim$4.5M tokens) published between 1980 and 1997. We follow the experimental setup used by Xue & Croft [129] and represent patents using the following six fields: title of invention (TTL), abstract field (ABST), primary claim (PCLM), drawing description (DRWD), detail description (DETD) and background summary (BSUM). Patents are mapped into a shared topic space using LDA with different number of topics. Due to the collection size and in order to perform efficient document-topic distribution inference, we use an online variational Bayes (VB) algorithm developed by Hoffman et al. [49]. More specifically we used the Vowpal Wabbit [69] implementation of this algorithm.

Topic representation of patents was done with a small subset of 32,609 tokens derived using a filtering approach which was based on token's frequency of occurrence across the whole collection. Tokens with less than 1k and more than 350k occurrences were removed as well numeric tokens and tokens with fewer than four characters. Since relevance judgments for patents are difficult to obtain, we use the patent's citation fields (UREF) entries as pseudo-relevance judgments. This approach was also used in [129]. Given a query patent the task is to find and retrieve topically similar patents.

3.7.2.2 Evaluation Task and Results

The test collection contains patents published in the period between 1980 and 1996 that contain the following five fields: TTL, ABST, PCLM, DRWD and DETS. The query data consists of patents published in 1997 that also contain the five fields and have more than 20 citations.

	Method Type	MAP	P@10	R@10
1	Xue and Croft	0.204	0.416	0.138
2	JS	0.172	0.343	0.111
3	He	0.178	0.345	0.112
4	He LSH R=0.4	0.056	0.161	0.051
5	He LSH R=0.6	0.091	0.248	0.078
6	He LSH R=0.8	0.161	0.344	0.111
7	He k-d trees	0.159	0.345	0.112
8	Aggregate rank (1 & 7)	0.232	0.442	0.145

Table 3.4: Prior-art patent search: Performance comparison between regular JS and He and approximate LSH and k-d trees based He distance. Patents were represented as topic distributions over 500 topics.

As in the case of evaluating the approximate methods for clustering NIH grants, here we compare the performance of regular JS and He computation, the approximate He using LSH (with three radius configurations) and k-d trees. Unlike the NIH task in this case we measure accuracy using precision of the top 10 (P@10) and recall of the top 10 (R@10) and we also compute MAP.

Accuracy results were computed using 70k patents chosen from the complete test collection that also contain the relevant patents for the query set. The query set is also the same as in [129]. Topics over patents were inferred using Vowpal Wabbit [69] configured with default values for the hyperparameters α = 0.1 and β = 0.1 and number of training passes set to five. Table 3.4 shows the obtained accuracy results across the three metrics when using LDA with 500 topics. In the first row of this table we present results of re-running the approach of Xue & Croft [129] with their best P@10 weight type (tf weight) and all five patent fields (field="all") on the same test collection of 70k patents. In this experimental setup LSH and k-d trees implementations were configured to return the top 200 NN points.

LSH with radius R=0.8 and k-d trees with default settings achieve the same P@10 and R@10 as in the case with regular JS divergence and He distance. MAP doesn't follow the same behavior since both approximate approaches are not configured to evaluate all points which is the case with regular JS and He.

Divergence Type	T=50	T=100	T=200	T=500
JS	0.212	0.266	0.306	0.343
He	0.222	0.273	0.320	0.345
He LSH R=0.4	0.168	0.170	0.181	0.161
He LSH R=0.6	0.219	0.252	0.276	0.248
He LSH R=0.8	0.223	0.274	0.319	0.344
He k-d trees	0.223	0.274	0.320	0.345

Table 3.5: Prior-art patent search: Performance comparison between regular JS and He and approximate LSH and k-d trees based He distance over P@10 for LDA models with different topic configurations.

Divergence Type	T=50	T=100	T=200	T=500
JS	6.4	4.0	2.3	1.0
He LSH R=0.4	85.6	75.3	57.8	35.6
He LSH R=0.6	59.2	53.4	38.1	27.9
He LSH R=0.8	46.1	33.0	29.0	16.7
He k-d trees	793.6	410.4	224.6	98.4

Table 3.6: Prior-art patent search: Relative speed improvement of the various approximate methods compared to regular JS divergence.

In Table 3.5 we show P@10 results obtained using LDA with number of topics set to T= 50, 100, 200 and 500 across results obtained from the various similarity methods.

As in the case of the NIH grants, we performed relative speed comparison of the various approximate methods with the regular JS divergence computation. More specifically we treat the best model configuration with 500 topics as our baseline speed (relative speed of one). From Table 3.5 we also see that performance across different similarity methods is not sensitive to the topic configuration of the model.

We achieve tremendous improvement in speed using k-d trees approximation of the He distance. This speed gain is significantly better compared to the speed gain achieved by LSH with R=0.8. The speed gain is consistent across the different topic configurations.

3.7.2.3 Combining Retrieval Approaches using Rank Aggregation

A straightforward way of utilizing topic models on a document retrieval task is to combine the results obtained as a standalone retrieval model with the results of an existing approach. While earlier work on using topic models in IR relies on integrating out topics to combine unigram and latent-variable models (e.g. [128, 131]) we decided to explore the most straightforward approach since it avoids the complexity of the integration and the parameter selection problem of interpolating the two probability models. Furthermore exploring weighted linear combinations offers lowest time complexity over more complex approaches. In case of the prior-art patent retrieval we could use the approach developed by Xue & Croft [129] as an existing approach and combine its ranked lists with the ranked lists obtained from the k-d trees similarity computation. We setup an experiment to explore whether such an approach would yield improvements and in this subsection we show the results obtained.

Rank aggregation is an active area of IR research which explores methods for combining ranked lists across different retrieval models. An overview of rank aggregation approaches are given in [29]. Examples of rank aggregation approaches are also given in [37], [64] and [105], to name a few. While there are various ways of performing the aggregation, for the patent retrieval task we wanted to utilize the most straightforward set and therefore we explored the approaches proposed by Shaw & Fox [105]. In particular we explore and experimented with the following five approaches: $CombMIN$, $CombMAX$, $CombSUM$, $CombANZ$ and $CombMNZ$.

For two ranked lists r_1 and r_2 and for a given ranked document D, that exists in both lists, $CombMIN$ and $CombMAX$ use the minimum $CombMIN(D) = min(r_1(D), r_2(D))$ and the maximum $CombMAX(D) = max(r_1(D), r_2(D))$ score of the two respectively. $CombSUM$, as the name implies, for a given ranked document that exists in both lists assign the sum of the two individual scores $CombSUM(D) = r_1(D) + r_2(D)$. $CombANZ$ combines the two scores of the given document that exists in both lists by computing its average $CombANZ(D) = \frac{r_1(D)+r_2(D)}{2}$ while $CombMNZ$ sums the two scores and multiplies them by the number of ranked lists where that document exists which in our case is always two, $CombMNZ = 2(r_1(D) + r_2(D))$. For the remaining documents in both ranked lists that exist in only one of them all of the five approaches retain the original score.

Combining ranked lists generated using different scoring functions (i.e. different retrieval models) requires first performing normalization. Following [29] we experimented with two types of normalization: (1) normalizing the original scores and (2) converting the actual ranks into normalized similarity scores. Score normalization is performed using the maximum and the minimum score values in the original ranked list:

$$normalized\ similarity \ = \frac{similarity\ -\ min(similarity)}{max(similarity)\ -\ min(similarity)} \tag{3.5}$$

The following formula is used for the rank based normalization:

$$normalized\,rank \ = 1 - \frac{rank-1}{|retrieved\ documents|} \tag{3.6}$$

Using the same patent set as in [129] we compared the performance of the five aggregation approaches and we show their performance comparison results in Table 3.7. In an extensive study on combining rankings done by Lee [73] outputs of retrieval models were treated as classifier outputs. When training classifiers, it is common to use a held-out development set to optimize parameters in order to avoid overfitting. We follow the same approach and evaluate the aggregation approaches on a held-out set. Between the two types of normalization, rank based gives the best performance when utilized with $CombSUM$ or the $CombMNZ$ aggregation approach. Both of these approaches yield the same performance since in our case all documents exist in both ranked lists and therefore $CombMNZ = 2CombSUM$ which translates to obtaining the same ranking of documents in both aggregation lists.

Aggregation Type & Baseline	Normalized Score	Normalized Rank
Xue and Croft	0.173	0.173
JS	0.158	0.158
$CombMIN$	0.077	0.143
$CombMAX$	0.174	0.178
$CombSUM$	0.174	0.187
$CombANZ$	0.079	0.151
$CombMNZ$	0.174	0.187

Table 3.7: Evaluating aggregation approaches based on P@10 using normalized scoring and rank function values.

In the last row of Table 3.4 we show the rank aggregation result when the CombMNZ approach with rank normalization was applied on our test collection.

3.7.3 Fast Relevance Models using Topic Models

Pseudo relevance feedback (PRF) is an approach that automates the relevance feedback process without the need of user's input. For a given query, PRF first obtains a list of relevant documents using a particular IR model. It then assumes that the top n (e.g. $n = 10$) retrieved documents are relevant to the query and uses them to derive a new ranked list. The most straightforward PRF approach computes tf–idf score over a set of words found in the top ranked documents to decide on the terms which would be used for the query expansion. In this section, we introduce an approach within the relevance model (RM) framework originally developed by Lavrenko and Allan [71] to come up with a new model that utilizes topical similarity computed across all documents in the collection. In the past, topic models have been used in context of PRF. For example, in [7] authors use topics extracted from the top n documents to perform query expansion while in [133] topic models are used to re-rank the the top n documents from the original ranked list. In our approach we go beyond using topic models on the top n documents and incorporate into the PRF model topical similarity computed across all documents in the collection which, to the best of our knowledge, has not been explored before. We assume that this is due to the fact that all-pairs comparisons in the probability simplex, especially for large document collections, up until now was practically not feasible to perform.

3.7.3.1 Topic Model based RM

We start off by giving an overview of the work by Lavrenko and Allan [70] in which they showed that a fast RM could be achieved by algebraically re-arranging the ranking formula of the original RM [71]. This original work showed that good retrieval results could be achieved across documents D in a collection using the cross-entropy between the language model (LM) and the estimated RM R:

$$H(R\|D) = \sum_{i=1}^{n} P(w_i \mid R) \log P'(w_i \mid D) \tag{3.7}$$

The cross-entropy is computed across the vocabulary of the n-document collection and the probability $P'(w_i|D)$ is a smoothed version of the unigram model for document D. Lavrenko and Allan [70] approximated the RM using the query words q by marginalizing out the set of document models M:

$$H(w|R) \approx P(w|q) = \sum_{M} P(w|M)P(M|q) \tag{3.8}$$

For a given query, document models M are estimated from the set of documents returned by the query that we treat them as pseudo-relevant. Combining the above two equations Lavrenko and Allan [70] derived the following form for the cross-entropy:

$$
\begin{aligned}
H(R\|D) &= \sum_{i=1}^{n} P(w_i \mid R) \log P'(w_i \mid D) \\
&= \sum_{i=1}^{n} \log P'(w_i \mid D) \times \left[\sum_{M} P(w_i \mid M) P(M \mid q) \right] \\
&= \sum_{M} \sum_{i=1}^{n} \left[P(w_i \mid M) \log P(w_i \mid D) \right] \times P(M \mid q) \\
&= \sum_{M} H(M\|D) P(M \mid q) \qquad (3.9)
\end{aligned}
$$

With this algebraic re-arrangement the cross-entropy is computed over the set of pseudo-relevant documents and as such is no longer dependent on the actual query. Furthermore, since this computation is performed over the whole set of documents in the collection, it could be performed in the process of indexing the collection. Performing all-pairs divergence computation is in reality very costly and often timely impractical to perform. In [22], for instance, authors used the MapReduce framework [32] for this computation. While in general all-pairs need to be computed, since we only want to know the k nearest neighbors for each document, the size of output does not need to be quadratic in the size of the input.

Here we use the original observation that the RM is merely a query-weighted average of the cross-entropies for the given pseudo-relevant set of documents and the rest of the collection. For a given generative model, used for document representation, the cross-entropy gives a quantitative representation of how similar the two representations are and as such could be computed using different generative models that could potentially reveal a better semantic similarity across the documents in the collection aside from the LM. Using different generative model of document representation for computing the cross-entropy also comes from the fact that the cross-entropy is independent of the original query likelihood (QL) model $P(M|q)$. Here we use LDA representation of the documents in the collection to compute the cross-entropy. In our final models, we interpolate this LDA RM with the original document posteriors from QL:

$$H(R\|D) \approx \lambda \sum_{M} JS(M_{LDA}, D_{LDA}) P(M \mid q) + (1-\lambda) P(D \mid q)$$

Rather than computing cross-entropy, note that we compute JS divergence using the approximate methods described above. In some experiments, we found it better to approximate the prior $P(M \mid q)$ as an uninformative $1/M$. We term this simpler model LDARM1 while the model with the original document priors is referred to as LDARM2.

3.7.3.2 Experimental Setup

To infer topics distributions over the documents in this collection, due to the size of the collection, as in the previous task, an online VB algorithm [49] was used to speed up the estimates of the posterior document-topic distributions. Representing documents as probability distribution over topics involves specifying a particular vocabulary of words over which topic distributions are inferred. In order to avoid sparsity and reduce computational time, singleton words from the collection are not considered along with the top n (usually top 25 or 50) words which are considered as stop words as well as words whose frequency of occurrence is less than ten across the whole collection. Documents were represented using a set of T=500 topics and approximate NN search was performed with kd-trees as it was shown in the previous tasks that this setup yields faster performance.

3.7.3.3 Evaluation Task and Results

We use the Robust04 data collection consisting of 528,155 newswire documents. This collection was indexed with the Indri [112] open source search engine. Indri was also used to run all related search queries. Our query set consists of the 250 topic queries from the TREC 2004 Robust track. We use as a baseline Indri's QL model results. Queries were created using the description fields. Table 3.8 shows the performance results between the three models using the values of $\lambda = 0.2$ for LDARM1 and $\lambda = 0.1$ for LDARM2 which yield the best results for the two approaches. With the QL model results as baselines we measured the significant differences of the LDARM1 and LDARM2 model results

Model	MAP	P@10
QL(baseline)	0.1795	0.3073
LDARM1	0.1815	0.3149
LDARM2	0.1820	0.3153

Table 3.8: Performance comparisons between QL model and LDA based PRF models using MAP and P@10.

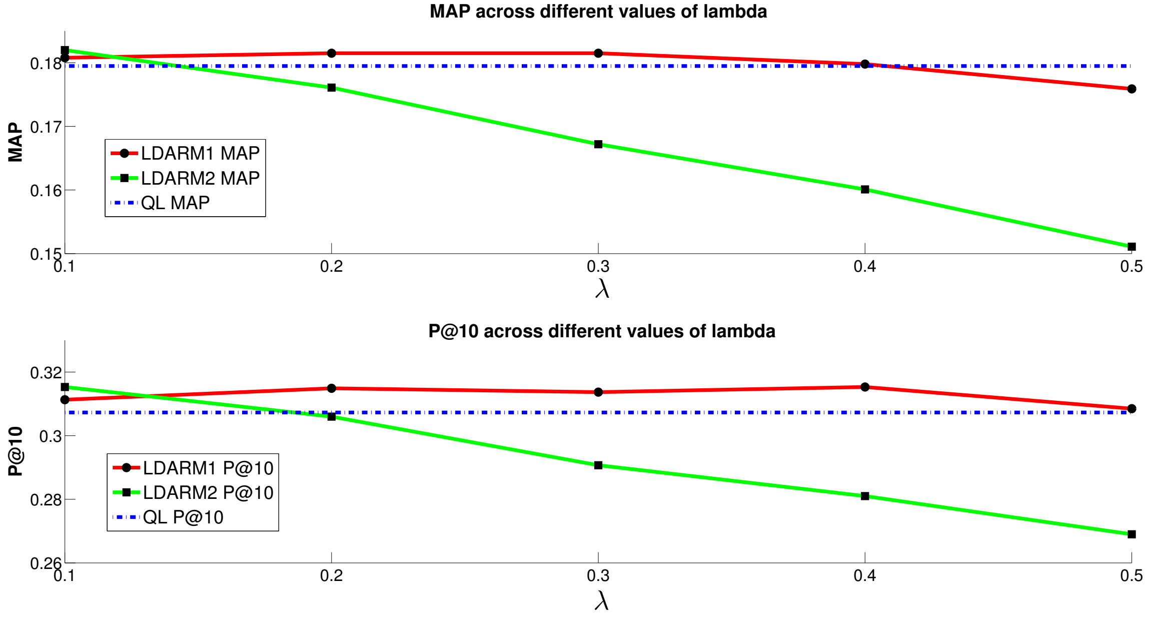


Figure 3.4: Relative performance comparison using MAP and P@10 between QL, LDARM1 and LDARM2 across different values of lambda.

using two-sided Fisher's randomization and bootstrap tests [109] which were run with 1M samples. Across the two tests both models achieve statistically significant (p-value$\leq$0.05) difference on the P@10 performance measurement.

Figure 3.4 shows the performance difference between the baseline (QL model), LDARM1 and LDARM2 models across different values of λ = 0.0, 0.1, 0.2, 0.3, 0.4 and 0.5 using MAP and P@10 as comparison metrics.

3.7.4 Finding Document Translation Pairs

In this section we evaluate our fast NN search methods on the task of finding document translation pairs. We give more detailed description of this task and the experimental setup in Chapter 4 while here we give a short task description. Unlike the previous tasks here we are dealing with an extreme form of vocabulary mismatch – documents are written in different languages. Latent variable models of text offer the ability to represent multilingual documents in a shared language-independent vector space i.e. the probability simplex. When documents written in different languages are represented in a shared space the task of finding document translation pairs is cast as a task of finding similar probability distributions. In the past this approach has been explored in many areas of MT and cross-language information retrieval (CLIR) but only on relatively small document collections. For example, Mimno et al. [85] showcased the performance of PLTM on the task of finding document translation pairs in a collection of 14,150 speeches from the English-Spanish Europarl collection [65]. The same task collection was used by Platt et al. [97] to introduce extensions of the principal component analysis (PCA) and probabilistic latent semantic analysis (PLSA) approaches for representing multilingual documents. We use the same task and test collection as in [85, 97] to evaluate the accuracy and speed of our approximate methods for computing similarity in the probability simplex. Aside from evaluating speed and accuracy, with this task we showcase how latent variable models of text could be used efficiently in large multilingual collections.

3.7.4.1 Experimental Setup

We use the same experimental setup as in Chapter 4. Mallet's implementation of PLTM [83] was used to represent these speeches into a shared topic space. More specifically we used four PLTM configurations with number of topics set to T=50, 100, 200 and 500. Across all model configurations we used the same settings of the Gibbs sampler and Dirichlet hyperparameters as in [85]. Document translation detection was performed by comput-

Divergence Type	T=50	T=100	T=200	T=500
JS	0.943	0.985	0.994	0.993
He	0.943	0.984	0.994	0.993
He LSH R=0.4	0.939	0.975	0.983	0.980
He LSH R=0.6	0.943	0.985	0.994	0.993
He LSH R=0.8	0.943	0.984	0.993	0.993
He kd-trees	0.949	0.989	0.995	0.996

Table 3.9: Finding document translation pairs: P@1 across different divergence measurements and different PLTM configurations.

ing regular JS divergence, He distance, the approximate LSH (with three radius configurations) and k-d trees across all English (query) and Spanish documents. Experiments performed in [85] evaluated the PLTM using regular JS divergence. Prior to performing our speed and accuracy analysis we replicated the results in [85] and therefore confirmed an equivalent experimental setup.

3.7.4.2 Evaluation Task and Results

In [85, 97] the performance of the latent variable models is evaluated based on the percentage of document translation pairs (out of the whole test set) that were discovered at rank one. This measurement is referred to as percentage at rank one which in regular IR jargon translates to precision of the top rank (P@1) with the size of the relevance set for each query being equivalent to one (each English speech has one Spanish translation). Table 3.9 shows the P@1 results across the four different measurements for all four PLTM models. As in the case of evaluating the approximate methods for clustering NIH grants and prior-art patent search here we compare the performance of regular JS divergence and He distance computation, the approximate LSH (with three radius configurations) and k-d trees.

Best performance with regular JS divergence is achieved using 200 topics. While using approximate methods we observe that we outperform regular JS and He across all topics. We attribute this to the false negatives created by the approximate NN search technique.

Divergence Type	T=50	T=100	T=200	T=500
JS	7.8	4.6	2.4	1.0
He LSH R=0.4	511.5	383.6	196.7	69.7
He LSH R=0.6	142.1	105.0	59.0	18.6
He LSH R=0.8	73.8	44.7	29.5	16.3
He kd-trees	196.7	123.7	76.7	38.5

Table 3.10: Finding document translation pairs: Relative speed improvement of the various approximate methods compared to regular JS divergence.

Rather than ranking true positives higher, these approaches either introduce false negatives or exclude true positives which are beneficial for this task. We achieve best performance with out approximate k-d trees using PLTM with T=500.

As in the case of the previous two tasks, we performed relative speed comparison of the various approximate methods with the regular JS divergence computation using 500 topics. We treat the relative speed of this setting as baseline (relative speed of one). Table 3.10 shows the relative speed differences between all-pairs JS, approximate LSH and kd-trees based He distance. The implementation of our all-pairs similarity computation using JS divergence uses hash tables to store all documents in the bilingual collection which is significantly faster than the code implementation used in the patent retrieval task.

The speed improvements follow the same trend as in the case of the previous two tasks: As we increase the LSH radius the speed improvement drops while the accuracy improves. With the increase of the number of topics we also see an increase in accuracy and at the same time the speed improvement decreases.

CHAPTER 4

ONLINE POLYLINGUAL TOPIC MODEL

4.1 Introduction

Earlier in Chapter 2 we pointed out the benefits of using topic models to represent multilingual collections and we also introduced the polylingual topic model (PLTM). We also mentioned that in the past the potential of PLTM has been evaluated on the task of finding document translation pairs [85, 97]. However current inference approaches for PLTM do not scale efficiently as the size of the multilingual collection grows. For example, the original implementation of PLTM uses Gibbs sampling which requires multiple iterations over the whole collection. The problem of efficient inference arises in conjunction with the problem of performing all-pairs comparisons which we discussed in the previous chapter.

In this chapter we present efficient inference method for multilingual topic models which is based on variational Bayes (VB) inference. We refer to this model as online PLTM (oPLTM). Through the task of finding document translation pairs embedded in a large multilingual corpus, we empirically evaluate the performance of oPLTM and showcase its efficiency over the existing PLTM model. We show that while being substantially more efficient in terms of the inference speed, oPLTM is as accurate as regular PLTM.

4.2 Online Variational Bayes for PLTM

Since their inception topic models have found their practical use in modeling different aspects of multilingual collections. For example in [17] authors propose a multilingual topic model which models pairs of words, i.e. matchings, that have similar document level contexts in two languages. Extensions of probabilistic latent semantic analysis (PLSA) and principal component analysis (PCA) to model multilingual collections are presented in the work by Platt et al. [97]. Fukumasu et al. [39] proposed an extension of the latent Dirichlet allocation (LDA) model that uses a hidden variable to control the “pivot” language in a multilingual collection that is used to draw the word topic assignments across document translation pairs.

Given a collection of document tuples d where each tuple contains one or many documents that are topically similar in different languages (L), the generative process of documents is modeled by PLTM in the following way. For each language l in the collection the model first generates a set of $t \in \{1, 2, ..., T\}$ topic-word distributions, φ_t^l which are drawn from a Dirichlet prior with language specific hyperparameter β^l: $\varphi_t^l \sim Dir(\beta^l)$. For each document d^l in tuple d, PLTM then assumes the following generative process:

- Choose $\theta_d \sim Dir(\alpha_d)$
- For each language l in document tuple d:
  - For each word w in d^l:
    - Choose a topic assignment $z_w \sim Multinomial(\theta_d)$
    - Choose a word $w \sim Multinomial(\varphi_z^l)$

The model first draws a tuple specific distribution over topics θ_d. As in the case with LDA, this distribution is drawn from a Dirichlet prior with hyperparameter α_d: $\theta_d \sim Dir(\alpha_d)$. Unlike LDA, where θ_d is used to specify the document specific distribution over topics, in case of PLTM this distribution is tuple specific. In the original PLTM model both α_d and $\beta_{1,...,L}$ are symmetric priors.

Going over each language l in the tuple, the model generates the N^l document specific words by first drawing a topic assignment z_w which is then used to select the language specific topic distribution over words φ_z^l. The actual word w is drawn from the chosen topic-word distribution.

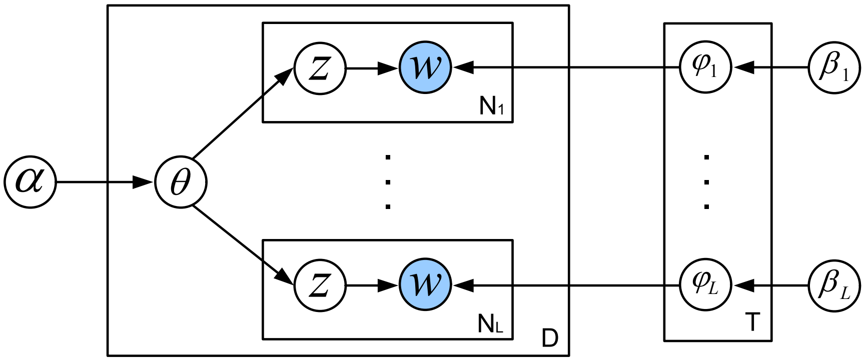


Figure 4.1: Graphical representation of the polylingual topic model (PLTM).

Figure 4.1 shows a graphical representation of the PLTM model. The original implementation of the PLTM uses the Gibbs sampling approach to estimate model's posterior distributions. With the Gibbs sampling approach extensions of LDA, such as PLTM, are more straightforward to conceptualize and implement. At the same time inference with Gibbs sampling over large collections is practically infeasible due to the need of performing multiple iterations over the whole collection.

In their original work, Blei et al. [16] derived a VB approach for approximating the posteriors in the LDA model. While more efficient to compute, compared to Gibbs sampling, the VB approach requires multiple iterations over the whole collection which also makes it impractical for use in large document collections. From the algorithmic layout of both approaches that we presented in Chapter 2 one could easily conclude that their time complexity grows linearly with the size of the collection. Furthermore, both algorithms require constant access to the whole collection either on disk or in memory. And while the most efficient implementations load the entire collection into memory their space complexity grows linearly with the size of the collection.

In the past researchers have proposed different approaches to alleviate this problem. For example in [132] authors proposed an algorithm which draws on the belief propagation [95] approach that uses the massage-passing algorithm to infer the posterior distributions. The Gibbs sampling approach has been extended in [21] and VB based approach has been pro-

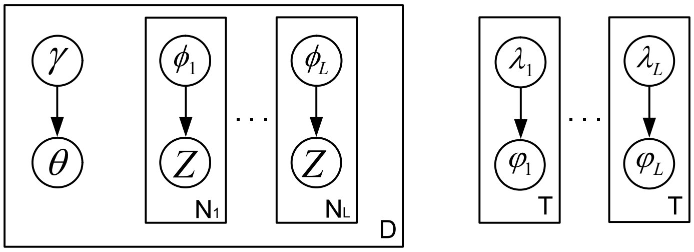


Figure 4.2: Graphical model representation of the free variational parameters for the online variational Bayes approximation of the PLTM posterior.

posed in [49]. A combination of the latter two is described in [84]. In our work we follow the VB inference approach [49]. Rather than iterating over the whole collection multiple times, in [49] authors use the stochastic gradient descent to optimize the topic-word distribution variational parameter on a batch of documents. The most important advantage of this approach is in its ability to generate good estimates of the LDA posteriors in a single pass over the whole collection.

4.3 Algorithmic Implementation

In this section we detail the online VB implementation of the approximate inference approach for computing the posterior tuple-topic and topic-word distributions across the different languages in our oPLTM model. Figure 4.2 shows the variational version of the oPLTM model and the free parameters used in our approach. As in the case of [49], we use the Expectation-Maximization (EM) algorithmic framework where in the E-step we iterate over all the documents in the batch b of d tuples and update the variational parameters γ^d and ϕ^{dl} for each tuple d until we find locally optimal values while holding the language specific variational parameters λ^l fixed:

$$\gamma_t^d = \alpha + \sum_l \sum_w \phi_{wt}^{dl} n_w^{dl} \tag{4.1}$$

$$\phi_{wt}^{dl} \propto \exp\left\{E_q\left[\log \theta_t^d\right] + E_q\left[\log \varphi_{tw}^{dl}\right]\right\} \tag{4.2}$$

In the M-step, for each language, we first compute the optimal value of $\tilde{\lambda}^l$ given the batch optimal values of ϕ^{dl}:

$$\tilde{\lambda}^l_{tw} = \beta + \frac{D}{|b|}\sum_{d=1}^{|b|} n^{dl}_w \phi^{dl}_{wt} \tag{4.3}$$

This is similar to the M-step performed in the batch VB. Unlike the batch VB where we have access to the whole collection, in this case we assume that across the whole collection we have the batch optimal values of ϕ^{dl} repeated $\frac{D}{|b|}$ times where $|b|$ is the batch size. This value is then combined with the the variational parameter $\lambda^{l(b-1)}$ computed on the previous batch through weighted average:

$$\lambda^{lb}_{tw} \leftarrow (1-\rho_b)\,\lambda^{l(b-1)}_{tw} + \rho_b \tilde{\lambda}^l_{tw} \tag{4.4}$$

Averaging is performed using a decay function $\rho_b = (\tau_0 + b)^{-k}$ where the k parameters controls the rate at which old values of $\lambda^{l(b-1)}$ are forgotten and τ_0 controls the rate at the starting stage of the algorithm. A detailed layout of these steps is shown in Algorithm 1.

Optimal values of γ^d and λ^l_t are used to approximate the tuple specific topic distributions θ^d and topic-word distributions φ^l_t for each language. Topic-word distributions inferred on a training set of parallel documents are the used to infer document-topic distributions on an unseen set of documents in different languages using the same algorithm without the updates in the M-step. In the following section we evaluate the performance of oPLTM and perform training and test time analysis.

4.4 Performance Analysis

We evaluated oPLTM in terms of its accuracy and speed and compared its performance with the original Gibbs sampling implementation of the PLTM [85]. In this section we

Algorithm 1 Online variational Bayes for PLTM

```
initialize $\lambda_l$ randomly
obtain the $b$th mini-batch of $m$ tuples
for $b = 1$ to $\infty$ do
    $\rho_b \leftarrow \left(\frac{1}{\tau_0+b}\right)^{\kappa}$
    initialize $\gamma_b$ randomly                                        ▷ Begin E-step:
    for each document tuple $d$ in mini-batch $b$
    for $d$ in $b$ do
        repeat
            for each language $l$ in document tuple $d$
            for $l \in 1,\ldots,$L do
                $\phi_{wt}^{dl} \propto \exp\left\{E_q\left[\log\theta_t^d\right] \; + \; E_q\left[\log\varphi_{tw}^{dl}\right]\right\}$
            end for
            $\gamma_t^d = \alpha + \sum_l \sum_w \phi_{wt}^{dl}\, n_w^{dl}$
        until convergence
    end for
    for $l \in 1,\ldots,$L do                                           ▷ Begin M-step:
        $\tilde{\lambda}_{tw}^{l} = \beta + \frac{D}{|b|}\sum_{d=1}^{|b|} n_w^{dl}\phi_{wt}^{dl}$
        $\lambda_{tw}^{lb} \leftarrow (1-\rho_b)\,\lambda_{tw}^{l(b-1)} + \rho_b\tilde{\lambda}_{tw}^{l}$
    end for
end for
```

demonstrate the efficacy of oPLTM over large multilingual document collections. Evaluations are performed on the cross-language information retrieval (CLIR) task of finding document translation pairs. More specifically, given a query document in English language the task is to retrieve its translation document in the foreign language. The model performance is measured by the number of times the retrieved translation document is at rank one, i.e. precision of the top rank (P@1).

Using topic models we first represent the test collection of bilingual documents in a shared topic space. Once documents are represented in this language independent space the task is formulated as finding similar probability distributions. We use Jensen-Shannon (JS) divergence to compute similarity between the query document and the documents in the foreign language and create a ranked list. Prior to inferring topics onto the test collection we first train the PLTM using a parallel collection of documents. Our training and test sets consists of a subset of English-Spanish speeches from the Europarl collection [65]. The training and test sets were constructed following the guidelines outlined in [97]. The training set consists of 64,5k parallel speeches distributed across 374 Europarl sessions from the years 1996-1999 and the year 2002 and a development set which consists of speeches from sessions in 2001. The test set contains speeches from the year 2000 and the first nine months of 2003.

4.4.1 Efficient Document Translation Detection

As stated earlier, in the past topic models have been explored and their potential showcased on the task of finding document translation pairs [85, 97]. Both of these approaches however explored only modest size collections of bilingual documents with less than 20k documents. In this work we are interested in expanding the PLTM to a much larger document collections and in this section we empirically show that we could achieve efficient representation of multilingual documents in large document collection without significant loss in accuracy while significantly improving the processing time.

English	Spanish
1. animals	1. animales
2. animal	2. prohibición
3. disease	3. carne
4. export	4. fiebre
5. foot	5. aftosa
6. mouth	6. exportación
7. meat	7. comisión
8. feed	8. fischler
9. fischler	9. crisis
10. crisis	10. animal

English	Spanish
1. funds	1. millones
2. million	2. fondos
3. year	3. euros
4. fund	4. para
5. billion	5. irlanda
6. ireland	6. estructurales
7. structural	7. fondo
8. irish	8. irlandés
9. funding	9. total
10. budget	10. presupuesto

English	Spanish
1. health	1. productos
2. food	2. salud
3. products	3. alimentos
4. consumers	4. medicamentos
5. scientific	5. alimentaria
6. product	6. consumidores
7. risk	7. para
8. labeling	8. pública
9. medicines	9. genéticamente
10. gmos	10. enfermedades

English	Spanish
1. world	1. paises
2. problems	2. para
3. country	3. mundo
4. consequences	4. como
5. poverty	5. problemas
6. global	6. consecuencias
7. problem	7. este
8. much	8. importante
9. poor	9. mundial
10. third	10. pobreza

English	Spanish
1. tourism	1. turismo
2. sport	2. deporte
3. internet	3. internet
4. exploitation	4. explotación
5. television	5. televisión
6. football	6. fútbol
7. sports	7. juegos
8. games	8. infantil
9. film	9. menores
10. olympic	10. material

English	Spanish
1. immigration	1. inmigración
2. belgian	2. belga
3. western	3. europa
4. helsinki	4. paises
5. communist	5. occidental
6. democracies	6. helsinki
7. tradition	7. tradición
8. west	8. democracias
9. world	9. comunista
10. bolkestein	10. bolkestein

English	Spanish
1. palestinian	1. israelí
2. israel	2. oriente
3. middle	3. palestina
4. east	4. palestinos
5. israeli	5. autoridad
6. authority	6. palestino
7. peace	7. israelíes
8. palestinians	8. medio
9. attacks	9. estado
10. united	10. sharon

English	Spanish
1. industry	1. industria
2. research	2. sector
3. sector	3. investigación
4. industrial	4. industrial
5. patent	5. innovación
6. innovation	6. marco
7. industries	7. industriales
8. technology	8. patente
9. technological	9. sectores
10. sixth	10. tecnología

Figure 4.3: Example sets of topic-word distributions inferred on a $\sim$64k subset of English-Spanish Europarl speeches using oPLTM configured with 400 topics. For each topic-word distribution we show the top ten most probable words for that topic.

4.4.1.1 Experimental Setup

We train oPLTM with number of topics T set to 50, 100, 200 and 500. Topic distributions were then inferred on the test collection using the trained topics. We then performed all-pairs comparison using JS divergence. We measured the total time that it takes to train and infer topics on a single machine consisting of Xeon quad processors with a clock speed of 2.66GHz and a total of 16GB of memory. For our oPLTM we used a Python implementation. We use Mallet's [83] implementation of the original PLTM which is Java based.

As an illustration in Figure 4.3 we show the top ten most probable words in a sample set of eight topic-word distributions that were inferred on our training set of English-Spanish Europarl speeches using oPLTM configured with 400 topics.

4.4.1.2 Evaluation Task and Results

As stated earlier, performance of the four oPLTM models was evaluated based on P@1 metric which was also used by [97] to show the absolute performance comparison. In [97] the test collection consists of speeches whose word length is greater or equal to 100 (total of 14,150 speeches) of the original test collection which we also use in our experiments. Figure 4.4 shows the results of these comparisons.

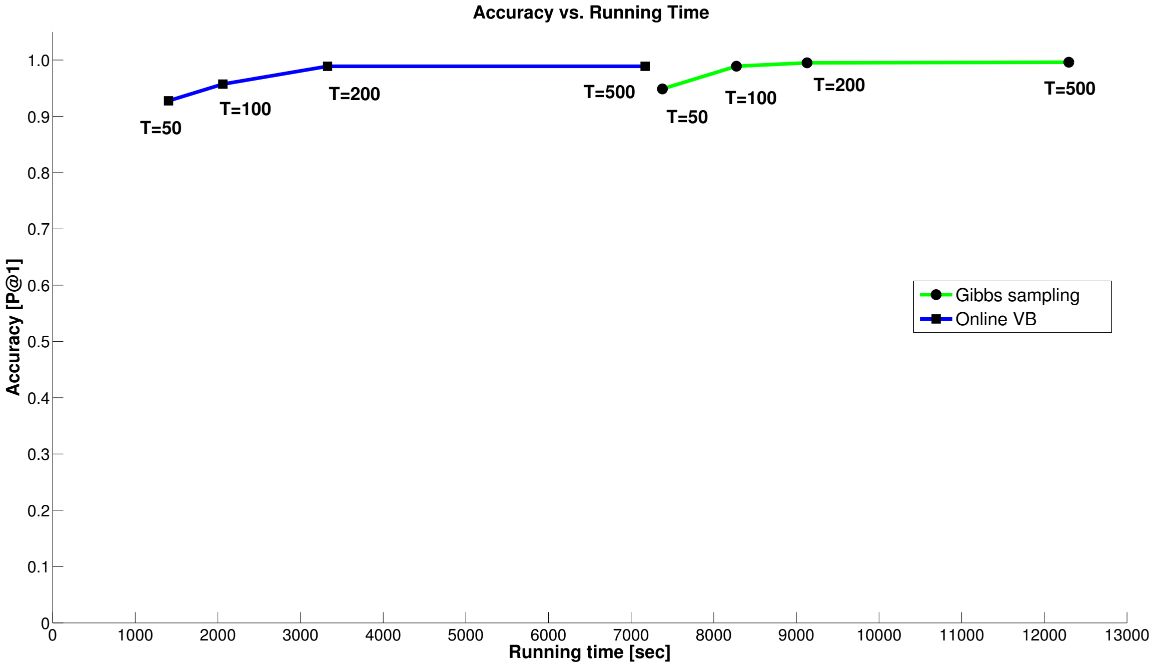


Figure 4.4: Speed vs. accuracy: Comparison between oPLTM and PLTM across four different model configurations T=50, 100, 200 and 500.

Across the four topic configurations we observe an accuracy improvement as we increase the number of topics. This trend is present in both families of models. While in this figure we present total time, which includes the training and test time, we observed that when we increase the number of oPLTM topics from 50 to 500 the speed improvement drops by a factor of 2.9 within the training step and by a factor of 4.45 in the test step. For a configuration with 50 topics, oPLTM is four times faster than PLTM while for 500 topics the speed ratio drops to almost two times. We also observe that the running time of oPLTM with 500 topics approaches the running time of PLTM with 50 topics. The gradual drop in speed improvement with the increase of the number topics is mostly attributed to the computation of the digamma function [10] whose time complexity increases linearly with the number of topics.

4.4.2 Efficient Training on Large Collections

We developed oPLTM in order to utilize the potential of PLTM on very large multilingual collections. We evaluated the efficiency of oPLTM on such collections by comparing the training speed between oPLTM and PLTM. Since very large parallel collections are often scarce, we created multilingual collections of different lengths by multiplying the original English-Spanish Europarl collection of $\sim$64k speeches. We came up with 6 such collections of lengths 50k, 100k, 250k, 500k, 750k and 1M. In this experimental setup we also observe how the performance of both models scale in terms of the collection size. Figure 4.5 shows comparison results across topic configurations with 50 and 500 topics.

In terms of collection size oPLTM configuration with 500 topics across collections of different size is faster than the Gibbs sampling configuration with 50 topics. We should also emphasize that MALLET's memory requirements scale with the collection size where PLTM requires fixed memory proportional to the number of topics and vocabulary size. For example reading collections beyond 500k documents, MALLET required a memory pool of 8G compared to the fixed size of 2G for oPLTM. Gibbs sampling speed could be easily attributed to the fact that in order to obtain good posterior estimates this approach requires multiple iterations over the collection. For collection sizes of 50k and 100k the training time for the oPLTM with T=500 approaches the training time of Gibbs sampling with T=50 and as we increase the collection size this proximity dissipates.

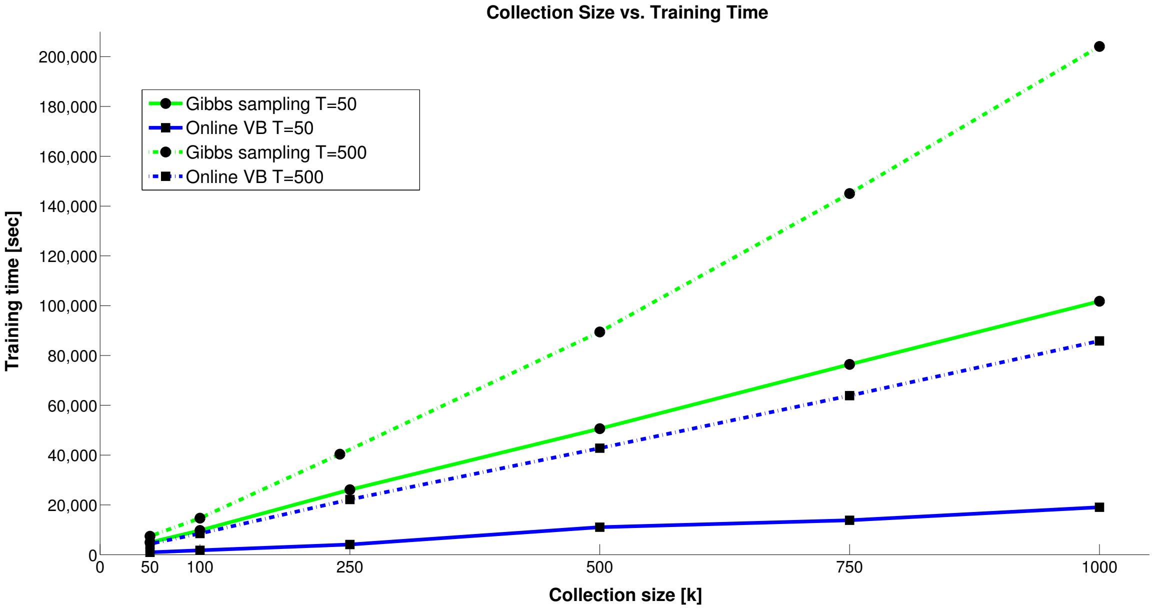


Figure 4.5: Collection size vs. training time: Comparison between oPLTM and PLTM over multilingual collections of 50k, 100k, 250k, 500k, 750k and 1M speech pairs.

CHAPTER 5

BOOTSTRAPPING TRANSLATION DETECTION AND SENTENCE EXTRACTION FROM COMPARABLE CORPORA

5.1 Introduction

In statistical machine translation (SMT), the quality of the translation model is highly dependent on the amount of parallel data used to build it. Parallel data has usually been generated through the process of human translation, which imposes significant costs when building systems for new languages and domains.

To alleviate this problem, researchers have looked into utilizing comparable corpora—a collection of documents in two languages that share the same domain. Or more formally defined by [40]—a collection of multilingual documents which are only topically aligned but not necessary translations of each other. An exemplar comparable corpus, that we are going to use to illustrate our approach, are daily news articles generated in multiple languages.

In this chapter we describe a bootstrapping approach for the problem of detecting document translation pairs and extracting parallel sentences from comparable corpora. Compared to previous approaches for extracting parallel sentences, this approach does not depend on linguistic resources such as parallel documents or translation dictionaries but only on observing documents published on similar dates and the co-occurrence of a small number of identical tokens across languages. We utilize our fast, online inference for a latent variable model to represent multilingual documents in a shared topic space. Unlike typical multilingual topic models that are trained on parallel data, we generate the training set from comparable corpora. Additionally, with our approximate nearest-neighbor (NN)

Training corpus	WMT’11
Parallel News Commentary	23.75
Extracted from comparable Gigaword	24.28
Both	24.92

Table 5.1: Summary of translation results: bilingual evaluation understudy (BLEU) score of systems trained from clean parallel text, bitext extracted from comparable corpora, and a combination of both.

search techniques we achieve fast translation detection for documents represented in the probability simplex and the metric space thus making our approach efficient for large comparable corpora. On the task of translation detection, we demonstrate that our approach achieves the same performance as a polylingual topic model (PLTM) trained on parallel text. More importantly, using only sentences extracted from comparable corpora, we are able to train a machine translation (MT) system that outperforms a baseline system trained on parallel collection.

While most previous approaches for mining comparable corpora heavily depend on initializing the learning process with some translation dictionaries or parallel text, in this chapter we will present an approach that generates and uses latent variable models of text to detect document translation pairs and extract parallel sentences with only a minimum cross-language prior knowledge: the publication dates of articles and the tendency of some vocabulary to overlap across languages. Table 5.1 shows the performance results of a MT system trained on parallel sentences extracted from the English and Spanish Gigaword collections using our proposed approach. Processing only four years of Gigaword news stories we are able to outperform the WMT’11 baseline system trained on parallel News Commentary corpus. The novelty of our approach lies in the combination of the several components of our processing pipeline which allows us to utilize topic models for mining large comparable corpora by combining the power of efficient approximate NN search techniques that we presented in Chapter 3 and our fast PLTM inference – online PLTM (oPLTM) (Chapter 4).

In the following section we introduce our approach of representing documents in a comparable corpora using topic models. We then detail the processing pipeline components (§ 5.3.1). We use the English-German and English-Spanish Europarl collections of speeches to demonstrate the effectiveness of our pipeline in retrieving document translation pairs without prior translation knowledge (§ 5.3.2). We first showcase the ability of our approach to extract parallel sentences from comparable corpora using a synthetic collection, generated from English-Spanish Europarl (§ 5.4.1), and then perform a more detailed and realistic performance analysis using the Gigaword collection (§ 5.4.2).

5.2 Previous Work on Mining Comparable Corpora

Most previous, if not all, approaches for mining comparable corpora heavily depend on bilingual resources, such as translation lexica, bitext, and/or a pretrained baseline MT system. In our work, in contrast, we investigate building MT systems from comparable corpora without such resources. In a widely cited early paper, [89] use a bilingual dictionary and a collection of parallel sentences to train IBM Model 1 [20] and a maximum entropy classifier to determine whether two sentences are translations of each other. In [116] and [107] authors detect parallel sentences by training IBM Model 1 and maximum entropy classifiers, respectively. In later work on detecting sentence and phrase translation pairs, [23] and [48] use SMT systems to translate candidate documents; [99] use parallel data to train a translation equivalence model; and [120] use a translation lexicon to build a scoring function for parallel documents. More recently, [76] trained IBM Model 1 on bitext to detect translationally equivalent phrase pairs within single microblog posts. In [1], [121], and [41], rather than trying to detect translated sentence pairs directly, authors translate the entire source language side of a comparable corpus into the target language with a baseline SMT system and then search for corresponding documents.

On the other hand, there exist approaches that mine comparable corpora without any prior translation information or parallel data. Examples of this approach are rarer, and we briefly mention two: [36] use singleton words (hapax legomena) to represent documents in a bilingual collection for the task of detecting document translation pairs, and our earlier work [67] where we construct a vocabulary of overlapping words to represent documents in multilingual collections. Our approach demonstrates high precision vs. recall values on various language pairs from different languages and writing systems when detecting translation pairs on a document level such as Europarl sessions. Recently proposed approaches, such as [63] use monolingual corpora to estimate phrase-based SMT parameters. Unlike our approach, however, they do not demonstrate an end-to-end SMT system trained without any parallel data.

Our approach most basically differs from these and other previous approaches by not relying on any initial translation dictionary or any bitext to train a seed SMT system. Therefore, the primary experimental comparison that we perform is between no bitext at all and a system trained with some bitext.

5.3 Bootstrapping Approach

The bootstrapping approach is a two-stage system that utilizes our previous work on detecting document translation pairs [67] as its first step. We refer to this step as Overlapping Cosine Distance (OCD). The output of OCD is a discovered set of document translation pairs which are then used to train PLTM and infer topics onto a test set in order to discover parallel text. We use our oPLTM approach, outlined in Chapter 4, which allows us to efficiently process multilingual collections of large sizes, such as comparable corpora where documents could be in the order of magnitude of millions.

5.3.1 Bootstrapping Pipeline

Figure 5.1 shows the layout of our approach. The output of the OCD step is a single list of document pairs ranked by translation similarity. We use the top n ranked document pairs which we believe are translations of each other.

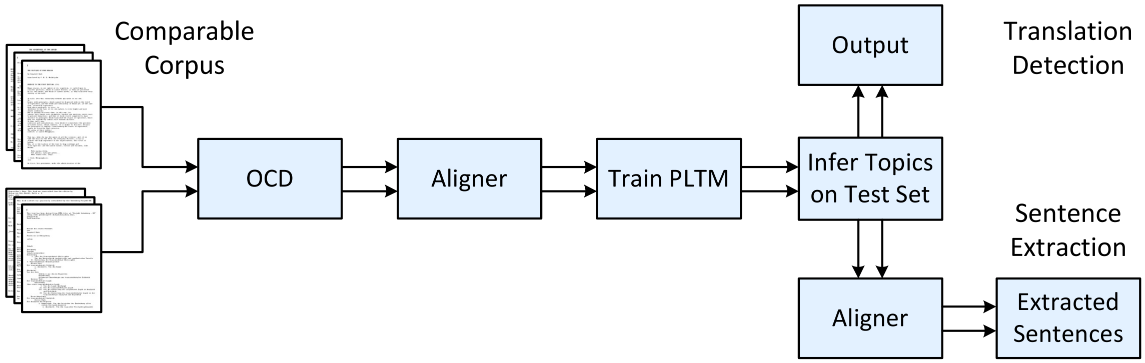


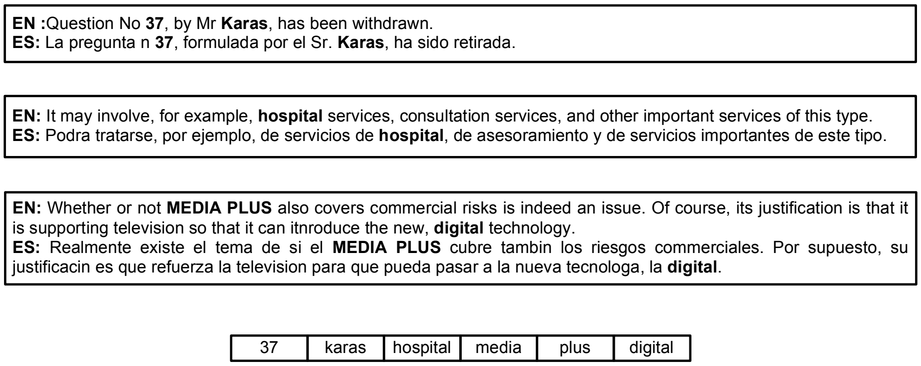
Figure 5.1: The bilingual collection processing pipeline.

These document pairs are then fed into a sentence aligner. In our experiments we used Moore's aligner [87]. The output of the sentence aligner is further aligned set of text structure. For each document pair, depending on the available per document information, we perform alignment on different levels. If we have explicit sentence information, alignment is performed on common sentence level. Otherwise we concatenate sentences that are part of a higher than sentence document structure (such as speeches in the Europarl collection) into a single sentence and perform the alignment.

PLTM is then trained using the aligned document level information. Using the trained model we infer topics on a particular test set. Once represented as points in the topic space, documents are then compared for similarity using Hellinger (He) distance. Results from these comparisons create a single ranked list of text translation pairs, which are on a sub-document length level. From this single ranked list, using thresholding, we again extract the top n candidate translation pairs that are then fed to an aligner for further refinement.

5.3.2 Discovering Document Translation Pairs

For a given comparable corpus, we use our OCD approach to represent documents in both languages as features using the template feature vector whose dimensions are the term frequency–inverse document frequency (tf–idf) values computed on the overlap-



EN : Question No **37**, by Mr **Karas**, has been withdrawn.
ES: La pregunta n **37**, formulada por el Sr. **Karas**, ha sido retirada.

EN: It may involve, for example, **hospital** services, consultation services, and other important services of this type.
ES: Podra tratarse, por ejemplo, de servicios de **hospital**, de asesoramiento y de servicios importantes de este tipo.

EN: Whether or not **MEDIA PLUS** also covers commercial risks is indeed an issue. Of course, its justification is that it is supporting television so that it can itnroduce the new, **digital** technology.
ES: Realmente existe el tema de si el **MEDIA PLUS** cubre tambin los riesgos commerciales. Por supuesto, su justificacin es que refuerza la television para que pueda pasar a la nueva tecnologa, la **digital**.

37	karas	hospital	media	plus	digital

Figure 5.2: Example parallel sentences extracted from three different documents in the English-Spanish Europarl corpus. Highlighted in black are words that occur in both document pairs which are used to construct the feature template vector (shown on the bottom).

ping words. Figure 5.2 shows sample sentences extracted from parallel documents in the English-Spanish Europarl collection along with the constructed feature template vector.

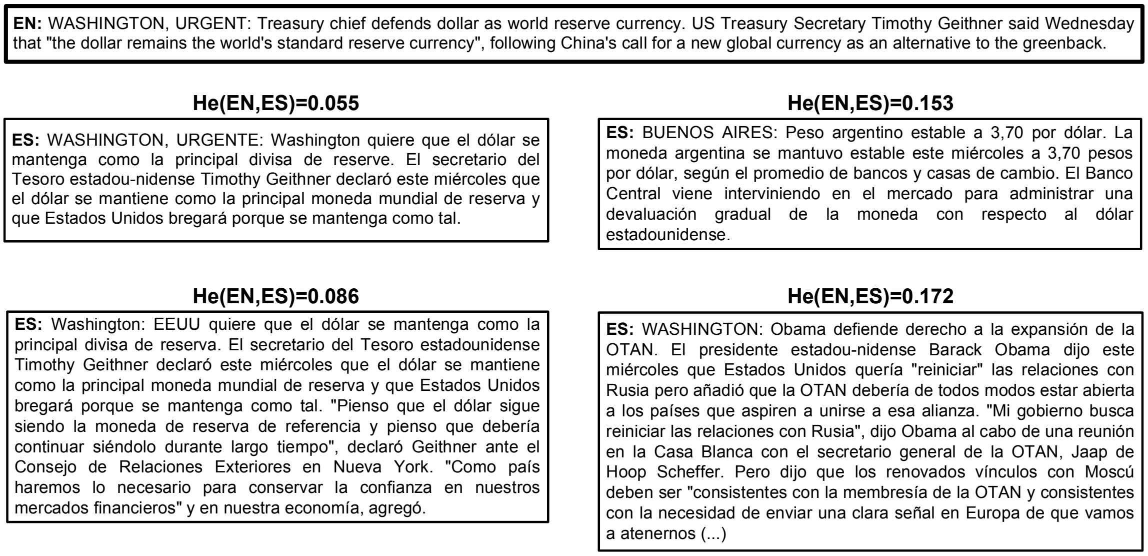
For a given comparable corpus, OCD assumes that there is a set of words that exist in both languages that could be used as features in order to discriminate between documents that are translations of each other, documents that carry similar content and documents that are not related. Firstly, for each language in the collection a vocabulary is created. Words found in both source (s) and target (t) languages are extracted and the overlapping list of words are then used as dimensions for constructing a feature vector template whose length is $\min(s, t)$. Documents in both languages are then represented using the template vector whose dimensions are the tf–idf values computed on the overlapping words which we now consider as features.

While the number of overlap words is dependent on the families of the source and target languages and their orthography, in our previous work [67] we showed that this approach yields good results across language pairs from different families and writing systems such as English-Greek, English-Bulgarian and English-Arabic where, as one would expect, most shared words are numbers and named entities.

Figure 5.2 shows sample sentences extracted from parallel documents in the English-Spanish Europarl collection along with the constructed feature template vector.

When documents are represented as features using the template vector, their original language information is not relevant. In this language independent vector space, documents are compared against each other through the Cosine (Cos) distance as a similarity metric. We compute the Cos distance across documents in large collections using our LSH based approximation approach for computing this metric which we described in Chapter 3.

The result of the comparisons is a single ranked list of all document pairs. Compared to the common cross-language information retrieval (CLIR) task where a set of document queries is know in advance, in this case there is no prior information on the documents in the source language that may or may not have translation documents in the target language of the collection. Therefore, creating separate ranked lists is not suitable for a discovery task of this nature and any attempts to show performance results on tasks of this nature by using prior knowledge of which source documents have translations in the target language of the collection oversimplifies the discovery task. Due to the length invariance of the Cos distance metric, the single ranked list may contain document pairs with high similarity value across all documents in the target language. This in turn could make the bulk of the top ranked document pairs contain the same source document. This issue in OCD is being resolved by applying length and diversity based filtering. The length filtering removes document translation pairs where the length of the target document t is not in the $\pm 20\%$ range of the source document s length $lf : 0.8 \leq \frac{|s|}{|t|} \leq 1.2$. For a given source document, diversity filtering is done by allowing only the top five ranked target document pairs to be considered in the single ranked list. Limiting the number of target documents for a given source document may discard actual document translation pairs such as in a comparable corpus of news stories where documents in the target language originate from large number of news source. While it may restrict more document translation pairs to be discovered, the diversity filtering, on the other hand prevents from limiting the number of discovered similar and translation documents to be from the same topic and domain and thus introduces diversity on another, domain or topic based, level.



EN: WASHINGTON, URGENT: Treasury chief defends dollar as world reserve currency. US Treasury Secretary Timothy Geithner said Wednesday that "the dollar remains the world's standard reserve currency", following China's call for a new global currency as an alternative to the greenback.

He(EN,ES)=0.055

ES: WASHINGTON, URGENTE: Washington quiere que el dólar se mantenga como la principal divisa de reserve. El secretario del Tesoro estadou-nidense Timothy Geithner declaró este miércoles que el dólar se mantiene como la principal moneda mundial de reserva y que Estados Unidos bregará porque se mantenga como tal.

He(EN,ES)=0.153

ES: BUENOS AIRES: Peso argentino estable a 3,70 por dólar. La moneda argentina se mantuvo estable este miércoles a 3,70 pesos por dólar, según el promedio de bancos y casas de cambio. El Banco Central viene interviniendo en el mercado para administrar una devaluación gradual de la moneda con respecto al dólar estadounidense.

He(EN,ES)=0.086

ES: Washington: EEUU quiere que el dólar se mantenga como la principal divisa de reserva. El secretario del Tesoro estadounidense Timothy Geithner declaró este miércoles que el dólar se mantiene como la principal moneda mundial de reserva y que Estados Unidos bregará porque se mantenga como tal. "Pienso que el dólar sigue siendo la moneda de reserva de referencia y pienso que debería continuar siéndolo durante largo tiempo", declaró Geithner ante el Consejo de Relaciones Exteriores en Nueva York. "Como país haremos lo necesario para conservar la confianza en nuestros mercados financieros" y en nuestra economía, agregó.

He(EN,ES)=0.172

ES: WASHINGTON: Obama defiende derecho a la expansión de la OTAN. El presidente estadou-nidense Barack Obama dijo este miércoles que Estados Unidos quería "reiniciar" las relaciones con Rusia pero añadió que la OTAN debería de todos modos estar abierta a los países que aspiren a unirse a esa alianza. "Mi gobierno busca reiniciar las relaciones con Rusia", dijo Obama al cabo de una reunión en la Casa Blanca con el secretario general de la OTAN, Jaap de Hoop Scheffer. Pero dijo que los renovados vínculos con Moscú deben ser "consistentes con la membresía de la OTAN y consistentes con la necesidad de enviar una clara señal en Europa de que vamos a atenernos (...)

Figure 5.3: English news story with its topically most similar (left) and dissimilar (right) Spanish news stories discovered within a He distance range of $He \in [0.0\,,\,0.2]$. News stories were extracted from the Gigaword collection using PLTM with T=30 trained on OCD output.

5.3.3 Topic Based Representation of Multilingual Collections

With the PLTM model we represent documents in multiple languages into a common shared topic space which allows us to perform similarity comparisons across documents and discover document translation pairs. Figure 5.3 shows an example English news story from the Gigaword collection with the topically similar and dissimilar news stories discovered in the Spanish collection using PLTM configured with 30 topics. We present the topically most similar and dissimilar news stories within a He distance range of $He \in [0.0\,,\,0.2]$.

5.4 Experiments and Results

We showcase the performance of our bootstrapping approach on two different tasks – CLIR and extracting parallel sentences. These tasks require two different pipeline setups whose difference is in the pipeline output. On the CLIR task, for a given query set of source documents that contain a known translation pair in the target collection, the pipeline outputs ranked lists of translation pairs. When extracting parallel sentences, we generate a single ranked list of document pairs which is then fed to an aligner for further refinement. On this task, we evaluate MT systems trained on extracted parallel sentences and we compare their performances against MT systems created using parallel collections. We show performance results across two official MT test sets.

5.4.1 Retrieving Translation Pairs

For a given query set of documents in a source language our task is to retrieve document translations. As in the case of the task in § 4.4 our performance metric is defined as the number of times the target language document was retrieved at rank one (i.e. precision of the top rank - P@1). We train and evaluate the performance of different PLTM settings with two bilingual collections – our English-Spanish Europarl collection, which we introduced in § 4.4, and a random split of the English-German Europarl collection of sessions. We consider the top n document translation pairs whose Cos similarity score is between the range of the max (i.e. the top one scored document translation pair) and $\frac{max}{2}$. We also utilize the length and diversity based filtering as outlined in the previous section.

Tables 5.2 and 5.3 show the pipeline performance compared to the performance of the PLTM model when trained with prior aligned document translation pairs. For this experimental setup we use Mallet's [83] implementation of the PLTM to train and infer topics with the number of topics set to T=50. Number of iterations and the values of the hyperparameters were set as in [85]. Document comparisons were done using the Jensen-Shannon (JS) divergence. We also compare the results of our pipeline with the results obtained when we consider the top n document translation pairs who are true translations of each other. We compute this set of translation pairs by traversing through the ranked list and stopping at the first occurrence of document pairs that are not translation of each other. We refer to the results obtained with this approach as oracle results. Length and diversity based filtering (len+div) results are shown across all documents and documents

Setup Type	P@1	len+div	$\|D\| \geq 100$	$\|D\| \geq 100$ & len+div
PLTM	0.755	0.758	0.861	0.844
Pipeline	0.756	0.757	0.865	0.846
Pipeline(oracle)	0.753	0.758	0.862	0.847

Table 5.2: English-German Europarl collection: Precision of the top rank (P@1) across different setups.

Setup Type	P@1	len+div	$\|D\| \geq 100$	$\|D\| \geq 100$ & len+div
PLTM	0.813	0.818	0.947	0.947
Pipeline	0.817	0.816	0.948	0.946
Pipeline(oracle)	0.814	0.809	0.943	0.941

Table 5.3: English-Spanish Europarl collection: Precision of the top rank (P@1) across different setups.

whose length is greater than or equally to 100 ($|D| \geq 100$). Tables 5.4 and 5.5 show the percentage of document translation pairs discovered at rank one whose bilingual evaluation understudy (BLEU) score measure [93] is greater than or equal to 0.90 (BLEU ≥ 0.90). We calculated the BLEU score by treating the target language document as the reference translation.

Results across the three setups were computed using all the speeches in the test collections. While there is a slight variance across the results of the three approaches, which could be easily attributed to the Dirichlet prior value of the PLTM hyperparameters, it is worth accenting that the performance of the setup where we use the exact true translation pairs obtained from the OCD step to train PLTM (oracle) yields worst performance over the

Setup Type	P@1	len+div	$\|D\| \geq 100$	$\|D\| \geq 100$ & len+div
PLTM	75.6	76.0	86.3	84.8
Pipeline	75.8	75.8	86.6	84.7
Pipeline(oracle)	75.4	75.9	86.3	84.8

Table 5.4: English-German Europarl collection: Percentage of document translation pairs discovered at rank one with BLEU ≥ 0.90 across different setups.

Setup Type	P@1	len+div	$\lvert D\rvert \geq 100$	$\lvert D\rvert \geq 100$ & len+div
PLTM	81.4	81.9	94.8	94.8
Pipeline	81.9	81.7	94.9	94.7
Pipeline(oracle)	81.5	81.0	94.5	94.2

Table 5.5: English-Spanish Europarl collection: Percentage of document translation pairs discovered at rank one with BLEU ≥ 0.90 across different setups.

setup where we consider the top ranked documents in the range of [max,$\frac{max}{2}$]. The latter approach also outperforms the baseline PLTM approach.

5.4.2 Extracting Parallel Sentences

To demonstrate the potential of the processing pipeline in extracting parallel sentences, we ran experiments where we compared the performance of SMT systems trained on extracted parallel sentences and parallel collections. MT systems were evaluated with the BLEU score on two official WMT test sets that cover different domains: News (WMT'11) and Europarl (WMT'08). MT system were trained using the Moses SMT system [66] following the guidelines for building a baseline system. We use two parallel training collections from WMT'11: English-Spanish News Commentary (v6) and Europarl (v6). We ran series of experiments over two types of comparable corpora – synthetic and actual. Default MT systems were trained using test domain specific language models (LMs) – English News Commentary for News test and English Europarl for the Europarl test.

Our synthetic comparable corpus was constructed from our English-Spanish Europarl collection which we introduced in § 4.4. This collection contains 64.5k parallel speeches distributed across 374 parallel documents. In our case we discard the prior information on the number of translation pairs and how speeches are aligned. We show relative performance comparison between PLTM models trained on true parallel speeches and extracted parallel speeches using our bootstrapping approach. From this collection we also create our second parallel training set through the following process. We extract all sentences found in our collection of speeches. Since these sentences are not aligned we go over the actual sentence aligned English-Spanish Europarl collection and keep the overlapping set of sentence pairs.

Our actual comparable corpus consists of news stories from the English (LDC2011T07) and Spanish (LDC2011T12) Gigaword collections.

We perform the following processing in each step of the pipeline. For the synthetic comparable corpus we start off by running OCD on the two language sets of Europarl sessions. In case of the Gigaword corpus we run OCD on days of news originating from multiple news agencies or more specifically on news stories originating from the same day which we consider as the "minimal supervision" in initiating the bootstrapping process. Since the OCD approach generates a single list of ranked document translation pairs, for the second stage of our pipeline we consider the top n document translation pairs. We define n to be all document translation pairs whose Cos similarity score is between the range of the max (i.e. the top one scored document translation pair in the single ranked list) and $\frac{max}{2}$. Unlike previous tresholding based on absolute values [119], this approach allows us to utilize threshold values that are automatically adjusted to the dynamic range of the Cos distance of a particular corpus. In case of the synthetic comparable corpus from the top n documents, speeches are extracted and concatenated as a single entry to the aligner. For the Gigaword corpus sentences from the top n news stories are extracted and are further aligned. The output of the aligner is then used as a training set for the PLTM model. For the Gigaword collection we represent each of the news stories using the per story aligned sentences. Once trained, we use the PLTM model to infer topics back on to the speeches and news stories. We than again create a single ranked list of translation pairs (across speeches and stories) by computing divergence based similarity using He distance as outlined in § 5.3.3.

Thresholding the single ranked list and keeping the top n ranked speech and news story pairs we obtain a list of what we believe are parallel documents which we then use to

Training Source	Parallel	Extracted	Test Set	
			News	Europarl
News Commentary	5k	0	17.45	15.90
News Commentary	10k	0	19.09	17.89
News Commentary	20k	0	20.73	18.90
Europarl	5k	0	15.29	19.10
Europarl	10k	0	16.72	21.59
Europarl	20k	0	18.25	22.81
Europarl extracted	0	8k	3.98	3.94
Europarl extracted	0	54k	4.05	9.96
Europarl extracted	0	136k	16.23	26.27

Table 5.6: BLEU score values on test collections from two different domains using MT systems developed on two different sources of training corpus - parallel and synthetic comparable.

extract sentence pairs. Sentences are finally processed through an aligner and then used as the training corpus to our MT system.

5.4.2.1 Synthetic Comparable Corpora

Typically approaches that mine comparable corpora to extract useful MT data (whether parallel sentences, translation lexicon, etc.) are evaluated based on the relative improvements of the MT system performance. In this experimental setup we follow the same notion but rather than showing relative improvements on the BLEU score, we ran tests to determine the absolute performance across different parallel and extracted training sets where the test set is either in- or out-of-domain relative to the training set. We do so by building translation models using random subsets of lengths: 5k, 10k and 20k from the two WMT'11 parallel collections and by running our approach on the synthetic comparable corpus using the simplest PLTM configuration with T=50 and He=0.1 threshold. Table 5.6 shows when similar BLEU score performance could be achieved between the two training data types.

With our largest extracted synthetic collection we are able to outperform both parallel collections on the Europarl test set. On the other hand, our largest extracted synthetic collection performs worse than the smallest in-domain parallel collection on the News test set. Interesting to note is that as we double the size of the extracted collection from 54k to 136k we obtain a significant increase in performance. This is mostly attributed to the content, topic diversity and length of the speech translation pairs ranked highest in the single ranked list. While important for training, parallel collection information of this nature could not be directly inferred by the value of the ranking metric.

5.4.2.2 Comparable Corpora from News Stories

In the Gigaword collection, news stories are generated from various news agencies in different languages. On any given day, depending on the popularity and the coverage, a news story in English may or may not be generated in a different language. To perform fair evaluation with the WMT'11 News test we considered news stories published in prior years and processed news stories form 2004, 2005, 200 and 2010 from multiple news agencies. We did not consider news stories from 2006-2008 due to a known issue with diacritic marks in the Spanish collection.

Figure 5.4 shows BLEU score performances across different years. We show results on MT systems whose LM were trained on the test domain data (which we refer to as Test domain LM) and MT systems with LM trained on the English set of extracted sentences – Gigaword Extracted LM, LM(GW).

While there is a variation across individual years, as we accumulate more years we see a linear BLEU score increase on both test sets. On the Europarl test, using LM(GW) gives us worst performance across all years due to difference in domains.

Table 5.7 shows the performance comparison of our MT system trained on extracted parallel sentences from four years of GW data with a MT system trained on two WMT'11 baseline parallel collections: Europarl (EP) and News Commentary (NC). As stated at the beginning of this section, all MT systems were trained using test domain LM – English NC for News test and English EP for the Europarl test.

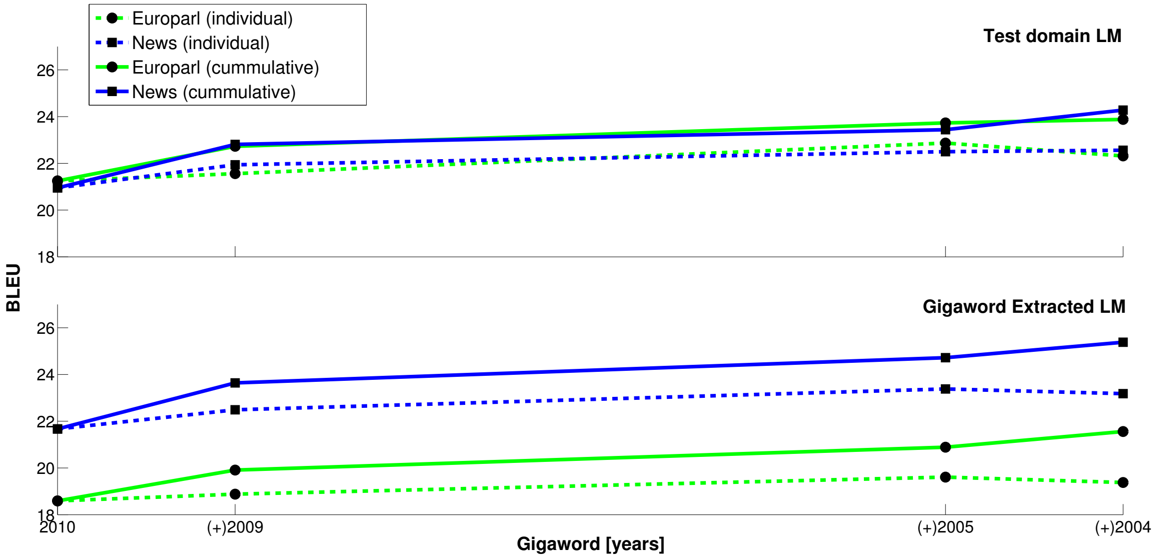


Figure 5.4: MT performance across models trained using extracted parallel sentences from individual and cumulative Gigaword years.

On the News test set, our extracted set of parallel sentences from only four years of Gigaword data outperforms the NC and EP parallel collections. Combining extracted with parallel corpus we see further improvement. On the Europarl test set, unsurprisingly, the EP baseline system performed very well.

On the News test set we ran the randomization test [109] in order to determine the statistically significant differences between the results of the different MT systems. More specifically, we were interested in determining the statistical difference in the performance of MT systems that were trained using extracted GW sentences and the MT system trained on NC collection. We used 10k iterations for our randomization tests. In each iteration we performed permutations between the translation sentences generated by the two MT systems whose statistical difference in performance we evaluate. In Table 5.7, denoted with * are BLEU scores whose statistical significance levels (p-value≤0.001) are above NC.

In order to observe the effect of the two stages of our boostrapping pipeline we ran ablation experiments where we evaluated the performance of an MT system trained on bitext ex-

Training Source	Parallel	Extracted	Test Set	
			News	Europarl
News Commentary (NC)	131k	0	23.75	25.43
Europarl (EP)	1,750k	0	23.91	32.06
Gigaword extracted (GW)	0	926k	24.28*	23.88
NC+GW	131k	926k	24.92*	25.61
EP+GW	1,750k	926k	25.90*	31.59

Table 5.7: BLEU score values on test collections from two different domains using MT systems developed using extracted and parallel sources of training data. * denotes statistical significance level (p-value$\leq$0.001) above NC.

Pipeline Configuration	Extracted	Test Set	
		News	Europarl
OCD	684k	24.00‡	23.84
OCD (deduplicated)	469k	23.84	23.75
OCD+PLTM	926k	24.28*,†	23.88
OCD+PLTM (deduplicated)	588k	24.20*,§	24.67

Table 5.8: BLEU score values of MT systems trained on extracted bitext by OCD alone and with PLTM reestimation along with the deduplication effect. * denotes statistical significance level (p-value$\leq$0.001) above NC. ‡ denotes statistical significance level (p-value$\leq$0.05) above NC. † denotes statistical significance level (p-value$\leq$0.001) above OCD. § denotes statistical significance level (p-value$\leq$0.03) above OCD.

tracted by OCD alone without PLTM reestimation. We also ran a set of experiments where we evaluated the effect of removing duplicate sentence pairs from the two extracted bitext sets. Deduplication was performed by going over the extracted set of English-Spanish sentence pairs and removing the duplicate ones. Table 5.8 gives a summary of these experiments. Denoted with *, ‡, † and § are different statistical significance levels above OCD and NC.

In Table 5.8 we observed that the bitext extracted from the OCD stage alone performed only slightly worse on the News test set. However, this set of extracted sentence pairs only had 70% overlap with the extracted set of parallel sentences from the second stage (OCD+PLTM). We also observe that the deduplication process on the News test set, hurts the performance of the OCD stage alone more than the second stage. On the Europarl test set, however, the deduplication process helped the second stage to improve the BLEU score from 23.88 to 24.67, while it caused slight performance drop for OCD (cf. NC-trained 25.43).

CHAPTER 6

EFFICIENT EVALUATION OF LATENT VARIABLE MODELS OF TEXT

6.1 Introduction

Latent variable models of text are evaluated in one of two different ways - intrinsically and extrinsically. In their original work, Blei et al. [16] used the intrinsic metric of perplexity to showcase the advantages of using latent Dirichlet allocation (LDA) for representing collections over other generative models of text, such as probabilistic latent semantic indexing (PLSI) [51]. This measure was later adopted as a standard evaluation metric for variety of variations of the original LDA model. Perplexity, which originates in information theory, is typically used to determine the right number of topics and the optimal values of other LDA parameters [49]. As an intrinsic metric it is also widely used for evaluating LMs [19] with or without latent variables. For example, when LMs are used in domains such as speech recognition and machine translation (MT) they are evaluated using perplexity which complements extrinsic evaluation measurements such as word error rate (WER) or the bilingual evaluation understudy (BLEU) score. For a language model p and a set of N test sentences $x_i = x_1, x_2, x_3, ..., x_n$ perplexity is defined as $Perplexity(x) = exp\{-\frac{1}{M}\sum_{i=1}^{N}\log p(x_i)\}$ where M is the total number of words in the test set. It is an unsupervised evaluation approach which measures how well does the topic model generalizes in its ability to represent the collection based on the log likelihood of a held-out text which is $\sum_{i=1}^{N}\log_2 p(x_i)$ for a language model p. The exact computation of the log likelihood for topic models is intractable and approximate methods are used instead. Recently, with the work of [125], the computation of the log likelihood in topic models has become more accurate but it still requires traversing over each word in the held-out text and integrating over all possible topic mixtures. For large held-out texts and especially large number of topics (e.g. several thousand), the time that it takes to compute perplexity grows linearly. In such settings, where LDA configurations with large number of topics require continues evaluations, using perplexity for model selection becomes inefficient. While perplexity is useful for relative comparisons, it was recently shown that this intrinsic evaluation does not correlate well with human judgments of topics [24].

Computing perplexity for topic models, assumes that we have already inferred the set of T topic-word distributions φ^t over a training set of D_{train} documents. Using these distributions we then infer the document-topic distributions over the d documents in our held-out set. For a held-out collection of $D_{held-out}$ documents and a collection wide vocabulary of V words, when evaluating topic models, perplexity is computed using the following formula:

$$Perplexity(D_{held-out}) = \exp\{-\frac{\sum_{d=1}^{D_{held-out}} \log(\sum_{n=1}^{N_d}\sum_{t=1}^{T}(\theta_{dt}\varphi_{tn}))}{\sum_{d=1}^{D_{held-out}} N_d}\} \qquad (6.1)$$

Where φ_{tn} is the inferred probability of word n in topic t and θ_{dt} is the probability of assigned topic t to held-out document d while N_d is the total number of words in that document.

Unlike perplexity, most recently introduced intrinsic evaluation approaches for topic models try to measure the coherence of the generated topic-word distributions. Newman et al. [92], for instance, explored various topic coherence evaluation methods and discovered that pointwise mutual information (PMI) computed over word co-occurrence in all Wikipedia articles using a sliding window of ten words give a good estimate of the topic quality as judged by humans. Aletras & Stevenson [2] obtained their best topic coherence estimates by computing similarity across context feature representation of the most probable words for a topic. While coherence metrics successfully predict human judgments, Stevens et al. [110] showed that these intrinsic metrics are not good indicators of the effectiveness of using topics as document feature representation for classification tasks. Moreover, these intrinsic evaluation methods do not offer us the insight on how well would the topic model perform on a specific task.

Extrinsic evaluations, on the other hand, involve measuring the performance of the model on a specific task. When using LDA to represent documents for Information Retrieval (IR), for instance, precision, recall, mean average precision (MAP), normalized discounted cumulative gain (NDCG) and so on are used to measure the effectiveness of the end-to-end system [128]. Extrinsic evaluations give us better insight into models' ability to represent the document collection in the context of a real world task. Evaluations on extrinsic tasks, such as retrieving similar documents, use a test collection of query documents along with their sets of query relevant i.e. similar documents. For a test collection of q queries and n documents, performing the evaluation task involves three steps: (1) computing similarity between qn document pairs - $O(qn)$; (2) sorting each query ranked list of documents - $O(qn \log n)$; (3) and traversing q sorted ranked lists and determining the location of the relevant documents. The time complexity of the latter step is also linear in the number of queries q and the rank of the last relevant document and it has a worst-case complexity of $O(qn)$. While these steps are often trivial to perform, in settings where topic models need to be continuously evaluated using large evaluation sets and large number of topics their computation becomes inefficient.

6.1.1 New evaluation measures

In this chapter we propose two new evaluation measures for latent variable models which are more efficient to compute compared to existing topic models' intrinsic and extrinsic evaluation measures on document comparison tasks. Both approaches analyze the relationship between histograms computed over the relevance scores of query relevant and non-relevant documents. Our first extrinsic metric, called Distributional Overlap (DO), predicts the performance by analyzing the distribution of pairwise distances between topic model document representations to separate a small number of short distances from the vast majority of longer distances. Our second approach, called Histogram Slope Analysis (HSA) measures the slope of the curve obtained by dividing the histograms of relevant and non-relevant documents in log space.

Over the proposed metrics we first conduct extensive analysis of their performance across different model configurations on two document similarity tasks: prior-art patent search and the cross-language information retrieval (CLIR) task of finding document translation pairs. We demonstrate that unlike MAP or perplexity, DO and HSA are more efficient to compute. We also show that unlike perplexity, HSA has a very high linear and rank correlation with existing extrinsic evaluation measures.

We then extend our proposed metrics to evaluate the performance of different IR models. Linear and rank correlation analysis are performed on four ad-hoc web-retrieval tasks and on a meta-evaluation of the ranked lists of ten Text REtrieval Conference (TREC) tracks from the past ten years. Across all evaluation tasks we show that HSA has a very high correlation with the actual performance of the retrieval model as measured by common IR metrics such as MAP and NDCG. However, unlike these metrics, computing HSA does not require sorting and traversing a ranked list for each query result, thus making it more efficient to compute.

We start off by analyzing the relationship between the similarity values of similar and dissimilar document pairs which we then use to introduce DO and HSA. We conclude this chapter by presenting a variation of HSA, called random HSA (rHSA), which we use to automatically evaluate retrieval models in large scholarly publication collections. Unlike typical evaluation settings that mostly rely on human annotated evaluation sets we use download logs to automatically generate pseudo-relevant set of similar document pairs. Rather than computing HSA over scores of relevant and non-relevant documents we automate the evaluation process by analyzing the histograms of scores over consecutively downloaded (CD) and randomly generated (RG) document pairs.

6.2 Histogram Analysis for Latent Variable Models of Text

6.2.1 Histograms

Histograms are commonly used as means to summarize distributions and obtain visual information over the frequency of occurrence of values in a given range. They partition the total range of values into i bins and then count the number of times n_i values fall into bin i. Dividing the number of observations in each bin by the total number of observations and the width of the bin, the histogram counts could be turned into a normalized probability distribution. This allows the histograms to be used as simple nonparametric approach for estimating the underlying probability density [12].

6.2.2 Similar vs. Dissimilar Document Pairs

As we pointed out in Chapter 1, finding documents that are topically similar involves representing them in a shared space and comparing their representations using a particular metric. Regardless of the representation used, within the shared space, similar documents tend to be positioned close to each other versus dissimilar documents that are further apart. Across different model configurations the shared space could vary based on the number of dimensions. For example, in the shared topic space, the number of topics define the dimensionality of the space. When retrieving similar documents different model configurations give different performance which is directly related to how the model positions documents in the shared space.

We hypothesized that one of the aspects of the model strength should be defined as how well does the model position documents that are similar compared to dissimilar documents. More powerful models should create a shared space such that for a given document d, represented as a topic vector θ, its topically similar documents would be in its nearest proximity while the topically dissimilar documents would be positioned further away. Creating a histogram of all divergence values provides us with an insight as to how distributed these values are. From the histogram plot, we would expect to see a bimodal distribution, where the divergence values of all topically similar documents originate from the first mode, which for example is in the lower range of Jensen-Shannon (JS) values, and all topically dissimilar documents reside in the second mode that assumes a larger set of JS values. As an example of this discussion, we show in Figure 6.1 a joint histogram plot of two sets of 10k JS divergence values computed as all-pairs comparison across 100 vector pairs sampled from two different Dirichlet distributions. Both vector sets were 50 dimensional and the Dirichlet distributions were configured with symmetric hyperparameter $\alpha = 0.001$ and 10. Since in Dirichlet distributions, the hyperparameter defines how sparse the distribution is, across the dimensions for small values of α we obtain very sparse vectors. This emulates documents where within each document only few topics dominate which in turn makes the JS divergence values very large as documents tend to be topically dissimilar simply due to the fact that there would not be that many topic hits across two documents. Large hyperparameter values helps us generate synthetic documents where topic proportions are spread through most of the topics thus creating more topically similar document pairs.

6.2.3 Evaluating Topic Models Through Histogram Analysis

Our approaches for evaluating latent variable models of text are solely based on the notion that when retrieving similar documents, the power of the retrieval model should be such that dissimilar documents would be assigned with lower similarity while similar documents should have higher similarity values. Focusing on the precision aspect of the model, more powerful models would have higher precision which implies that similar documents would need to have higher similarity values in order to be ranked higher. As the rank of the similar documents increases we would expect to see their mass on the histogram of similarity values to shift towards minimum values. Same notion applies for dissimilar doc-

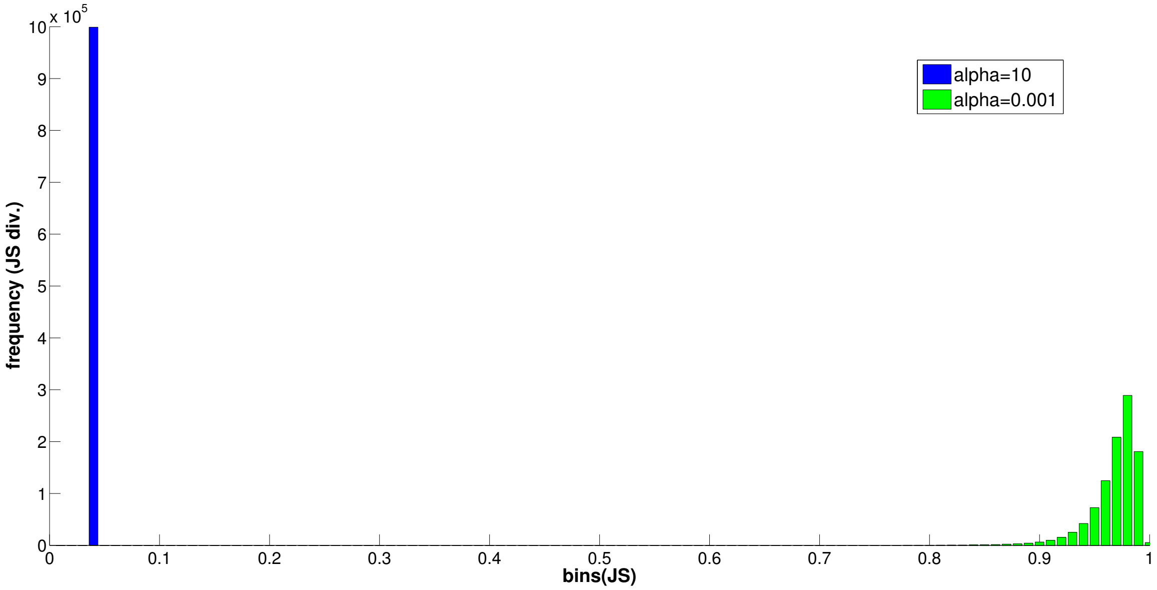


Figure 6.1: Joint histogram plot of two sets of JS divergence values. Each set was obtained through all-pairs similarity computation across 100 samples drawn from 50 dimensional Dirichlet distributions. The two Dirichlet distributions were defined with symmetric hyperparameter α set to 10 and 0.001. Histogram frequency values were normalized to depict the difference in shapes between the two distributions.

uments – as the precision of the model increases we should expect to see their similarity values to decrease or remain in the same region which in turn implies that their histogram mass would shift towards lower values or assume the same range of values across different models therefore reducing the distributional overlap. Within the topic space, i.e. the probability simplex, this distributional overlap is composed of document pairs whose topical relatedness is mixed – document pairs are topically related and unrelated.

Reflecting on the position of the documents in the shared space, more powerful models should position similar documents close to each other while dissimilar documents should be placed further apart thus making the volume of the distributional overlap region smaller. We use the volume of the distributional overlap between the histogram of topically similar and dissimilar documents to define DO.

Computing the ratio between the histograms of the similar and dissimilar document pairs across all bins gives us the proportion of similar documents that we would expect to find at certain similarity values. In the shared space, for a given query document, this ratio gives us the proportion of similar documents that we would expect to find at certain distances. Across different models, as the performance of the model improves the proportions of similar documents found close to the query document should increase thus making the slope of the histogram analysis steeper going down from left to right. We define HSA as the absolute value of the slope.

Unlike existing extrinsic evaluation measures, when computing DO and HSA we only need a fixed sample of relevant and non-relevant documents rather than computing similarity across the entire collection.

6.3 Previous Work in Information Retrieval

In their seminal work, Jardine and van Rijsbergen [57] used histograms to define one of the prominent concepts in information retrieval – the cluster hypothesis which states that "closely associated documents tend to be relevant to the same requests". The hypothesis introduced the cluster based retrieval [103] and many of its variants [28, 104].

Since their introduction, various cluster-based retrieval approaches have been proposed [47, 77, 117]. Jardine and van Rijsbergen [57] proposed the cluster hypothesis test in order to measure the potential of the cluster based retrieval on a collection of documents. For a given collection and a set of queries (i.e. "requests" [57]) the test compares the histogram of the similarity values computed between all-pairs of query relevant documents and the histogram of the similarity values computed between the query relevant and non-relevant documents. Aside from the original cluster hypothesis test, researchers have proposed other measures of the cluster hypothesis [108, 122]. While different measures exist for the cluster hypothesis they have not found their use in evaluating the performance of different retrieval models and especially latent variable models of text. Our evaluation approaches were developed independently from the cluster hypothesis. And while developed independently, DO metric is in line with the cluster hypothesis. In our approach we analyze the similarity of relevance scores computed between the query and the query relevant and non-relevant documents.

An earlier work that uses the observation of the distributions of scores over the retrieved relevant and non-relevant documents is the work by Swets [113]. In this work, Swets [113] introduced the relative operating characteristic (ROC) which is a recall-fallout curve. In his analysis Swets characterizes the IR system by defining two conditional probabilities. Swets treats the proportion of non-relevant and retrieved documents as an estimate of the conditional probability that a non-relevant document will be retrieved while the recall is referred to as the conditional probability of a relevant document to be retrieved. ROC is then obtained by observing the pdf of the two probabilities at certain cutoff values which are equivalent to the different ranks in the results list.

6.4 Performing Histogram Analysis

Given a test collection of k query documents $Q = D_{q1}, D_{q2}, D_{q3}, ..., D_{qk}$ along with their query relevant $R_{D_{qi}} = r_1^i, r_2^i, r_3^i, ..., r_m^i$ and non-relevant $NR_{D_{qi}} = nr_1^i, nr_2^i, nr_3^i, ..., nr_n^i$ documents, computing DO and HSA requires two steps. For both metrics we first obtain histograms: Using document's topic model representation we compute similarity across the query relevant and non-relevant documents and obtain similarity values using a similarity metric f: $Sim_R = Score_f(D_{qi}, R_{D_{qi}})$ and $Sim_{NR} = Score_f(D_{qi}, NR_{D_{qi}})$. We use these similarity values to obtain histograms for the two sets of similarity scores. Histograms are computed using a set of B equally spaced bins:

$$H_R = [h_1^r, h_2^r, h_3^r, ..., h_{|B|}^r] \tag{6.2}$$

$$H_{NR} = [h_1^{nr}, h_2^{nr}, h_3^{nr}, ..., h_{|B|}^{nr}] \tag{6.3}$$

Where h_b^r and h_b^{nr} are the counts of the number of times relevant and non-relevant similarity values fall within the bin centered at b: $h_b^r = \#(V_R \in b)$, $h_b^{nr} = \#(V_{NR} \in b)$.

In most real world collections the portion of documents that are topically similar (relevant) to the query is significantly smaller than topically dissimilar (non-relevant) documents. When creating the histograms we normalize the disproportion by using log scale for the frequency axis. In addition, for our analysis we only consider bins where both histograms contain non zero counts. We call this set of bins a supported set $B' = \{b' : h_{b'}^r, h_{b'}^{nr} \neq 0\}$.

6.4.1 Computing DO

For DO, in the second step we simply measure the volume of the overlap region between the two histograms:

$$DO = \sum_{b' \in B'} \log(\min(h_{b'}^r, h_{b'}^{nr})) \tag{6.4}$$

In Figures 6.2 and 6.3 we illustrate examples of computing DO across two different tasks and two different families of topic models. More specifically, in Figure 6.2 we evaluate four different LDA configurations with number of topics set to T=50, 100, 200 and 500 on the patent retrieval task which we detailed in Section 3.7.2.

Figure 6.3 shows the joint histogram plots across four different document similarity models on the task of finding document translation pairs which was detailed in Section 4.4.1. On this task the polylingual topic model (PLTM) was also configured with number of topics set to T=50, 100, 200 and 500.

6.4.2 Computing HSA

For HSA, we compute the log ratio of the two histograms for the support bins $b' \in B'$ where they are both observed $O_{b'} = \log(\frac{H_{R_{b'}}}{H_{NR_{b'}}})$. We then fit a linear function $O_{b'} = \alpha + \beta b' + \epsilon_{b'}$ using linear least squares regression. HSA is defined as the estimated slope:

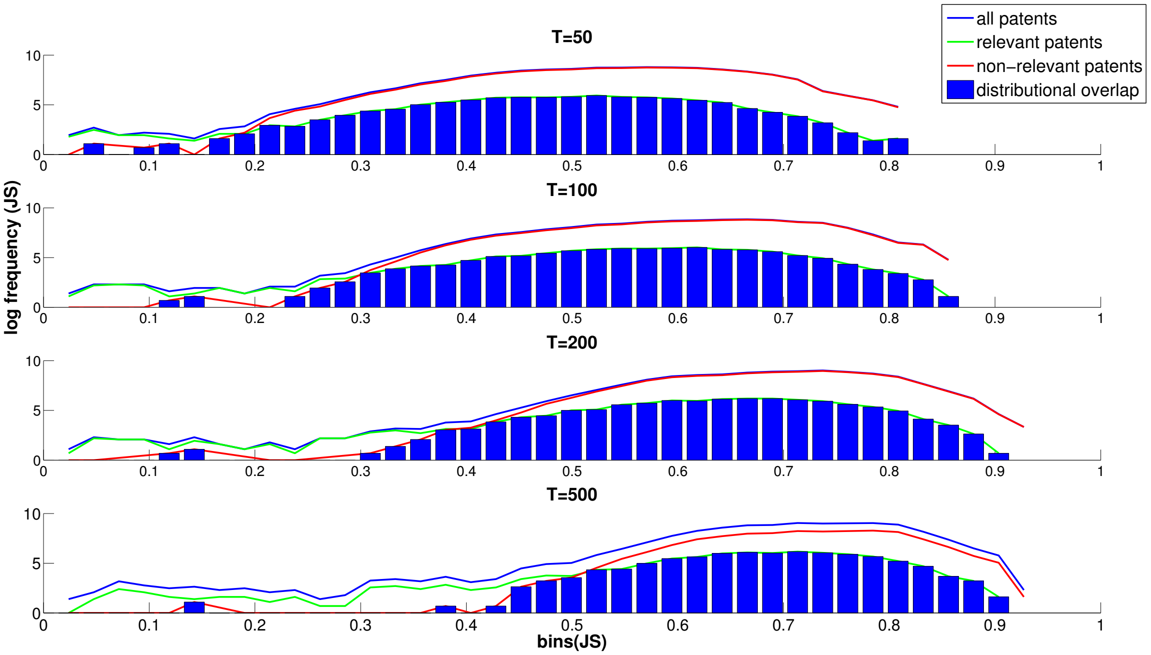


Figure 6.2: Computing DO example: Evaluation of LDA models on the task of retrieving related patents.

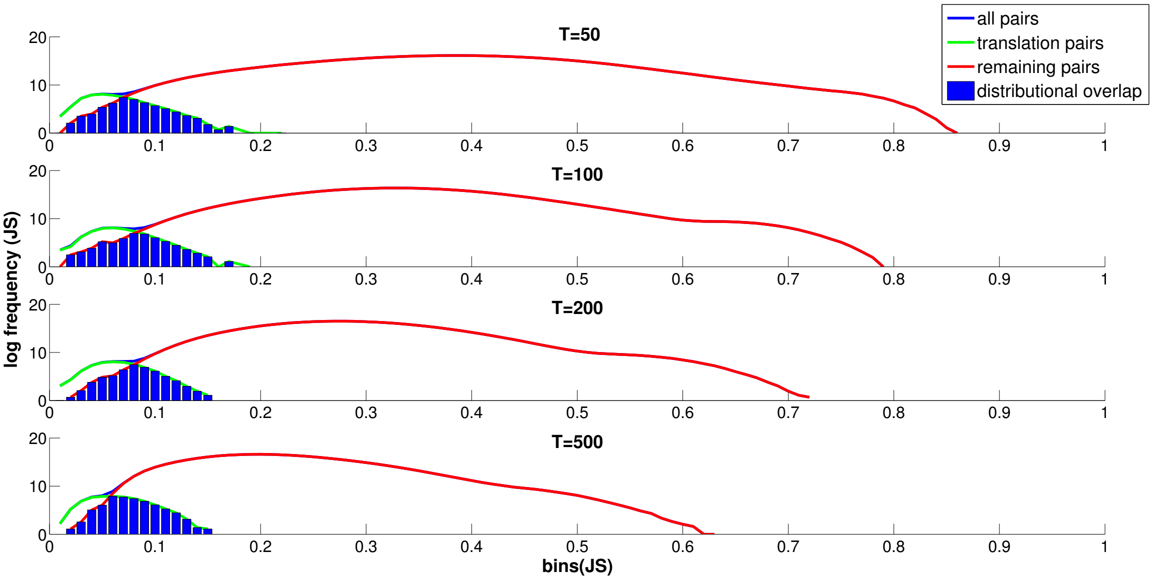


Figure 6.3: Computing DO example: Evaluation of PLTM models on the task of retrieving document translation pairs.

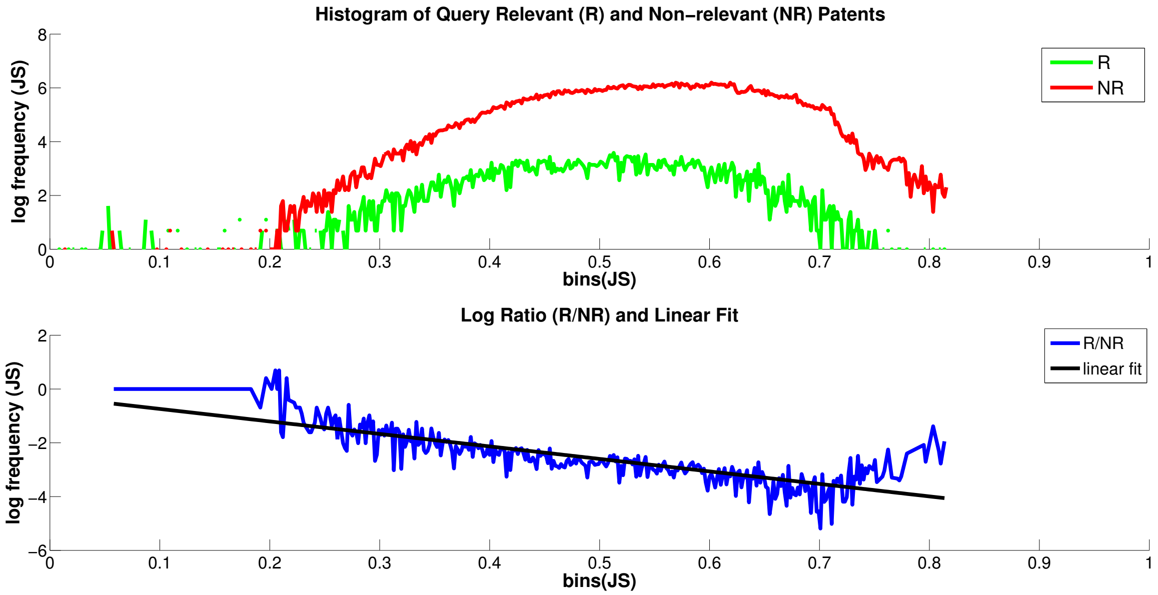


Figure 6.4: Computing HSA example: Evaluation of LDA model with T=50 on the task of retrieving related patents. HSA measures the slope of the linear fit.

$$HSA \ = \ \hat{\beta} = \frac{\sum_{b' \in B'} (b' - \bar{B}')(O_{b'} - \bar{O})}{\sum_{b' \in B'} (b' - \bar{B}')^2} \tag{6.5}$$

Figure 6.4 shows examples of computing HSA to evaluate an LDA model with T=50 on the task of retrieving related patents.

6.5 Predicting Latent Variable Model Performance

One of the purposes of developing DO and HSA was to be able to evaluate and predict the performance of latent variable models of text. To demonstrate this ability we compared ranked lists of topic models performance generated using DO and HSA with ranked lists obtained using existing intrinsic and extrinsic evaluation measures such as perplexity and MAP. The predictive power of DO and HSA was evaluated by performing linear correlation analysis using Pearson correlation coefficient (R). Using Spearman's rank correlation coefficient (ρ) we computed correlation between the ranked list of models' performance sorted by existing metrics and the ranked list obtained using DO and HSA. Our correlation analysis were performed on a monolingual and on a multilingual task: retrieving related

Model	P@1	DO	HSA	Perplexity
PLTM T=50	0.943	235.51	-108.30	4628.00
PLTM T=100	0.985	221.80	-109.09	1692.00
PLTM T=200	0.994	201.84	-138.77	4419.00
PLTM T=300	0.994	224.91	-149.95	1405.01
PLTM T=400	0.995	244.76	-161.70	1345.80
PLTM T=500	0.993	246.83	-167.19	1323.14
PLTM T=700	0.990	282.07	-148.47	1295.92

Table 6.1: Absolute P@1, DO, absolute HSA and perplexity values computed across different PLTM model configurations on the task of finding document translation pairs.

Correlation	MAP(LDA)			P@1(PLTM)		
	DO	HSA	Perplexity	DO	HSA	Perplexity
R	-0.75	0.92	-0.84	0.00	0.71	-0.63
ρ	0.64	0.93	0.89	0.11	0.64	0.25

Table 6.2: Evaluating latent variable model performance using DO, HSA and perplexity: Pearson (R) and Spearman's (ρ) coefficients computed over MAP and P@1.

patents and the CLIR task of finding document translation pairs. Across both tasks we used 7 different model configurations. On the patent retrieval task we evaluated LDA with number of topics set to T=50, 100, 200, 500, 1k, 2k and 5k. On this task LDA models were originally evaluated using MAP. While on the CLIR task PLTMs were configured with T=50, 100, 200, 300, 400, 500 and 700. PLTM models were evaluated based on the number of times the model ranked true translation pairs as the most similar, i.e. precision of the top rank (P@1). As an example, in Table 6.1 we show absolute P@1, DO, HSA and perplexity values computed across different PLTM configurations on the task of finding document translations.

Table 6.2 shows the correlation coefficients computed between our proposed evaluation measures and MAP and P@1. Since topic models are typically evaluated intrinsically using perplexity on held-out data, correlation coefficients were also computed for this metric in order to compare its predictive power with DO and HSA.

Model	MAP[s]	DO[s]	HSA[s]	Perplexity[s]
LDA T=50	288.1	28.2	28.2	11.1
LDA T=100	224.8	33.0	33.0	27.2
LDA T=200	240.1	47.5	47.6	53.6
LDA T=500	345.9	99.2	99.2	143.1
LDA T=1000	405.6	176.3	176.3	3166.7
LDA T=2000	559.5	333.6	333.6	32930.0
LDA T=5000	1037.4	816.2	816.2	46340.0

Table 6.3: Absolute time required for computing MAP, DO, HSA and perplexity when evaluating LDA models on the task of retrieving related patents.

Across both tasks, HSA offers higher linear and rank correlation compared to perplexity. Unlike HSA, DO only yields good rank correlation on the patent search task. On average all metrics have higher correlation coefficients with MAP compared to P@1. The high HSA correlation in practice allows to predict the performance of the document similarity model by computing similarity only on the set of relevant and non-relevant patents. This in turn makes the evaluation process more efficient since it does not require processing all patents in the collection.

Table 6.3 shows the absolute time required for computing MAP, DO, HSA and perplexity when different topic model configurations are evaluated on the patent retrieval task. The slight difference in the computation time between DO and HSA comes from the second step of computing these metrics. For example, on our computing platform, for DO computing the volume required $\sim$10ms while computing the slope takes $\sim$60ms.

Compared to MAP and perplexity, DO and HSA are most efficient to compute. Due to the nature of the P@1 evaluation metric, which only requires checking the document at rank one, the computation time for MAP, DO and HSA, while being different for each metric, is equal across the different model configurations. On this task computing DO and HSA gives us a relative speed improvement of 5.12 times over MAP. In the case with the patent retrieval task the computation time for perplexity grows linearly with the number of topics.

6.6 Beyond Latent Variable Models: Evaluating IR Models

IR models are typically evaluated extrinsically using ranking metrics, such as MAP and NDCG which requires computing, in the worst case, the relevance score of each query against each member of a collection of n documents. In practice, of course, most retrieval systems achieve vastly better performance by: using inverted indices to avoid scoring documents missing some, or any, features; performing lossless or lossy pruning of posting lists; keeping track of only the top k documents for each query. For large collections, however, building even one index can be costly, and evaluating multiple models may require the creation of multiple indices. Reindexing can become even more common when working with continuous representations, as in image retrieval or in using topic models for text.

Before building new indices and tuning other efficiency parameters of an IR system, researchers may want some validation that a new feature, such as skip n-grams or LDA topics, will positively impact effectiveness on the target task. Rescoring ranked lists generated by a baseline system provides one such check on model validity. However, new models will be most useful when they identify relevant results outside the output of the baseline system.

To alleviate these drawbacks we extend the use of DO and HSA to evaluating common IR models. Instead of using document similarity values of relevant and non-relevant documents, when evaluating common IR models, we compute DO and HSA using the relevance scores of query relevant and non-relevant documents. Unlike latent variable models of text, where similarity is computed using JS divergence or He distance, across different retrieval models relevance values have different scales. In order to compare and rank the performance of different models, relevance values generated by each model are normalized to a range of [0, 1].

Figures 6.5 and 6.6 show examples of computing HSA across two retrieval models on the TREC Web Track 2009. In this example we compute HSA over the relevance scores of query relevant and non-relevant web pages obtained using a Query Likelihood (QL)

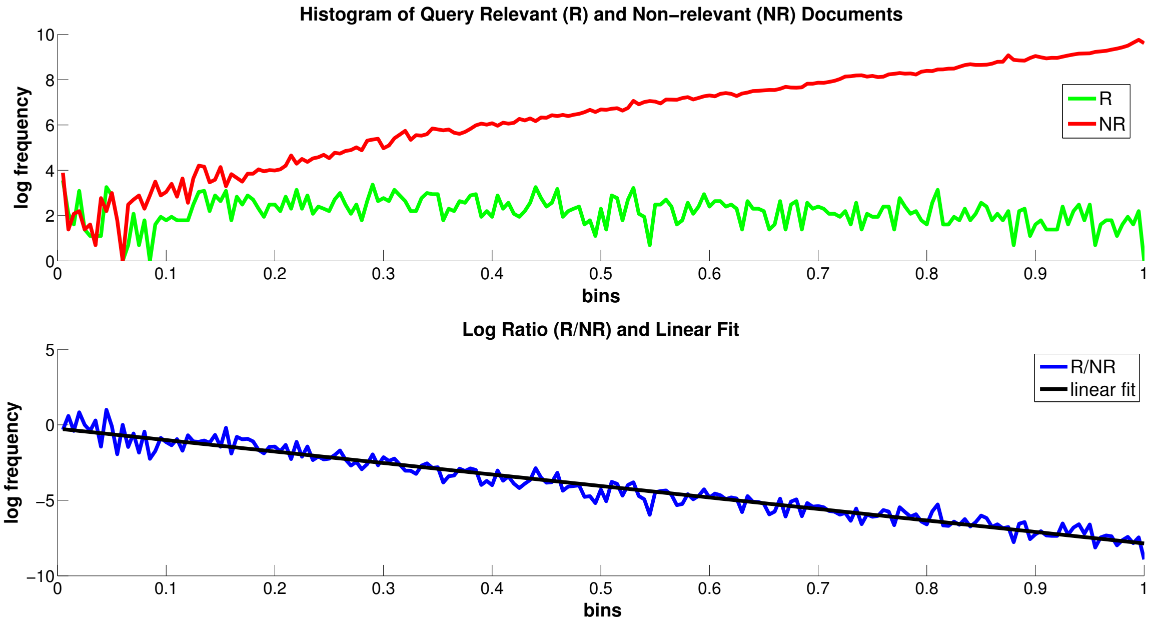


Figure 6.5: Computing HSA example: Evaluation of a QL model on the Text REtrieval Conference (TREC) Web Track 2009. NDCG(QL)=0.227. HSA measures the slope of the linear fit.

.

model configuration, which gave the worst NDCG, and the sequential dependence model (SDM), which achieved the highest NDCG. Details of this experimental setup are presented in Section 6.6.1.1. The first two subplots show the log histograms of the query relevant (R) and non-relevant documents (NR). The bottom two subplots show the log ratio of the empirical distributions of R and NR scores $O_{b'}$ along with the linear fit $\hat{O}_{b'}$ whose absolute slope value defines HSA.

6.6.1 Correlation with Existing IR Metrics

In this section we demonstrate the generality of our evaluation approach across different IR tasks, models and scoring functions. For that purpose we use two experimental setups: (1) set of ad-hoc retrieval tasks where the scoring function represents the relevance score of the document given the query and (2) a meta-evaluation of ranked lists submitted to ten TREC tracks in the past ten years where we use the rank of the retrieved documents as relevance values.

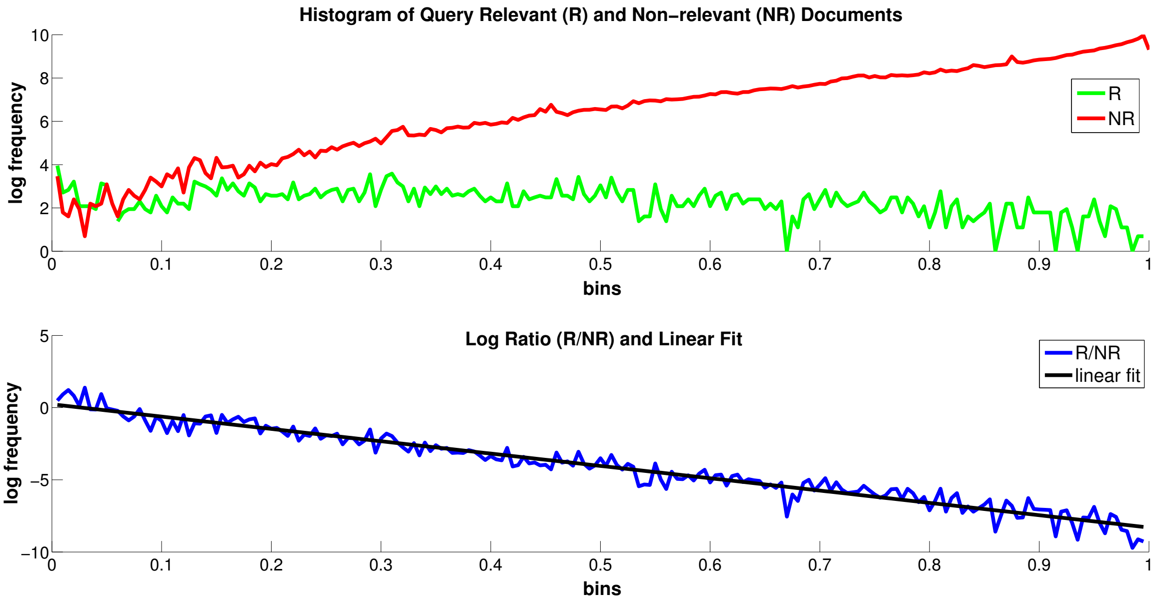


Figure 6.6: Computing HSA example: Evaluation of a SDM model on the TREC Web Track 2009. NDCG(SDM)=0.293. HSA measures the slope of the linear fit.

6.6.1.1 Ad-Hoc Retrieval Tasks

We used query sets from four previous TREC Web Tracks (2009-2012). Experiments were performed on the ClueWeb09 Category-B with spam filtering (a threshold of 60 using the Waterloo spam scores) collection [26]. We evaluated 5 different retrieval models: Relevance Model (RM), SDM and three QL models with various parameter settings. In our experiments we used the open source retrieval engine Galago [42]. Tables 6.4 and 6.5 show the linear and rank correlation coefficient values for DO and HSA computed across three IR metrics: MAP, precision of the top 10 (P@10) and NDCG. When computing all evaluation measures we only used the top 10k retrieved documents and their relevance scores.

Across the three IR metrics and evaluation sets HSA has a high linear and rank correlation. While DO has a high linear correlation, its rank correlation is negative, since a large overlap between the distributions of relevant and non-relevant documents is undesirable.

Web Track	HSA			DO		
	MAP	P@10	NDCG	MAP	P@10	NDCG
2009	0.80	0.86	0.79	0.86	0.84	0.87
2010	0.99	0.98	0.99	0.88	0.86	0.91
2011	0.97	0.95	0.99	0.79	0.74	0.83
2012	0.78	0.76	0.76	0.58	0.67	0.60

Table 6.4: Evaluating ad-hoc retrieval models using DO and HSA: Pearson (R) coefficient computed across MAP, P@10 and NDCG.

Web Track	HSA			DO		
	MAP	P@10	NDCG	MAP	P@10	NDCG
2009	0.80	0.90	0.90	-0.60	-0.70	-0.70
2010	0.90	0.80	0.90	-0.60	-0.70	-0.60
2011	1.00	0.90	1.00	-0.90	-0.80	-0.90
2012	0.50	0.80	0.50	-0.30	-0.70	-0.30

Table 6.5: Evaluating ad-hoc retrieval models using DO and HSA: Spearman's (ρ) coefficient computed across MAP, P@10 and NDCG.

6.6.1.2 TREC Ranked Lists

In our correlation analysis so far we have computed DO and HSA using a relatively large set of query based relevance scores. For example on the patent retrieval task we used 70k patents. The CLIR task was done on $\sim$14k Spanish Europarl speeches. On our set of ad-hoc retrieval tasks we used the top 10k retrieved ClueWeb documents. Typical IR system in many real world scenarios are actually configured to return a relatively small percentage of all the documents in the collection. This is certainly the case across different TREC tracks [8] where submitted ranked lists by participants typically consist of 1k retrieved documents. In most of the cases relevance values found in these ranked lists are a very small subset of the relevant documents across the whole collection.

We created our second experimental setup to measure the correlation when DO and HSA are computed using such small sample sets of relevance values. More specifically, we used ranked lists submitted on ten TREC tracks from the time period between 2004 and 2013. For each year's TREC conference we randomly choose a track and from the selected

TREC Track	HSA			DO		
	MAP	P@10	NDCG	MAP	P@10	NDCG
Microblog ’13	0.98	0.95	0.93	-0.82	-0.80	-0.70
Medical ’12	0.90	0.84	0.94	0.30	0.10	0.50
Web ’11	0.75	0.86	0.79	0.80	0.51	0.90
Session ’10	0.82	0.75	0.84	0.46	0.27	0.68
Chemical ’09	0.85	0.95	0.82	0.66	0.84	0.65
Enterprise ’08	0.93	-0.14	0.93	0.99	0.21	0.99
Million ’07	0.85	0.93	0.88	0.87	0.89	0.94
Terabyte ’06	0.97	0.98	1.00	0.94	0.97	0.97
Robust ’05	0.75	0.60	0.74	0.82	0.83	0.91
Web ’04	0.87	0.63	0.88	-0.41	-0.02	-0.36

Table 6.6: Evaluating TREC track submissions using DO and HSA: Pearson (R) coefficient computed across MAP, P@10, and NDCG.

track we randomly chose 7 submitted ranked lists. In TREC tracks relevance scores across submitted ranked lists are generated using different relevance functions whose information is not readily available. We therefore use the normalized values of the document ranks when computing DO and HSA. Using document ranks rather than relevance scores also helps alleviate formatting issues and missing scores which occur in some ranked lists.

Tables 6.6 and 6.7 show the linear (R) and rank correlation (ρ) coefficients computed across various TREC tracks from the past ten years. When HSA is computed on a relatively small subset of relevance values, which are generated by normalizing the document ranks, we still achieve high linear and rank correlation.

6.7 Evaluating Document Similarity Models using Download Logs

Evaluating retrieval models on document similarity tasks, as in the case with most retrieval models, heavily depends on having a test set of human annotated relevant documents. Human annotation of relevant documents is a time consuming processing and also costly to perform. One way to alleviate the human annotation cost is to annotate a union of the top n retrieved documents from multiple retrieval systems rather than to annotate the whole collection or individual retrieval models. In the past this approach has been widely

TREC Track	HSA			DO		
	MAP	P@10	NDCG	MAP	P@10	NDCG
Microblog ’13	0.82	0.79	0.86	-0.53	-0.64	-0.51
Medical ’12	0.96	0.82	0.96	-0.32	-0.14	-0.32
Web ’11	0.71	0.86	0.75	-0.68	-0.43	-0.71
Session ’10	0.71	0.86	0.64	-0.54	-0.32	-0.75
Chemical ’09	0.82	0.86	0.79	-0.96	-0.89	-0.93
Enterprise ’08	0.86	-0.14	0.75	-0.86	-0.04	-0.89
Million ’07	0.86	0.68	0.86	-0.89	-0.89	-0.89
Terabyte ’06	0.86	0.43	0.86	-0.18	-0.29	-0.12
Robust ’05	0.78	0.57	0.79	-0.67	-0.54	-0.68
Web ’04	0.68	0.39	0.71	0.00	-0.07	-0.07

Table 6.7: Evaluating TREC track submissions using DO and HSA: Spearman’s (ρ) coefficient computed across MAP, P@10, and NDCG.

used across different TREC tracks [8]. Another approach for annotating documents that has gained momentum in the past several years is the use of crowdsourcing [4, 60, 72]. While it offers the potential of reducing the time and cost of human annotation this approach requires building a proper infrastructure to task the annotation and to evaluate its quality. When dealing with new document collections developers are often faced with the dilemma when trying to decide on what models to be initially used on tasks such as retrieving similar documents. In absence of human annotated test collections a good alternative for evaluating models is a creation of pseudo-relevant set of similar documents from existing human generated information such as query log information.

In this section, we present a novel approach for generating a pseudo-relevant set of similar documents that could be used to evaluate document similarity models. More specifically, we explore utilizing information on downloaded documents from a scholarly IR system logs to automatically generate a pseudo-relevance set of similar documents. Unlike previous approaches for creating pseudo-relevant sets that utilize query log information, such as clickthrough data [58], in our approach we utilize information on downloaded documents i.e. documents that the user has downloaded in a given time window. We do however follow the same assumption of relevance that is present when using clickthrough data which is that the user is more likely to click on a document that is more relevant to the query than a random one. In our case the assumption is that the user is more likely to consecutively download a document more similar to the previously downloaded document than a random one. With this assumption we treat documents that have been downloaded consecutively after a given document as pseudo-relevant. We should point out that various collaborative filtering approaches for recommender systems [102] use the same notion of similarity. Product recommender systems (e.g. [52, 106]), for example, use information on consecutively purchased products and library recommender systems (e.g. [74]) use information on consecutively loaned books.

To further automate the process of evaluating document similarity models, in this section we also derive rHSA. Unlike HSA that uses relevance scores computed over the set of query relevant and non-relevant documents, when computing rHSA we use document similarity scores computed over automatically generated sets of CD and RG document pairs. Furthermore when computing rHSA we also model the errors in the observed similarity values which was omitted when computing the HSA metric due to efficiency. Modeling the error helps us get more accurate values of the histogram slopes. Similar to HSA, rHSA, as we will show later in this section, achieves very high correlation with IR metrics, such as MAP.

6.7.1 Automatic Construction of Evaluation Sets

We introduce our method for automatic construction of test collections for evaluating document similarity models using download logs from the SAO/NASA Astrophysics Data System (ADS) [68] which is a scholarly IR system that covers scholarly literature in the fields of astronomy, astrophysics and physics. ADS contains a bibliographic database of over 11 million records. Scholarly IR systems cover many scientific disciplines. They either focus on one or more closely related scientific fields (e.g. computer science (ACM Digital Library), biomedicine (PubMed) and electrical engineering (IEEE Xplore)) or they encompass a broader and not closely related range of disciplines (e.g. Google Scholar, Thomson Reuters Web of Science, Elsevier ScienceDirect, etc.). These and many other scholarly IR systems, which are mostly used by experts in the particular field, contain rich user community logs including information of downloaded articles which what we explore in our approach. While ADS covers a closely related set of scientific disciplines we are sure that the methods presented here could be applied to any scholarly IR system.

Across many scientific journals that are indexed by ADS, we mined download logs of articles from Astrophysical Journal (ApJ) which is the most widely cited journal in the domain of astrophysics. Logs were mined from the time period between January 2010 and January 2013. Within this period we extracted a total of $\sim$23m downloads. Across them there were $\sim$1.7m unique downloads. We selected CD document pairs (A_i, B_i) that occurred within a time window of more than ten seconds and less than one hour. With these heuristics we attempted to eliminate document download pairs that are too short to be generated by humans and too long to belong to the same query session. We obtained a total of 3,898,994 CD document pairs which contained 90,395 unique documents. With the extracted document pairs we created two test sets which we describe in the following two sections.

6.7.1.1 Set of CD and RG Document Pairs

From the above total set, we randomly chose one million CD document pairs. Using the 90,395 unique documents we randomly generated one million document pairs. In the following sections we refer to this set as the one million set (1M) and we use it to introduce rHSA which allows us to predict the performance of different document similarity models.

6.7.1.2 Query Set

Our second test set consists of 100 query documents. These documents were extracted by computing statistics across the 90,395 unique documents A_i in terms of the number of times unique documents B_i were downloaded after document A_i: $|(A_i, Unique(B_i)|$. In our query set each document Q_i had more than 100 unique succeeding downloads across all extracted document pairs $Unique(B_i) \geq 100$.

Our pseudo-relevant set of documents, for each of the query documents Q_i, consists of documents that were consecutively downloaded after the query document more than 10 times: $PR(Q_i) = B_i$, where $(Q_i, |B_i|) \geq 10$. We use this test set, which we refer to as the 100 query set (100q), to perform a more conventional ranking evaluation which include computing MAP across different document similarity models.

6.7.2 Computing rHSA

Figures 6.7 and 6.8 show examples of computing rHSA for evaluating two model configurations across two families of similarity models. In addition to the LDA model, we analyze and present rHSA using the term frequency–inverse document frequency (tf–idf) model which represents documents in the vector space and measures similarity using the Cosine (Cos) distance. Similar to HSA, computing rHSA requires two steps. Unlike HSA, where in the first step, we obtained the log histograms of the similarity values computed across the relevant and non-relevant documents, when computing rHSA we use similarity values computed across the automatically generated sets of CD and RG document pairs. In the first subplot of each figure we show log histogram over the normalized Cos distance (for the tf–idf models) and JS divergence (for the LDA models) values computed across the 1M CD and RG document pairs.

The second step of computing rHSA follows the same procedure as in the case of HSA. In this step we divide CD and RG, fit a linear function using linear least squares regression, and compute the slope of the fit whose absolute value defines rHSA. Results of this step are shown in the second subplots of each figure.

Computing rHSA also involves modeling errors in the observed similarity value which was omitted in the HSA metric due to computational efficiency. For each histogram bin we model the error in the observations of the similarity values using Poisson distribution which helps us estimate the observations error variance. For the Poisson distribution the variance is equivalent to the mean which in our case are the observed counts. Using Gaussian distribution we model the error in the observed similarity values for each bin. The variance of the Gaussian distribution is computed from the similarity values that fall within the bin. We then propagate the modeled observation errors through the log division across both histogram axis. For the error in the observations we have: $\sigma^2_{LogDiv_y} = \frac{\sigma^2_{Div_y}}{Div^2_y}$, where $\sigma^2_{Div_y} \simeq Div^2_y \left(\frac{\sigma^2_{CD_y}}{CD^2_y} + \frac{\sigma^2_{RG_y}}{RG^2_y} \right)$ assuming that for each bin there is no correlation between the fluctuations in the similarity value observations and their counts.

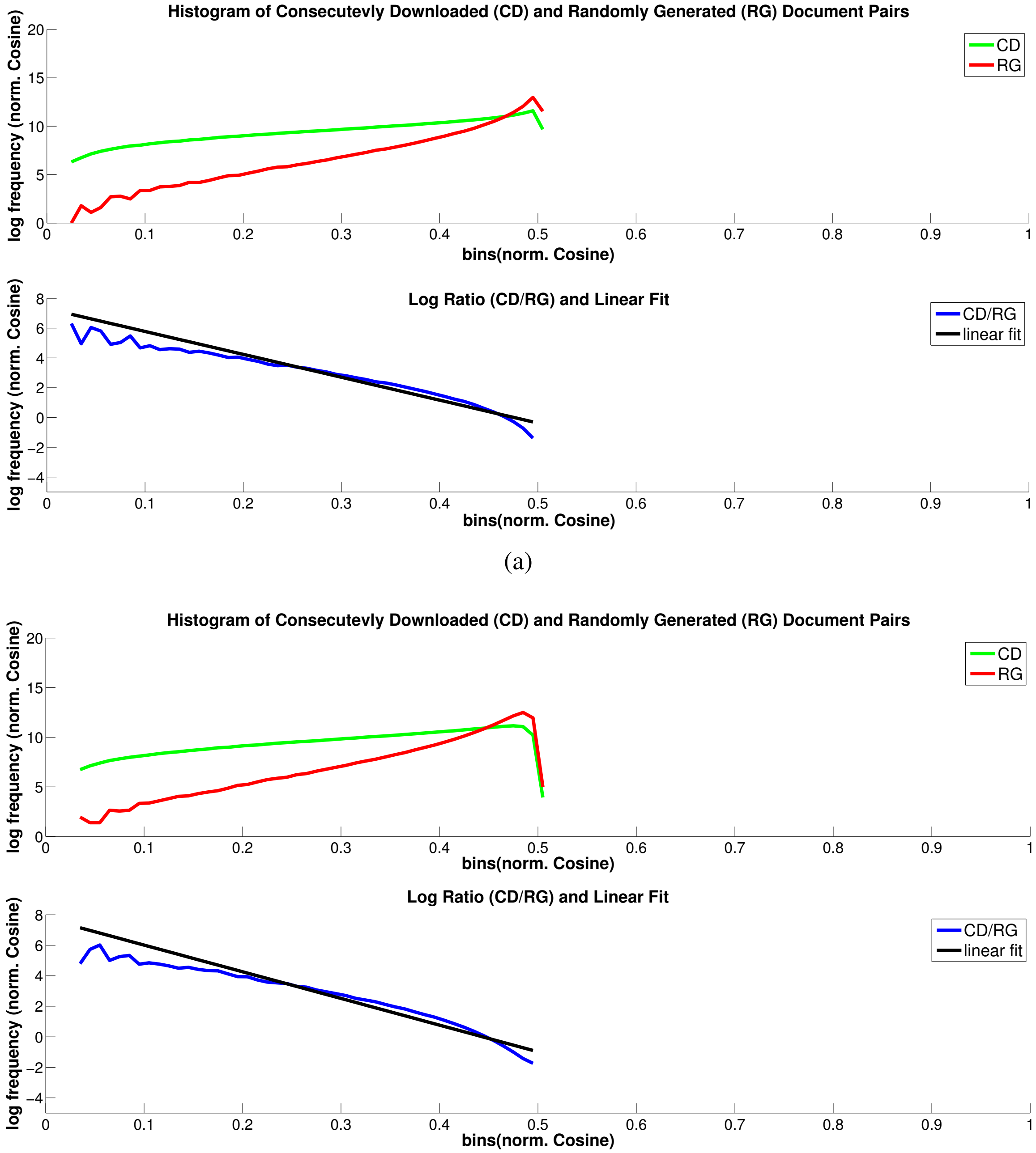
In all log division plots we observe an exponential form of the function which for a given query document, as explained earlier, gives us the distribution of observing similar (i.e. query relevant) documents at certain similarity values. This distribution for different similarity metrics could be parametrized with a single parameter. For example, α for normalized Cos distance and β for JS divergence:

$$LogDiv_{norm.(Cosine)} \approx \exp^{\alpha norm.(Cosine)} \tag{6.6}$$

$$LogDiv_{JS} \approx \exp^{\beta JS} \tag{6.7}$$

6.7.3 Correlation Analysis of rHSA

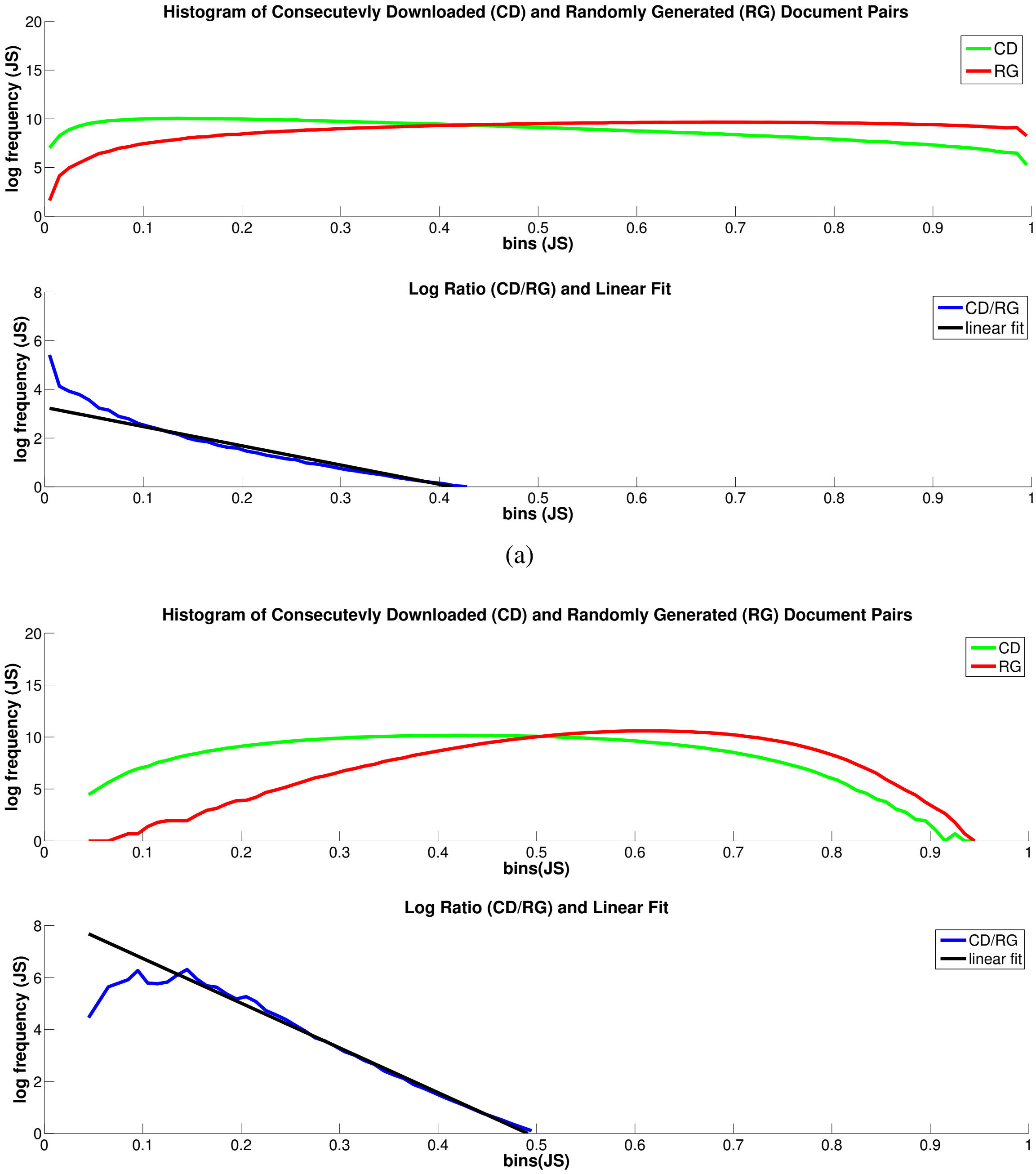
In this section we analyze the performance of rHSA using evaluation experiments similar to the ones performed on HSA. We analyze the ability of rHSA to predicting the performance of document similarity models on the task of retrieving CD document pairs using the 1M and 100q test collections. We perform linear and rank correlation analysis between ranked lists obtained using rHSA and ranked list obtained using MAP which was computed on the 100q test set. On the 100q test set rHSA was computed using similarity scores between the query and query pseudo-relevant documents which defines the set of CD document pairs while the set of RG document pairs consisted of the query and non-relevant



(a)

(b)

Figure 6.7: Computing rHSA example: Evaluation of term frequency–inverse document frequency (tf–idf) models on the task of retrieving CD documents given a query document. rHSA was computed over 1M CD and RG document pairs that were obtained using the vector space model where each document was represented using its top 50 (a) and all (b) tf–idf terms.



(a)

(b)

Figure 6.8: Computing rHSA example: Evaluation of LDA models on the task of retrieving CD documents given a query document. rHSA was computed over 1M CD and RG document pairs that were obtained using LDA configured with 50 (a) and 2000 (b) topics.

documents. For this test set this is equivalent to how we compute HSA with the exception that in this case we propagate the modeled errors in the observed similarity values. Analysis are performed over two families of retrieval models with different configuration settings. The two families of models represent the CD documents in two different spaces, the topic space using LDA and the vector space using tf–idf values.

For our LDA model we use the Vowpal Wabbit [69] implementation of oLDA. Topical representation of articles was done using an effective vocabulary of 16,448 tokens which were extracted from the total vocabulary of tokens present across the 90,395 ApJ articles. The effective vocabulary was generated by filtering out tokens whose frequency count across the collection was less than 10 and by removing the 50 most frequent tokens. We also removed all the tokens whose character length was less than four and tokens that had non alphabetic characters such as numeric and special characters. We used 9 different topic configurations with the number of topics set to T=50, 100, 500, 1k, 2k, 3k, 4k, 5k and 10k.

In the vector space, articles were represented using models [30] where documents are represented using tf–idf values. Non-query documents in our collection were represented using the tf–idf values computed across all the tokens in the effective vocabulary. Query documents on the other hand were represented using four different tf–idf configurations where each configuration varies by the number of top tf–idf values that were used to represent the document. When representing a query article we first compute a ranked list of tf–idf values across the tokens present in the document. From the generated ranked list we use the top n tf–idf tokens (n =50, 100 and 500) to represent the query. We also create a fourth tf–idf representation using all of the tf–idf tokens found in the query. Similarity across the documents is computed using normalized Cos distance:

$$normalized\, Cosine = \frac{Cosine}{max(Cosine) - min(Cosine)} \tag{6.8}$$

Document Similarity Model Type	Collection	R	ρ
tf–idf	100q	0.97	1.00
	1M	0.98	1.00
LDA	100q	0.99	0.95
	1M	0.97	0.88

Table 6.8: Evaluating document similarity models in a scholarly IR system using rHSA: Pearson (R) and Spearman's (ρ) coefficients computed over MAP.

Table 6.8 shows results of our linear (R) and rank (ρ) correlation analysis between ranked lists of the models' performance sorted by MAP and ranked lists obtained by rHSA across the 1M and 100q test sets. Coefficients were computed across the two families of models and different model configurations.

rHSA yields a very high linear and rank correlation with MAP across both families of retrieval models and compute collections. Over the tf–idf family of models and across the two rHSA compute collections we achieve almost identical linear and rank correlation. Over the family of LDA models, linear correlation across the two test sets is almost the same. Over the 1M compute set we achieve the lowest rank correlation. Overall, the high correlation coefficients allow us to use rHSA and predict the performance of a document similarity model prior to using that model on particular document similarity task. We achieve this by computing rHSA on a set of CD and RG document pairs from the same collection.

CHAPTER 7

TOPIC MODELS WITH MULTI-LEVEL DIRICHLET PRIORS FOR MODELING TOPICAL VARIATIONS ACROSS TEXTUAL REGIONS

7.1 Introduction

Many latent variable models of text use the bag of words assumption [81] when modeling documents. This assumption ignores the document structure, ordering of words and discards syntactic information such as sentence boundary and other grammatical information. When modeling the generative process of documents latent variable models of text, such as topic models, rely on word co-occurrence statistics. Since bag of words models ignore the document specific grammatical information, word co-occurrence statistics are computed on a document level. When co-occurrence statistics are computed on a document level they ignore the effects of the words co-occurring close to each other versus words co-occurring further apart. While this representation simplifies the modeling process it prevents topic models from modeling topical variations that occur across different textual regions of the document. Take for example long documents commonly found in books or long scholarly works such as journal articles. In these documents words co-occurring in the introduction or the conclusion sections have different co-occurrence effects versus words that are present in the same section. With the bags of words assumption, in order to model topical variations in collections of large documents one must first decide on what will be considered as a document. This is certainly the case when modeling documents with hierarchical structures such as messages from discussion threads or posts in posting lists.

Often a decision could be made to treat individual textual regions, such as sections of the journal article or a discussion thread message, as individual documents. This simplifies the modeling approach and certainly captures the word co-occurrence effect within the textual region but it completely ignores the natural coupling that exist across regions of text that are part of the same document. Across document sections, for example, there exists a hierarchical relationship between the sections of the document and the document as a whole. Therefore this type of an approach, while more straightforward, it would not explicitly model the topical variations across the whole document. Topics present in textual regions of a large document originate from the same pool of topics that are document specific. Therefore while different textual regions of the document may not be on the same topic or set of topics they will originate from the same pool of, document specific, topics. To illustrate this notion take for example a scientific journal article that contains seven sections which includes the introduction section and the conclusion. While one would not expect a strong topical relatedness between the introduction and one of the article's experimental sections for example, all sections would need to originate from the same family of distributions specified by the document level topic distribution. This natural hierarchical relationship that exists across textual regions and the document as a whole requires for the topical variations to be modeled explicitly using topic distributions that are from the same family of distributions.

In this chapter we introduce an online variational Bayes (VB) inference approach which uses multi-level Dirichlet prior structure to directly model document specific topical variations across textual regions. The proposed approach assigns Dirichlet priors to individual textual regions that are coupled by a document level hierarchical Dirichlet prior. Our proposed online inference approach streamlines the modeling of topical variations in large documents with predetermined textual region structure. We create two variants of our approach which enhances the topical model representation of documents in mono- and multilingual collections. We call our approach for modeling topical variations in monolingual collections multi-level hyperpriors latent Dirichlet allocation (mlhLDA). Our approach for multilingual collections is an extension of our previous oPLTM model that we presented in Chapter 4. We refer to this new variant of oPLTM as multi-level hyperpriors polylingual topic model (mlhPLTM). In Section 7.2 we give an overview of the previous work that was done on modeling hierarchical relationships in document collections. We then introduce our modeling approach using online VB and present mlhLDA and mlhPLTM. The modeling advantages of both proposed approaches are demonstrated on two collections. For mlhLDA we used a collection of journal articles in the domain of physics and astrophysics. The advantages of mlhPLTM are demonstrated on our test collection of English-Spanish Europarl speeches (§ 4.4).

7.2 Previous Work on Modeling Topic Variations across Textual Regions

Our approach most closely follows the approach by Wallach et al. [124] which showed that using asymmetric Dirichlet priors over document-topic distributions offer modeling advantage over symmetric priors as measured by perplexity. This approach treats the base measures vector u as a hidden variable and assigns a symmetric Dirichlet prior to it which in turn creates a hierarchical Dirichlet prior structure over all document-topic distributions in the collection. Similar to the original LDA, the approach in [124] assigns a single topic distribution for each document in the collection. While it provides better modeling for the whole collection, this approach does not model variations across textual regions of the same document. Furthermore inference is performed using Gibbs sampling which makes the asymmetric approach to become inefficient when dealing with large document collections.

A new modeling alternative that relies on VB was introduced by Kim et al. [62] which models monolingual document collections with nested hierarchies. This variant of the LDA model, called tiLDA, achieves better modeling performance over large document collections and on document collections with deep hierarchies, such as posting lists with multiple hierarchical levels, by utilizing parallel variational inference. The drawback of this approach is that it is depended on a parallel computational approach for updating the variational parameters. But more importantly, similar to Gibbs sampling, it requires multiple iterations over the whole collection.

In the past, various other variations of the LDA model have been proposed that take into account the word ordering in the document. For example, Griffiths et al. [44] models the sequence of words, i.e the syntactic dependency found in the document by introducing Hidden Markov Model (HMM) within the LDA framework. Topical n-gram model (TNG) introduced in [127] models unigram and n-gram phrases as mixture of topics based on the nearby word context. Jameel and Lam [54] proposed an extension of the LDA model that does topic representation of documents on two levels: using word sequence information to generate topic distribution over n-grams and segment and paragraph information to perform topic segmentation. These and many other extensions have been shown to outperform the basic LDA model on various tasks.

While the above approaches offer a better and more realistic modeling of the word sequence they don't model the variations of topics across different sections of the document especially in multilingual collections.

7.3 Efficient Multi-level Hyperpriors

In the original implementation of the LDA model, as it was highlighted in § 2.2.4, symmetric Dirichlet priors are used to model the document-topic distributions θ_d and topic-word distributions φ_t. Symmetric Dirichlet priors are also used in the original polylingual topic model (PLTM). This approach assumes that all documents in the collection are drawn from the same document-topic family of distributions which doesn't generalize well especially in instances where the collection consists of documents that cover diverse range of topics.

One way to alleviate this problem is to assign asymmetric Dirichlet priors to individual documents by treating the base measures vector u as a hidden variable and assign a symmetric Dirichlet prior to it which creates a hierarchical Dirichlet prior structure over all document-topic distributions in the collection. This is the approach that was introduced in [124] which was shown to offer modeling advantage over symmetric priors. In our case we model topic variations across textual regions within the document and therefore assign textual region-topic distribution θ_r which uses the document-topic distribution as a symmetric Dirichlet priors. This tight coupling between the R textual regions and the document d as a whole creates a hierarchical Dirichlet structure ($\theta = \theta_d, \theta_{r_1}, \theta_{r_2}, ..., \theta_{r_R}$):

$$p(\theta|\alpha_d u, \alpha_r) \propto p(\theta_d|\alpha_d u) \prod_r p(\theta_r|\alpha_r \theta_d) \tag{7.1}$$

For each textual region the model assumes that words are drawn from a specific distribution over topics whose prior is the document specific Dirichlet distribution. The hierarchy defines a Dirichlet-multinomial problem where the goal is to estimate the Dirichlet distribution given the drawn multinomial distributions which in our case are the textual region-topic distributions. The most widely used approach for estimating θ_d is Minka's [86] fixed-point iteration approach which is also used in [62]. In [123] author showcased that estimating Dirichlet-multinomial hyperparameters could be more efficient by approximating the digamma differences in Minka's approach which we also follow in our online implementation. In our case the update step is defined as:

$$\gamma_t^{d*} = \gamma_t^d \frac{\sum_{n=1}^{M_t} C_t(n) \left(\frac{1}{\gamma_t^d} + \log \frac{n+\gamma_t^d - \frac{1}{2}}{\gamma_t^d + \frac{1}{2}} \right)}{\sum_{n=1}^{M_s} C(n) \left(\frac{1}{\alpha_d} + \log \frac{n+\alpha_d - \frac{1}{2}}{\alpha_d + \frac{1}{2}} \right)} \tag{7.2}$$

Where γ^d is the document-topic distribution; $C_t(n)$ is the number of textual regions r in which topic t was assigned n times with $M_t = max(C_t(n))$; $C(n)$ is the number of textual regions in the document whose length is n with $M_s = max(C(n))$;

7.3.1 Inference using Online VB

Unlike Gibbs sampling [43] and the more efficient computation approach using VB [16], both of which require iterating over the whole collection multiple times, in our implementation of mlhLDA and mlhPLTM we use online VB inference [49]. This approach, which was also used for our online polylingual topic model (oPLTM) implementation (§ 4.2), requires a single pass over the whole collection. When inferring the hierarchical Dirichlet relationship we use the lower bound which was derived by Kim et al. [62] . This lower bound, which is looser than the original VB Evidence Lower Bound (ELBO), allows for the batch VB approach to be used with asymmetric priors. In our online VB approach we follow the proof of their theorem when computing the textual region-topic variational parameters which we also expand in our mlhPLTM model. More specifically, given the document-topic variational parameter γ_t^d in the E-step of our online VB approach the update for the textual region-topic variational parameter γ_t^r becomes:

$$\gamma_t^r = \alpha_r(\frac{\gamma_t^d}{\sum_t \gamma_t^d}) + \sum_w \phi_{wt}^r \, n_w^r \tag{7.3}$$

7.4 Multi-level Hyperpriors LDA (mlhLDA)

Similar to the LDA model, given a collection of D documents, mlhLDA first generates a set of $t \in \{1, 2, ..., T\}$ topic-word distributions φ_t which are drawn from a Dirichlet prior with hyperparameter β, $\varphi_t \sim Dir(\beta)$. For each document d with r_d textual regions in the collection, mlhLDA assumes the following generative process:

- Choose $\theta_d \sim Dir(\alpha_d)$

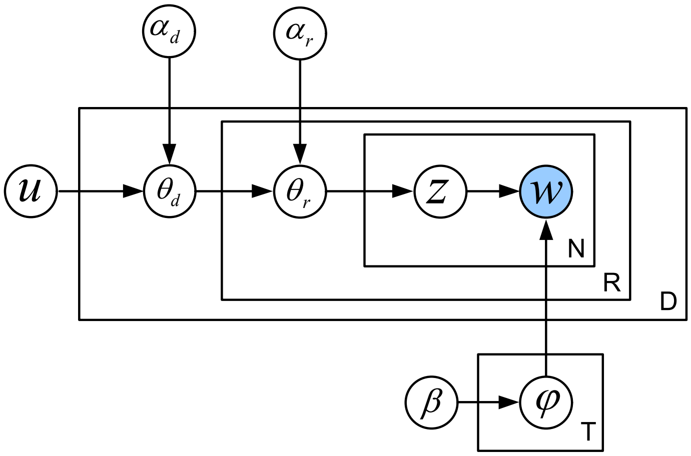


Figure 7.1: Graphical representation of the multi-level hyperpriors LDA (mlhLDA).

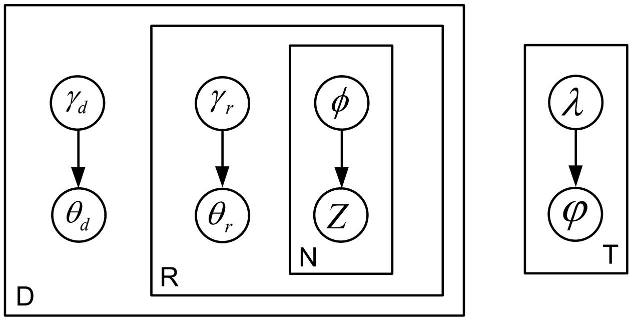


Figure 7.2: Free variational parameters for the online VB approximation in the mlhLDA model.

- For each textual region r in document d:
  - Choose $\theta_r \sim Dir(\alpha_r \theta_d)$
  - For each word w in textual region r_d:
    - Choose a topic assignment $z \sim Multinomial(\theta_r)$
    - Choose a word $w \sim Multinomial(\varphi_z)$

Figure 7.1 shows a graphical representation of the mlhLDA model while in Figure 7.2 we show a graphical representation of the variational parameters for the online VB approximation of the model posteriors. In Algorithm 2 we detail the implementation of mlhLDA.

Algorithm 2 Online VB for multi-level hyperpriors LDA (mlhLDA)

```
initialize $\lambda$ randomly
*obtain the $b$-th mini-batch of documents $d$*
**for** $b = 1$ to $\infty$ **do**
    $\rho_b \leftarrow \left(\frac{1}{\tau_0 + b}\right)^{\kappa}$
    initialize $\gamma_b$ randomly                                  ▷ *Begin E-step:*
    *for each document $d$ in mini-batch $b$*
    ***for** $d$ in $b$ **do***
        ***repeat***
            *for each region $r$ in document $d$*
            ***for** $r$ in $d$ **do***
                ***repeat***
                    $\phi^r_{wt} \propto \exp\left\{E_q\left[\log \theta^r_t\right] + E_q\left[\log \varphi^r_{tw}\right]\right\}$
                    $\gamma^r_t = \alpha_r(\frac{\gamma^d_t}{\sum_t \gamma^d_t}) + \sum_w \phi^r_{wt}\, n^r_w$
                ***until** convergence*
            ***end for***
            *Update($\gamma^d_t$)*                                  ▷ *[Eq. 7.2 in § 7.4]*
        ***until** convergence*
    ***end for***
    $\tilde{\lambda}_{tw} = \beta + \frac{D}{|b|} \sum_{d=1}^{|b|} n^d_w \phi^d_{wt}$            ▷ *Begin M-step:*
    $\lambda^b_{tw} \leftarrow (1 - \rho_b)\, \lambda^{b-1}_{tw} + \rho_b \tilde{\lambda}_{tw}$
***end for***
```

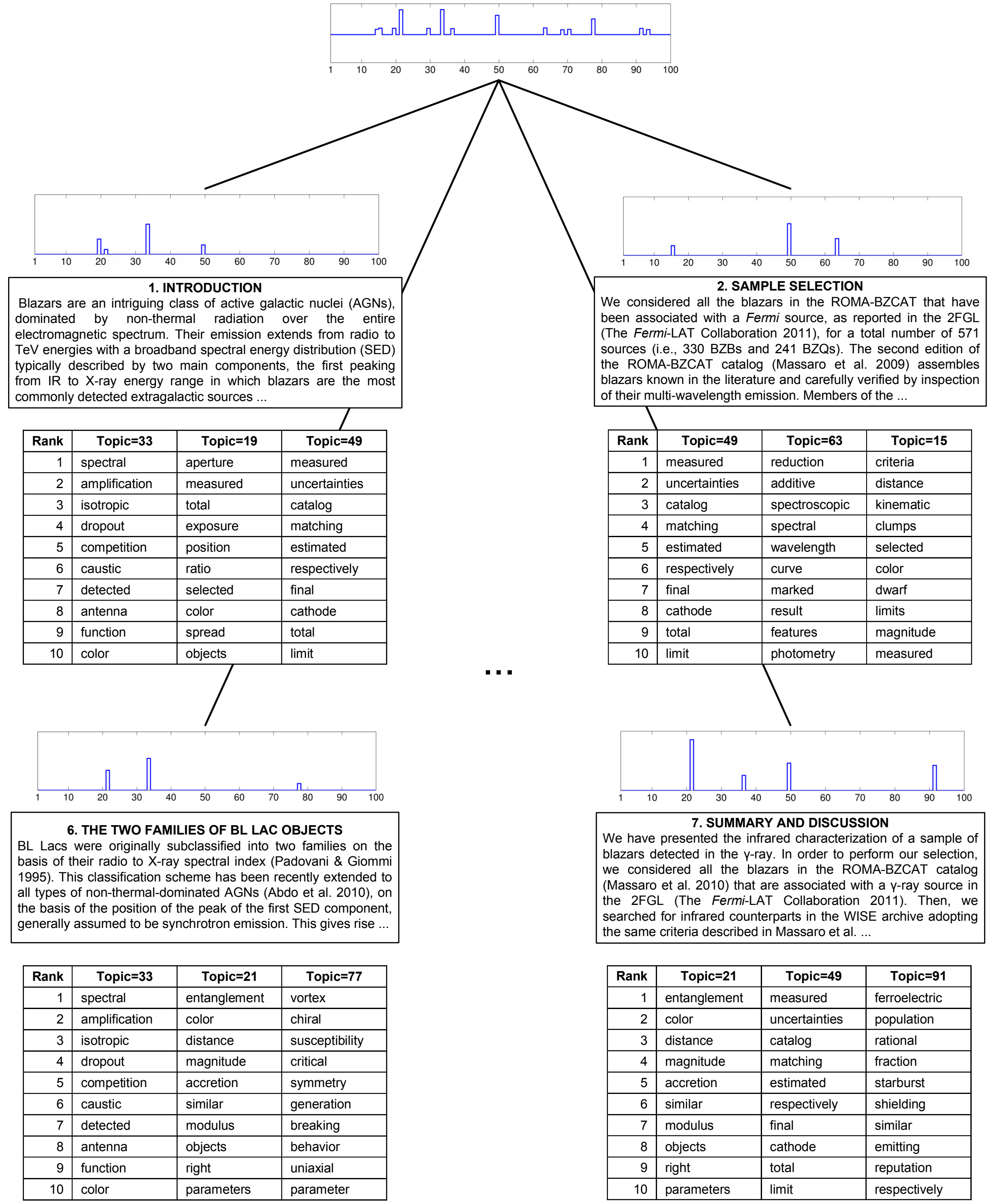
7.4.1 Modeling Sections in Scientific Articles

We compared the modeling performance of mlhLDA, oLDA and batch VB using a collection of journal articles from the Astrophysics Data System (ADS) (§ 6.7.1). More specifically we used all scientific articles that were published in the Physical Review journals A, B, C, D, E, L and S and the Astrophysical Journal (ApJ) in the past ten years and created a training collection of 130k articles (888,346 sections) and a held-out set of 8,078 articles (54,502 sections). On average, across both domains, journal articles contain $\sim$7 sections not including the abstract. Figure 7.3 shows a mlhLDA representation (configured with 100 topics) of the ApJ article titled "Infrared colors of the Gamma-Ray Detected Blazers". Shown on the top is the inferred topic representation of the whole document (θ_d) which in the hierarchic structure of the mlhLDA model serves as a hyperprior for the textual region-topic distributions (θ_r) or on this case section-topic distributions. Shown on the bottom are examples of four article sections (out of seven) and their inferred topic distributions along with the top ten words for each of the top three topics found in the section which are ordered from left to right.

Due to lack of an extrinsic evaluation task, comparison between the two models was performed by computing perplexity on the held-out set. When modeling journal articles using oLDA, articles sections were treated as individual documents. We first compared the perplexity between oLDA and mlhLDA using 13 different topic configurations T=50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1k, 1.5k and 2k. Both models were also configured with the same concentration parameter values (α_d=α_r=β=$\frac{1}{T}$ for mlhLDA). Figure 7.4 shows the results of the perplexity comparison.

7.4.1.1 Comparisons with Batch VB

In addition to comparing oLDA and mlhLDA we also compared the inference time and perplexity of these two models with the original batch VB implementation [13] of Blei et al. [16]. Unlike the implementations of oLDA [50] and mlhLDA which are written in



1. INTRODUCTION

Blazars are an intriguing class of active galactic nuclei (AGNs), dominated by non-thermal radiation over the entire electromagnetic spectrum. Their emission extends from radio to TeV energies with a broadband spectral energy distribution (SED) typically described by two main components, the first peaking from IR to X-ray energy range in which blazars are the most commonly detected extragalactic sources ...

Rank	Topic=33	Topic=19	Topic=49
1	spectral	aperture	measured
2	amplification	measured	uncertainties
3	isotropic	total	catalog
4	dropout	exposure	matching
5	competition	position	estimated
6	caustic	ratio	respectively
7	detected	selected	final
8	antenna	color	cathode
9	function	spread	total
10	color	objects	limit

2. SAMPLE SELECTION

We considered all the blazars in the ROMA-BZCAT that have been associated with a *Fermi* source, as reported in the 2FGL (The *Fermi*-LAT Collaboration 2011), for a total number of 571 sources (i.e., 330 BZBs and 241 BZQs). The second edition of the ROMA-BZCAT catalog (Massaro et al. 2009) assembles blazars known in the literature and carefully verified by inspection of their multi-wavelength emission. Members of the ...

Rank	Topic=49	Topic=63	Topic=15
1	measured	reduction	criteria
2	uncertainties	additive	distance
3	catalog	spectroscopic	kinematic
4	matching	spectral	clumps
5	estimated	wavelength	selected
6	respectively	curve	color
7	final	marked	dwarf
8	cathode	result	limits
9	total	features	magnitude
10	limit	photometry	measured

6. THE TWO FAMILIES OF BL LAC OBJECTS

BL Lacs were originally subclassified into two families on the basis of their radio to X-ray spectral index (Padovani & Giommi 1995). This classification scheme has been recently extended to all types of non-thermal-dominated AGNs (Abdo et al. 2010), on the basis of the position of the peak of the first SED component, generally assumed to be synchrotron emission. This gives rise ...

Rank	Topic=33	Topic=21	Topic=77
1	spectral	entanglement	vortex
2	amplification	color	chiral
3	isotropic	distance	susceptibility
4	dropout	magnitude	critical
5	competition	accretion	symmetry
6	caustic	similar	generation
7	detected	modulus	breaking
8	antenna	objects	behavior
9	function	right	uniaxial
10	color	parameters	parameter

7. SUMMARY AND DISCUSSION

We have presented the infrared characterization of a sample of blazars detected in the γ-ray. In order to perform our selection, we considered all the blazars in the ROMA-BZCAT catalog (Massaro et al. 2010) that are associated with a γ-ray source in the 2FGL (The *Fermi*-LAT Collaboration 2011). Then, we searched for infrared counterparts in the WISE archive adopting the same criteria described in Massaro et al. ...

Rank	Topic=21	Topic=49	Topic=91
1	entanglement	measured	ferroelectric
2	color	uncertainties	population
3	distance	catalog	rational
4	magnitude	matching	fraction
5	accretion	estimated	starburst
6	similar	respectively	shielding
7	modulus	final	similar
8	objects	cathode	emitting
9	right	total	reputation
10	parameters	limit	respectively

Figure 7.3: mlhLDA representation of the ApJ article titled “Infrared Colors of the Gamma-Ray Detected Blazers” in a 100 dimensional topic space.

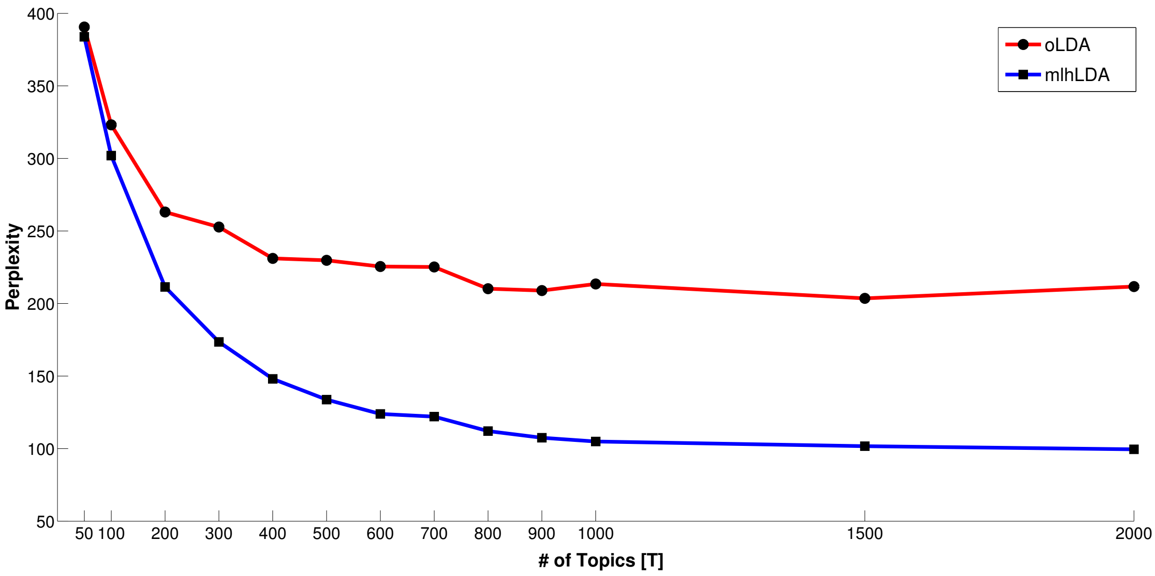


Figure 7.4: oLDA vs. mlhLDA: Perplexity comparison on the task of modeling scientific articles using 13 different topic configurations.

Python the original VB algorithm is written in C and requires multiple iterations over the whole collection. As an illustration, running the batch VB implementation (called lda-c), configured with 100 topics, on the original training set required more than a day compared to oLDA which took around two hours. In this experimental setup we used a random subset of 10k training articles and a held-out set of 1k articles. Figure 7.5 shows the speed vs. perplexity comparison across the three models using a set of 8 topics T=5, 10, 20, 30, 50, 70, 90 and 100.

7.5 Multi-level Hyperpriors PLTM (mlhPLTM)

In Chapter 4 we introduced oPLTM which provides similar performance compared to original implementation of PLTM while being more computationally efficient. In this section we extend oPLTM to better model topical variations across textual regions by introducing textual region specific topic distributions θ_r across different languages in the multilingual document tuple which are coupled by the tuple specific document-topic distribution θ_d. Figure 7.6 shows the graphical representation of mlhPLTM.

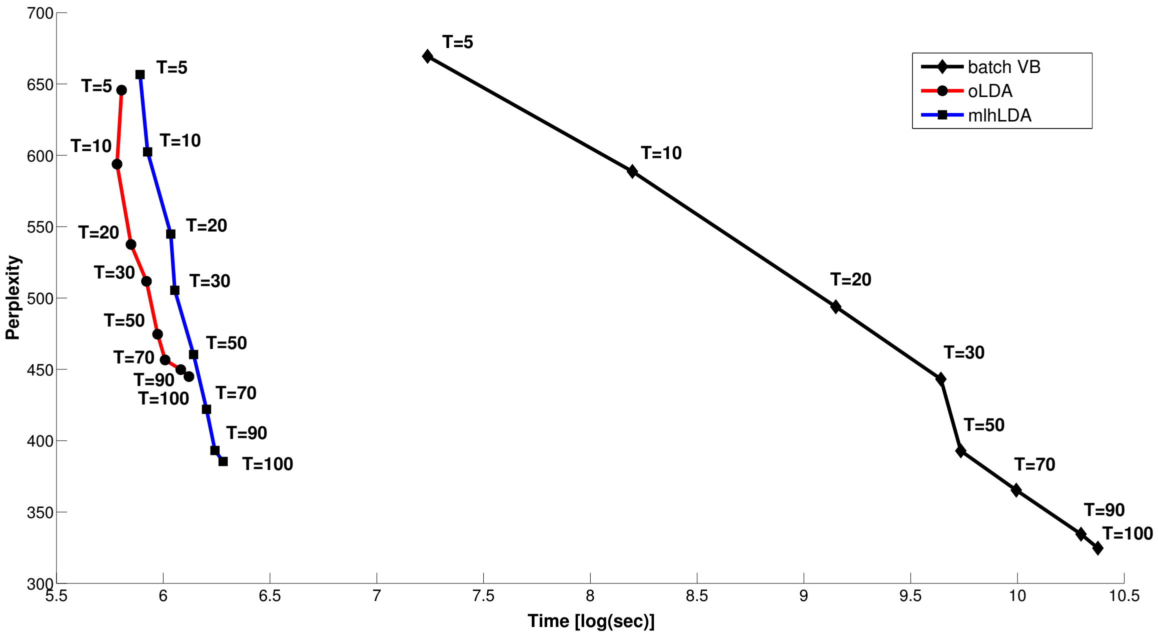


Figure 7.5: Speed vs. perplexity: Comparisons between oLDA, mlhLDA and batch VB across 8 topic configurations. Inference time is shown in natural log scale.

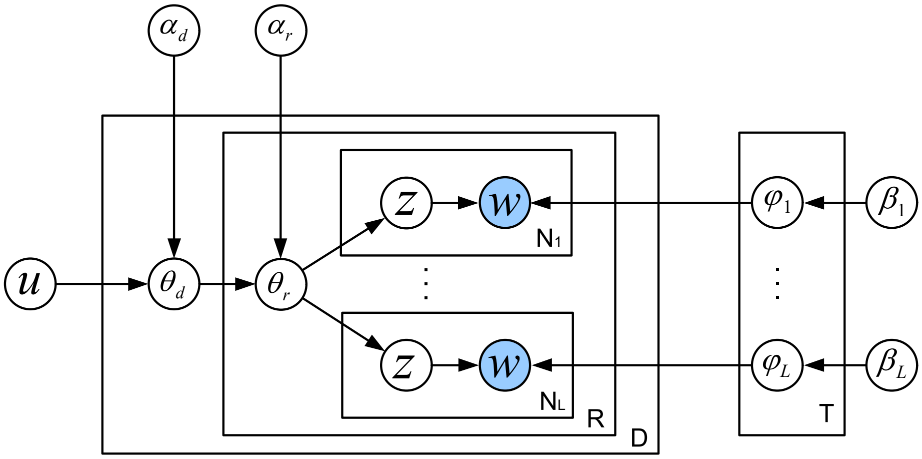


Figure 7.6: Graphical representation of the multi-level hyperpriors PLTM (mlhPLTM).

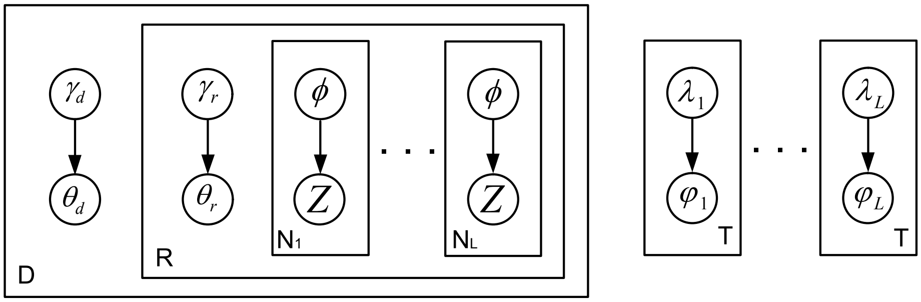


Figure 7.7: Free variational parameters for the online VB approximation in the mlhPLTM model.

Given a collection of document tuples d where each tuple contains one or many documents that are topically similar in different languages (L), mlhPLTM assumes the following generative process. For each language l in the collection the model first generates a set of $t \in \{1, 2, ..., T\}$ topic-word distributions, φ_t^l which are drawn from a Dirichlet prior with language specific hyperparameter β^l: $\varphi_t^l \sim Dir(\beta^l)$. For each document d^l with r_d^l textual regions in tuple d, mlhPLTM then assumes the following generative process:

- Choose $\theta_d \sim Dir(\alpha_d)$
- For each textual region r in the document tuple d:
- Choose $\theta_r \sim Dir(\alpha_r \theta_d)$
  - For each language l in textual region r:
    - For each word w in textual region r_d^l:
      - Choose a topic assignment $z \sim Multinomial(\theta_r)$
      - Choose a word $w \sim Multinomial(\varphi_z^l)$

The algorithmic representation of mlhPLTM is detailed in Algorithm 3 with the free variational parameters shown in Figure 7.7.

Algorithm 3 Online VB for multi-level hyperpriors PLTM (mlhPLTM)

```
initialize $\lambda_l$ randomly
*obtain the $b$th mini-batch of tuples $d$*
**for** $b = 1$ to $\infty$ **do**
    $\rho_b \leftarrow (\frac{1}{\tau_0+b})^\kappa$
    initialize $\gamma_b$ randomly                                    ▷ *Begin E-step:*
    *for each document tuple in mini-batch $b$*
    ***for** $d$ in $b$ **do***
        ***repeat***
            *for each textual region $r$ in document tuple $d$*
            ***for** $r$ in $d$ **do***
                ***repeat***
                    *for each language $l$ in textual region $r$*
                    ***for** $l \in 1, \ldots, L$ **do***
                        $\phi_{wt}^{rl} \propto \exp\left\{E_q\left[\log \theta_t^r\right] + E_q\left[\log \varphi_{tw}^{rl}\right]\right\}$
                    ***end for***
                    $\gamma_t^r = \alpha_r(\frac{\gamma_t^d}{\sum_t \gamma_t^d}) + \sum_l \sum_w \phi_{wt}^{rl} \, n_w^{rl}$
                ***until** convergence*
            ***end for***
            *Update($\gamma_t^d$)*                                    ▷ *[Eq. 7.2 in § 7.4]*
        ***until** convergence*
    ***end for***
    ***for** $l \in 1, \ldots, L$ **do***                                ▷ *Begin M-step:*
        $\tilde{\lambda}_{tw}^l = \beta + \frac{D}{|b|} \sum_{d=1}^{|b|} n_w^{dl} \phi_{wt}^{dl}$
        $\lambda_{tw}^{lb} \leftarrow (1 - \rho_b)\, \lambda_{tw}^{l(b-1)} + \rho_b \tilde{\lambda}_{tw}^l$
    ***end for***
***end for***
```

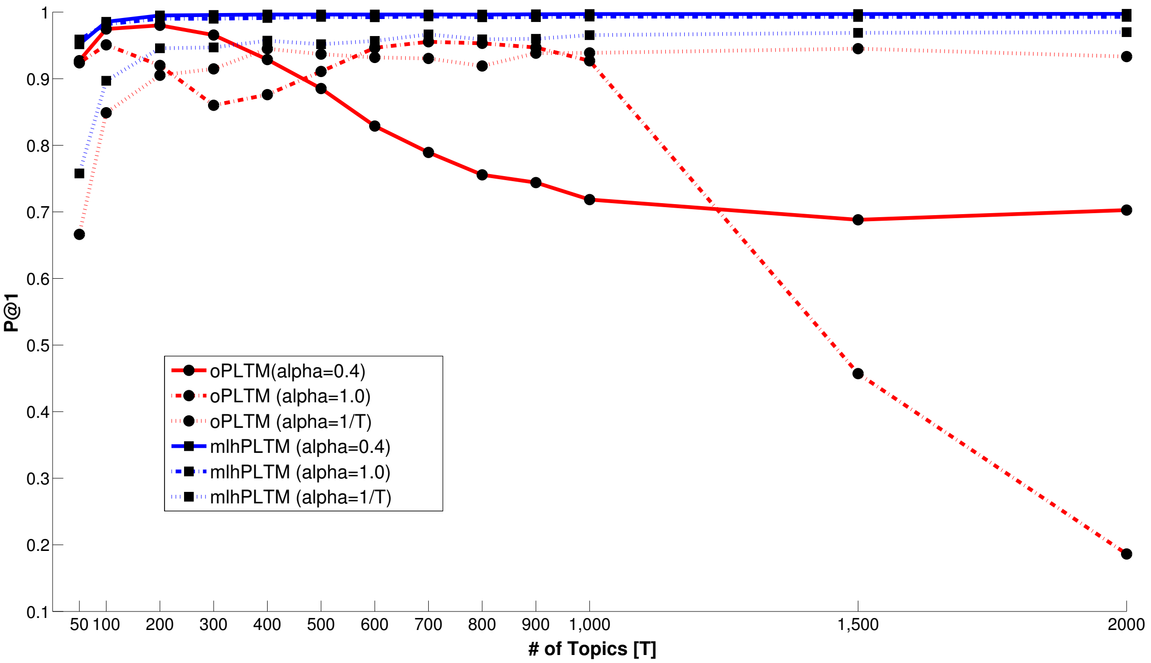


Figure 7.8: oPLTM vs. mlhPLTM: Performance comparison on the CLIR task using chronological ordering of sessions across different hyperparameter settings (α_d=α_s=β_l=$\frac{1}{T}$, 0.4 and 1.0).

7.5.1 mlhPLTM Performance Analysis

Using our English-Spanish Europarl evaluation set (§ 4.4) we compared the model performance of mlhPLTM and oPLTM. This set consists of ∼64k training pairs of English-Spanish speeches which originate from 374 European Parliament sessions and a test set of ∼14k speech translation pairs from 112 sessions. On average each Europarl session contains ∼170 speeches. With oPLTM we modeled individual speech pairs while with mlhPLTM we utilized the session hierarchy and modeled pairs of speeches as textual regions. Model performance was compared intrinsically (using perplexity) and extrinsically on our task of retrieving document translation pairs (§ 3.7.4) where performance was evaluated using precision of the top rank (P@1). Figure 7.8 shows the performance comparison between oPLTM and mlhPLTM across three different settings of the concentration parameters α_d, α_r and β_l (α_d=α_r=β_l=$\frac{1}{T}$, 0.4 and 1.0) and 13 different topic configurations. Across both models, in the training and test steps each batch contained a single Europarl session.

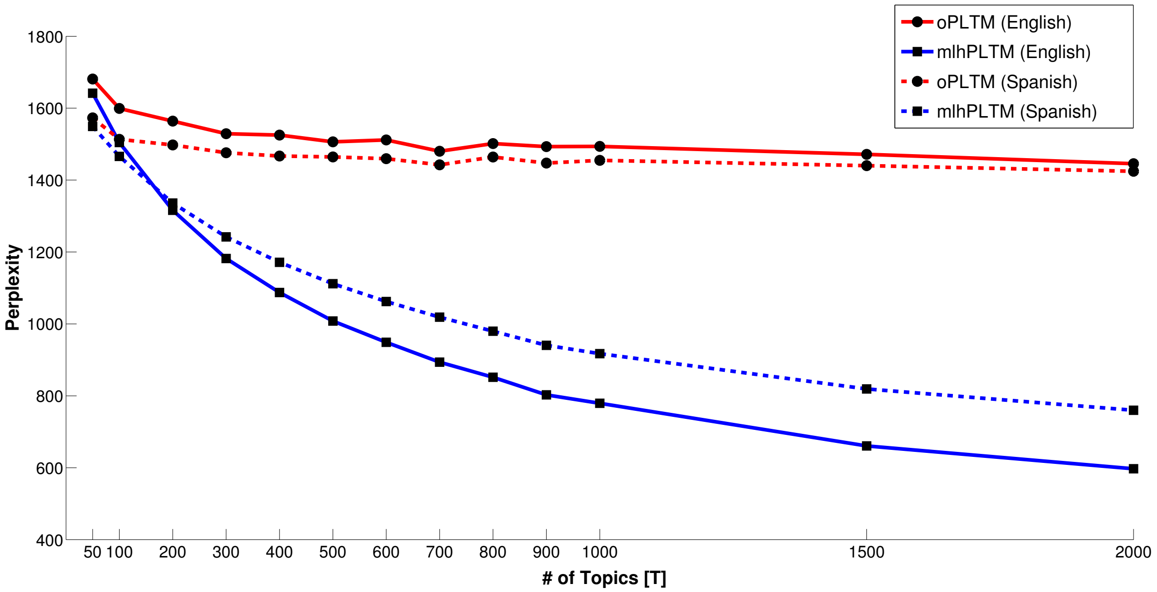


Figure 7.9: oPLTM vs. mlhPLTM: perplexity comparison across 13 different topic configurations on the task of modeling English-Spanish Europarl speeches (α_d=α_r=β_l=$\frac{1}{T}$). Documents were presented out of chronological order.

Unlike the performance of oPLTM which tends to fluctuate across different values of the concentration parameter and different topic configurations, the performance of mlhPLTM is very steady. As we increase the number of topics, across the three different concentration parameter settings, we also observe that mlhPLTM has tendency to improve its performance. Lastly, we observe that the overall best performance for both models is achieved with $\alpha_d = \alpha_r = \beta_l = 0.4$ and the worst performance is obtained with $\alpha_d = \alpha_r = \beta_l = \frac{1}{T}$.

In the initial experimental setup that we constructed to evaluate mlhPLTM and compare its performance with oPLTM we unintentionally reordered the training set of Europarl speeches based on two digit years which was different from our original setup in (§ 4.4) where speeches were presented chronologically. This emphasized the importance of the presentation order in online VB especially in the training step. Figure 7.9 shows the perplexity comparison between the two models when the order of Europarl speeches in the training and test step is numerical rather than chronological.

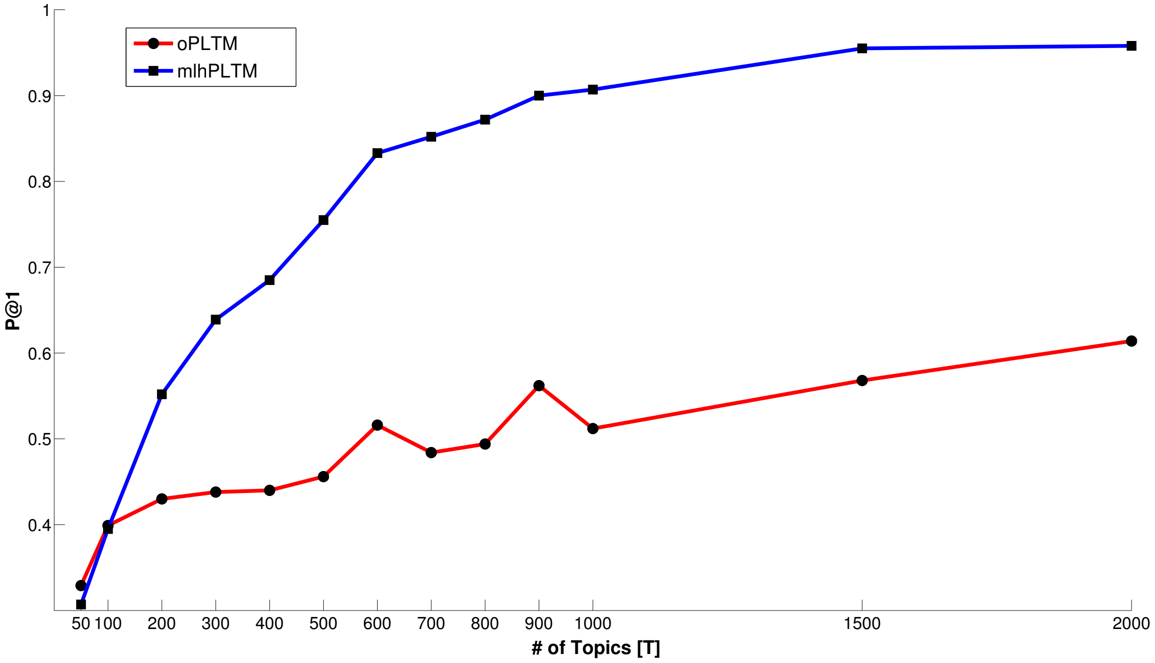


Figure 7.10: oPLTM vs. mlhPLTM: Performance comparison on the CLIR task (α_d=α_r=β_l=$\frac{1}{T}$). Documents were presented out of chronological order and thus performance is lower, especially for oPLTM.

As in the case of the experimental setup in (§ 7.4.1), in our initial experimental setup we set the concentration parameters to α_d=α_r=β_l=$\frac{1}{T}$. We also trained both models using an arbitrary batch size of 11 sessions while the test batch contains a single session. Figure 7.10 shows the performance comparison between oPLTM and mlhPLTM on the CLIR task when documents are presented out of chronological order.

In this experimental setup, mlhPLTM achieves the best performance with T=2k. While it takes higher number of topics to achieve high performance, mlhPLTM ultimately achieves similar performance results as ordered mlhPLTM.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The governing question in this dissertation was how to make latent variable models of text more practical especially when dealing with large document collections and collections with large documents. To that end we introduced techniques for performing efficient modeling, inference, search and evaluation for topic models which are the most commonly used latent variable models of text. Listed below are the contributions that we made along with the type of efficiency improvements that we achieved.

We started of by introducing efficient nearest-neighbor (NN) search approach for the probability simplex. More specifically, by transforming probability distributions we showed that we are able to compute Hellinger (He) distance using approximate NN search techniques in the probability simplex. Approximating Jensen-Shannon (JS) divergence with He distance we showed that we were able to also use approximate NN search for computing JS divergence. Across three different tasks we evaluated the efficiency of the approximate NN search approach by measuring the relative speed improvement over exhaustive NN search. Based on the significant speed improvements (e.g. $\sim$105 times on the patent retrieval task) we concluded that approximate NN search techniques provide efficient means to perform similarity computation between probability distributions while maintaining similar level of accuracy as exhaustive NN search.

To streamline the use of the polylingual topic model (PLTM) in large multilingual collections we presented online PLTM oPLTM) which uses online variational Bayes (VB) to provide efficient inference. The efficiency of oPLTM was showcased by analyzing the inference time and the memory requirements needed to process collections in a single stream fashion. Across collections of different sizes we showed that oPLTM requires significantly less training time and constant size memory compared to the original PLTM which uses Gibbs sampling. When evaluated on the task of retrieving document translation pairs, oPLTM achieved similar results compared to regular PLTM.

The efficiency of our approximate NN search approach and oPLTM was also demonstrated on the task of extracting parallel sentences from comparable corpora. The two approaches were used to construct a bootstrapping pipeline whose output was a set of sentence pairs in two languages that where then used to train a machine translation (MT) system. We showed that the trained MT system was able to outperform a MT system trained on a bitext, i.e. a human annotated set of parallel sentences. Since generating parallel sentences through human annotation is a time and costly process with our bootstrapping pipeline we were able to reduce the cost of building MT systems.

To efficiently evaluate topic models on large document collections we introduced two evaluation measures - Distributional Overlap (DO) and Histogram Slope Analysis (HSA). Both measures require more than 4 times less computation time than existing information retrieval (IR) metrics, such as mean average precision (MAP), and 25 times less time than perplexity. Across three evaluation settings we showed that HSA, while being more efficient to compute, achieves high linear and rank correlation with the existing IR evaluation measures: MAP, precision of the top 10 (P@10), and normalized discounted cumulative gain (NDCG). In addition, with the HSA variant, which we call random HSA (rHSA) we were able to perform efficient evaluation of document similarity models in large scholarly IR systems using evaluation sets of document that are automatically generated from download logs.

Lastly we introduced mlhLDA and mlhPLTM. These two topic models efficiently model topical variations across textual regions in mono- and multilingual collections using online VB. The efficiency of these models was demonstrated by their ability to offer better representations of collections with hierarchical document structures compared to online LDA and PLTM while requiring similar inference times. The efficiency of mlhLDA was also analyzed by comparing its running time to the batch VB implementation of LDA and showed that it is significantly faster. We also showed that mlhPLTM achieves better accuracy compared to oPLTM when evaluated on the task of retrieving document translation pairs.

8.1 Future Work

In this section we give an outline of suggested future work that could be done across the different approaches that we presented in this dissertation.

Efficient NN search in the probability simplex

We presented efficient NN search approach using two approximate techniques (LSH and k-d trees). Future experiments should include comparing the performance of various other approximate NN search approaches for the Euclidean distance. Across families of Dirichlet distributions with different hyperparameters and dimensions we observed that the theoretical bounds between He distance and JS divergence are in reality fairly loose. This raises the question whether the theoretical bounds could be reduced and exploring this question would be an interesting future work. In our work so far we've been using point estimates of the document-topic distributions to measure similarity between two documents in the topic space. In many instances the point estimate may not be a good approximation of the topic distribution such as in queries or other short documents. In such instances rather than computing divergence between two point estimates one should explore directly computing similarity between two distributions using a Bayesian approach.

Bootstrapping Translation Detection and Sentence Extraction from Comparable Corpora

Our approach for extracting parallel sentences was demonstrated using the English-Spanish language pair. In the future experiments should be performed across different language pairs and across different families of languages. Since the bootstrapping approach utilizes our two efficient techniques it would interesting to observe the performance of an MT system when our approach is used to process much larger collections and collections of documents in different domains.

Efficient Evaluation of Latent Variable Models of Text

We introduced DO and HSA by analyzing the relationships between the histograms computed over the scores of query relevant and non-relevant documents or as in the case of rHSA between similar and dissimilar documents. We believe that the evaluation approach based on histogram analysis could be further streamlined by analyzing only the characteristics of the histograms computed over the non-relevant or dissimilar documents. In the future one could explore the use of our histogram analysis to come up with a better dynamic threshold for the single ranked list problem that we observed in our bootstrapping approach for extracting parallel sentences. Furthermore one could explore developing a variation of rHSA that could be used to dynamically adjust the configuration of the PLTM when modeling a stream of multilingual data. Such a variation of rHSA could for example rely on minimum translation knowledge such as the number of overlapping words between two documents in different languages or documents that are published on the same date to automatically generate a set of similar documents in two languages that may be translations of each other.

Topic Models with Multi-Level Dirichlet Priors for Modeling Topical Variations across Textual Regions

Across both proposed models (mlhLDA and mlhPLTM) we used a single value for the concentration parameters which was assigned manually. While in the past approaches have been proposed that estimate the value of this parameter (e.g. [91]) in our case we focused only on estimating the base measure. In the future, experiments should be performed to assess the importance of approximating the concentration parameters in addition to the base measure. One should also consider evaluating both models on the task of retrieving document sections in mono- and multilingual collections. While evaluating retrieval of document sections is a difficult task due to lack of section level relevance judgments one could try to evaluate models on a document level. This would require performing section level retrieval and devise an approach for combing similarity scores across document sections in order to rank documents. mlhLDA should also be explored to model other collections where documents have hierarchical structure such as patent applications which consist of fixed set of fields, online discussion forums and on other tasks that deal with retrieving textual regions rather than the whole document, such as passage retrieval. mlhPLTM should also be explored on the task of modeling comparable corpora and evaluated on tasks that involve retrieving translations of textual regions (e.g. Europarl speeches) with small number of words. While in our work we dealt with processing data in a single stream performance analysis of these models should be expanded to comparing it with existing approaches that use parallel architectures. The performance of both models should also be evaluated in terms of the inference time and the size of the vocabulary used to represent documents in the topic space. While our assumption are that we are dealing with a single stream of data it is worth comparing the performance of mlhLDA with parallel implementations such as TiLDA [62] in terms of terms of how well does the model representation the collection and the number of compute cycles required to infer topics onto the collection.

APPENDIX

LIST OF ABBREVIATIONS

ADS	Astrophysics Data System
ApJ	Astrophysical Journal
BLEU	bilingual evaluation understudy
BNB	Bayesian Naive Bayes
CD	Consecutively downloaded
CfA	Center for Astrophysics
CIIR	Center for Intelligent Information Retrieval
CLIR	Cross-language information retrieval
Cos	Cosine
DAGs	Directed acyclic graphs
DBN	Deep belief network
DO	Distributional Overlap
E2LSH	Exact Euclidean locality sensitive hashing
ELBO	Evidence lower bound
EM	Expectation-Maximization

EP	Europarl
Eu	Euclidean
GW	Gigaword
He	Hellinger
HMM	Hidden Markov model
HSA	Histogram Slope Analysis
idf	Inverse document frequency
IR	Information Retrieval
JS	Jensen-Shannon
k-d	K-dimensional
KL	Kullback-Leibler
LDA	Latent Dirichlet allocation
LM	Language model
LSH	Locality sensitive hashing
LSI	Latent Semantic Indexing
MAP	Mean average precision
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
mlhLDA	Multi-level hyperpiors latent Dirichlet allocation
mlhPLTM	Multi-level hyperpriors polylingual topic model

MRF	Markov random field
MT	Machine translation
NB	Naive Bayes
NC	News Commentary
NDCG	Normalized discounted cumulative gain
NIH	National Institutes of Health
NLP	Natural language processing
NN	Nearest-neighbor
OCD	Overlapping Cosine distance
oPLTM	Online polylingual topic model
P@1	precision of the top rank
P@5	Precision of the top 5
P@10	Precision of the top 10
PCA	Principal component analysis
pdf	Probability density function
PLSA	Probabilistic latent semantic analysis
PLSI	Probabilistic latent semantic indexing
PLTM	Polylingual topic model
pmf	Probability mass function
PMI	Pointwise mutual information

PRF	Pseudo relevance feedback
QL	Query likelihood
R@5	Recall of the top 5
R@10	Recall of the top 10
RG	Randomly generated
rHSA	Random Histogram Slope Analysis
RM	Relevance model
ROC	relative operating characteristic
SDM	Sequential dependence model
SMT	Statistical machine translation
tf	Term frequency
TNG	Topical n-gram model
TREC	Text REtrieval Conference
USPTO	United States Patent and Trademark Office
VB	Variational Bayes
WER	word error rate

BIBLIOGRAPHY

[1] Abdul-Rauf, Sadaf, and Schwenk, Holger. On the use of comparable corpora to improve SMT performance. In *EACL* (2009), pp. 16–23.

[2] Aletras, Nikolaos, and Stevenson, Mark. Evaluating topic coherence using distributional semantics. In *IWCS* (2013), pp. 13–22.

[3] Aly, Mohamed, Pruhs, Kirk, and Chrysanthis, Panos K. Kddcs: A load-balanced in-network data-centric storage scheme for sensor networks. In *CIKM* (2006), pp. 317–326.

[4] Anderton, Jesse, Bashir, Maryam, Pavlu, Virgil, and Aslam, Javed A. An analysis of crowd workers mistakes for specific and complex relevance assessment task. In *CIKM* (2013), pp. 1873–1876.

[5] Andoni, Alexandr, Datar, Mayur, Immorlica, Nicole, Indyk, Piotr, and Mirrokni, Vahab. Locality-sensitive hashing using stable distributions. In *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2005, pp. 61–72.

[6] Andoni, Alexandr, and Indyk, Piotr. *LSH Algorithm and Implementation (E2LSH)*, 2005. `http://www.mit.edu/~andoni/LSH`.

[7] Andrzejewski, David, and Buttler, David. Latent topic feedback for information retrieval. In *KDD* (2011), pp. 600–608.

[8] Armstrong, Timothy G., Moffat, Alistair, Webber, William, and Zobel, Justin. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *CIKM* (2009), pp. 601–610.

[9] Arya, Sunil, and Mount, David M. Approximate nearest neighbor queries in fixed dimensions. In *SODA* (1993), pp. 271–280.

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

[11] Bentley, Jon Louis. Multidimensional binary search trees used for associative searching. *CACM 18*, 9 (1975), 509–517.

[12] Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., USA, 2006.

[13] Blei, David. lda-c. `https://github.com/Blei-Lab/lda-c`. Accessed January 15, 2016.

[14] Blei, David M., and Lafferty, John D. Dynamic topic models. In *ICML* (2006), pp. 113–120.

[15] Blei, David M., and Lafferty, John D. A correlated topic model of science. *AAS 1*, 1 (2007), 17–35.

[16] Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet allocation. *JMLR 3* (2003), 993–1022.

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

[18] Boyd-Graber, Jordan, and Resnik, Philip. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *EMNLP* (2010), pp. 45–55.

[19] Brown, Peter F., Pietra, Vincent J. Della, Mercer, Robert L., Pietra, Stephen A. Della, and Lai, Jennifer C. An estimate of an upper bound for the entropy of English. *Computational Linguistics 18*, 1 (1992), 31–40.

[20] Brown, Peter F., Pietra, Vincent J. Della, Pietra, Stephen A. Della, and Mercer, Robert L. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics 19*, 2 (1993), 263–311.

[21] Canini, Kevin R., Shi, Lei, and Griffiths, Thomas L. Online inference of topics with latent Dirichlet allocation. In *AISTATS* (2009), pp. 65–72.

[22] Cartright, Marc-Allen, Allan, James, Lavrenko, Victor, and McGregor, Andrew. Fast query expansion using approximations of relevance models. In *CIKM* (2010), pp. 1573–1576.

[23] Cettolo, Mauro, Federico, Marcello, and Bertoldi, Nicola. Mining parallel fragments from comparable texts. In *IWSLT* (2010), pp. 227–234.

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

[25] Charikar, Moses S. Similarity estimation techniques from rounding algorithms. In *STOC* (2002), pp. 308–388.

[26] Cormack, Gordon V., Smucker, Mark D., and Clarke, Charles L. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval 14*, 5 (2011), 441–465.

[27] Cover, Thomas M., and Thomas, Joy A. *Elements of Information Theory*. John Wiley, 1991.

[28] Croft, W. Bruce. A model of cluster searching based on classification. *Information Systems 5*, 3 (1980), 189–195.

[29] Croft, W. Bruce. Combining approaches to information retrieval. In *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*. Kluwer Academic Publishers, 2000, ch. 1, pp. 1–36.

[30] Croft, W. Bruce, Metzler, Donald, and Strohman, Trevor. *Search Engines: Information Retrieval in Practice*, 1st ed. Addison-Wesley Publishing Company, USA, 2009.

[31] Csiszár, Imre. Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems. *Annals of Statistics 19*, 4 (1991), 2032–2066.

[32] Dean, Jeffrey, and Ghemawat, Sanjay. Mapreduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008), 107–113.

[33] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, O. W., and Harshinan, R. A. Indexing by latent semantic analysis. *JASIS 41*, 6 (1990), 391–407.

[34] Duda, R. O., and Hart, P. E. *Pattern Classification and Scene Analysis*. John Willey & Sons, New York, 1973.

[35] Eisenstein, Jacob, O'Connor, Brendan, Smith, Noah A., and Xing, Eric P. A latent variable model for geographic lexical variation. In *EMNLP* (2010), pp. 1277–1287.

[36] Enright, Jessica, and Kondrak, Grzegorz. A fast method for parallel document identification. In *NAACL/HLT* (2007), pp. 29–32.

[37] Fernández, Miriam, Vallet, David, and Castells, Pablo. Probabilistic score normalization for rank aggregation. In *ECIR* (2006), pp. 553–556.

[38] Friedman, J. H., Bentley, J. L., and Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software 3*, 3 (1977), 209–226.

[39] Fukumasu, Kosuke, Eguchi, Koji, and Xing, Eric P. Symmetric correspondence topic models for multilingual text analysis. In *NIPS* (2012), pp. 1286–1294.

[40] Fung, Pascale, and Cheung, Percy. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *EMNLP* (2004), pp. 57–63.

[41] Gahbiche-Braham, Souhir, Bonneau-Maynard, Hélène, and Yvon, François. Two ways to use a noisy parallel news corpus for improving statistical machine translation. In *the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web* (2011), pp. 44–51.

[42] Galago. Galago is a toolkit for experimenting with text search. `http://www.lemurproject.org/galago.php`. Accessed January 15, 2016.

[43] Griffiths, T. L., and Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences 101*, Suppl. 1 (2004), 5228–5235.

[44] Griffiths, Thomas L., Steyvers, Mark, Blei, David M., and Tenenbaum, Joshua B. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, vol. 17. MIT Press, 2005, pp. 537–544.

[45] Guha, Sudipto, McGregor, Andrew, and Venkatasubramanian, Suresh. Streaming and sublinear approximation of entropy and information distances. In *SODA* (2006), pp. 733–742.

[46] Hall, David L. W., Jurafsky, Daniel, and Manning, Christopher D. Studying the history of ideas using topic models. In *EMNLP* (2008), pp. 363–371.

[47] Hearst, Marti A., and Pedersen, Jan O. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *SIGIR* (1996), pp. 76–84.

[48] Hoang, Cuong, Le, Anh-Cuong, Nguyen, Phuong-Thai, Pham, Son Bao, and Ho, Tu Bao. An efficient framework for extracting parallel sentences from non-parallel corpora. *Fundamenta Informaticae - Computing and Communication Technologies 130*, 2 (2014), 179–199.

[49] Hoffman, Matthew, Blei, David, and Bach, Francis. Online learning for latent Dirichlet allocation. In *NIPS* (2010), pp. 856–864.

[50] Hoffman, Matthew D. onlineldavb. `https://github.com/Blei-Lab/onlineldavb`. Accessed January 15, 2016.

[51] Hofmann, Thomas. Probabilistic latent semantic indexing. In *SIGIR* (1999), pp. 50–57.

[52] Hong, Wenxing, Li, Lei, and Li, Tao. Product recommendation with temporal dynamics. *Expert Systems with Applications 39*, 16 (2012), 12398–12406.

[53] Indyk, Piotr, and Motwani, Rajeev. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC* (1998), pp. 604–613.

[54] Jameel, Shoaib, and Lam, Wai. An unsupervised topic segmentation model incorporating word order. In *SIGIR* (2013), pp. 203–212.

[55] Jammalamadaka, Nataraj, Zisserman, Andrew, Eichner, Marcin, Ferrari, Vittorio, and Jawahar, C. V. Video retrieval by mimicking poses. In *ICMR* (2012), pp. 34:1–34:8.

[56] Jansen, Aren, and Durme, Ben Van. Efficient spoken term discovery using randomized algorithms. In *ASRU* (2011), pp. 401–406.

[57] Jardine, N., and van Rijsbergen, C. J. The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval 7* (1971), 217–240.

[58] Joachims, Thorsten. Optimizing search engines using clickthrough data. In *KDD* (2002), pp. 133–142.

[59] Jordan, Michael I., Ghahramani, Zoubin, Jaakkola, Tommi S., and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine Learning 37*, 2 (1999), 183–233.

[60] Kazai, Gabriella, Kamps, Jaap, and Milic-Frayling, Natasa. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information Retrieval 16*, 2 (2013), 138–178.

[61] Ke, Yan, Sukthankar, Rahul, and Huston, Larry. An efficient parts-based near-duplicate and sub-image retrieval system. In *MULTIMEDIA* (2004), pp. 869–876.

[62] Kim, Do-Kyum, Voelker, Geoffrey, and Saul, Lawrence K. A variational approximation for topic modeling of hierarchical corpora. In *ICML* (2013), vol. 28, pp. 55–63.

[63] Klementiev, Alexandre, Irvine, Ann, Callison-Burch, Chris, and Yarowsky, David. Toward statistical machine translation without parallel corpora. In *EACL* (2012), pp. 130–140.

[64] Klementiev, Alexandre, Roth, Dan, and Small, Kevin. An unsupervised learning algorithm for rank aggregation. In *ECML* (2007), pp. 616–623.

[65] Koehn, Philipp. Europarl: A parallel corpus for statistical machine translation. In *MT Summit* (2005), pp. 79–86.

[66] Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondřej, Constantin, Alexandra, and Herbst, Evan. Moses: Open source toolkit for statistical machine translation. In *ACL on Interactive Poster and Demonstration Sessions* (2007), pp. 177–180.

[67] Krstovski, Kriste, and Smith, David A. A minimally supervised approach for detecting and ranking document translation pairs. In *WMT* (2011), pp. 207–216.

[68] Kurtz, Michael J., Eichhorn, Guenther, Accomazzi, Alberto, Grant, Carolyn S., Murray, Stephen S., and Watson, Joyce M. The NASA Astrophysics Data System: Overview. *Astronomy and Astrophysics Supplement Series 143* (2000), 41–59.

[69] Langford, John, Li, Lihong, and Zhang, Tong. Sparse online learning via truncated gradient. *JMLR 10* (2009), 777–801.

[70] Lavrenko, Victor, and Allan, James. Real-time query expansion in relevance models. Tech. Rep. IR-473, University of Massachusetts Amherst, 2006.

[71] Lavrenko, Victor, and Croft, W. Bruce. Relevance based language models. In *SIGIR* (2001), pp. 120–127.

[72] Lease, Matthew, and Yilmaz, Emine. Crowdsourcing for information retrieval. *SIGIR Forum 45*, 2 (2012), 66–75.

[73] Lee, Joon Ho. Combining multiple evidence from different properties of weighting schemes. In *SIGIR* (1995), pp. 180–188.

[74] Liao, I-En, Hsu, Wen-Chiao, Cheng, Ming-Shen, and Chen, Li-Ping. A library recommender system based on a personal ontology model and collaborative filtering technique for english collections. *The Electronic Library 28*, 3 (2010), 386–400.

[75] Lin, Jianhua. Divergence measures based on Shannon entropy. *IEEE Transactions on Information Theory 37*, 1 (1991), 145–151.

[76] Ling, Wang, Xiang, Guang, Dyer, Chris, Black, Alan, and Trancoso, Isabel. Microblogs as parallel corpora. In *ACL* (2013), pp. 176–186.

[77] Liu, Xiaoyong, and Croft, W. Bruce. Cluster-based retrieval using language models. In *SIGIR* (2004), pp. 186–193.

[78] Lupu, Mihai, and Hanbury, Allan. Patent retrieval. *Foundations and Trends in Information Retrieval 7*, 1 (2013), 1–97.

[79] Lupu, Mihai, Huang, Jimmy, Zhu, Jianhan, and Tait, John. TREC-CHEM: Large scale chemical information retrieval evaluation at TREC. *SIGIR Forum 43*, 2 (2009), 63–70.

[80] Lupu, Mihai, Mayer, Katja, Tait, John, and Trippe, Anthony J. *Current Challenges in Patent Information Retrieval*, 1st ed. Springer Publishing Company, 2011.

[81] Manning, Christopher D., and Schütze, Hinrich. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[82] Marsolo, Keith, Parthasarathy, Srinivasan, and Ramamohanarao, Kotagiri. Structure-based querying of proteins using wavelets. In *CIKM* (2006), pp. 24–33.

[83] McCallum, Andrew Kachites. *MALLET: A Machine Learning for Language Toolkit*, 2002. `http://mallet.cs.umass.edu`.

[84] Mimno, David, Hoffman, Matthew D., and Blei, David M. Sparse stochastic inference for latent Dirichlet allocation. In *ICML* (2012), pp. 1599–1606.

[85] Mimno, David, Wallach, Hanna, Naradowsky, Jason, Smith, David A., and McCallum, Andrew. Polylingual topic models. In *EMNLP* (2009), pp. 880–889.

[86] Minka, Thomas P. Estimating a Dirichlet distribution. Tech. rep., MIT, 2000.

[87] Moore, Robert C. Fast and accurate sentence alignment of bilingual corpora. In *AMTA* (2002), pp. 135–144.

[88] Mount, David M., and Arya, Sunil. *ANN: A Library for Approximate Nearest Neighbor Searching*, 2010. `http://www.cs.umd.edu/~mount/ANN`.

[89] Munteanu, Dragos Stefan, and Marcu, Daniel. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics 31*, 4 (2005), 477–504.

[90] Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. The MIT Press, USA, 2012.

[91] Neal, Radford M. Slice sampling. *Annals of statistics 31*, 3 (2003), 705–741.

[92] Newman, David, Lau, Jey Han, Grieser, Karl, and Baldwin, Timothy. Automatic evaluation of topic coherence. In *NAACL/HLT* (2010), pp. 100–108.

[93] Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: A method for automatic evaluation of machine translation. In *ACL* (2002), pp. 311–318.

[94] Patent & Trademark Office, United States. Patent full-text databases. `http://patft.uspto.gov`. Accessed January 15, 2016.

[95] Pearl, Judea. Reverend Bayes on inference engines: A distributed hierarchical approach. In *The Second National Conference on Artificial Intelligence* (1982), pp. 133–136.

[96] Petrović, Saša, Osborne, Miles, and Lavrenko, Victor. Streaming first story detection with application to Twitter. In *NAACL/HLT* (2010), pp. 181–189.

[97] Platt, John, Toutanova, Kristina, and tau Yih, Wen. Translingual document representations from discriminative projections. In *EMNLP* (2010), pp. 251–261.

[98] Power, David M. W. Applications and explanations of Zipf's law. In *NeMLaP3/CoNLL* (1998), pp. 151–160.

[99] Quirk, Chris, U, Raghavendra Udupa, and Menezes, Arul. Generative models of noisy translations with applications to parallel fragment extraction. In *MT Summit* (2007), pp. 321–327.

[100] Rao, C. Radhakrishna. Diversity: Its measurement, decomposition, apportionment and analysis. *Sankhyā: The Indian Journal of Statistics 44*, A1 (1982), 1–22.

[101] Ravichandran, Deepak, Pantel, Patrick, and Hovy, Eduard. Randomized algorithms and NLP: Using locality sensitive hash function for high speed noun clustering. In *ACL* (2005), pp. 622–629.

[102] Resnick, Paul, and Varian, Hal R. Recommender systems. *Communications of the ACM 40*, 3 (1997), 56–58.

[103] Rijsbergen, C. J. Van. *Automatic Information Structuring and Retrieval*. PhD thesis, University of Cambridge, 1972.

[104] Rijsbergen, C. J. Van, and Croft, W. Bruce. Document clustering: An valuation of some experiments with the Cranfield 1400 collection. *Information Processing & Management 11*, 5.7 (1975), 171–182.

[105] Shaw, Joseph A., and Fox, Edward A. Combination of multiple searches. In *TREC-2* (1994), pp. 243–252.

[106] Shih, Ya-Yueh, and Liu, Duen-Ren. Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands. *Expert Systems with Applications 35*, 1 (2008), 350–360.

[107] Smith, Jason R., Quirk, Chris, and Toutanova, Kristina. Extracting parallel sentences from comparable corpora using document level alignment. In *NAACL/HLT* (2010), pp. 403–411.

[108] Smucker, Mark D., and Allan, James. A new measure of the cluster hypothesis. In *ICTIR* (2009), pp. 281–288.

[109] Smucker, Mark D., Allan, James, and Carterette, Ben. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM* (2007), pp. 623–632.

[110] Stevens, Keith, Kegelmeyer, Philip, Andrzejewski, David, and Buttler, David. Exploring topic coherence over many models and many topics. In *EMNLP* (2012), pp. 952–961.

[111] Steyvers, Mark, and Griffiths, Tom. Probabilistic topic models. *Handbook of latent semantic analysis 427*, 7 (2007), 424–440.

[112] Strohman, Trevor, Metzler, Donald, Turtle, Howard, and Croft, W. Bruce. Indri: A language model-based search engine for complex queries, 2005.

[113] Swets, John . A. Effectiveness of information retrieval methods. *American Documentation 20*, 1 (1969), 72–89.

[114] Tait, John, Harris, Christopher, and Lupu, Mihai, Eds. *PaIR '10: Proceedings of the 3rd international Workshop on Patent information retrieval* (2010).

[115] Talley, Edmund, Newman, David, Mimno, David, Herr, Bruce, Wallach, Hanna, Burns, Gully, Leenders, Miriam, and McCallum, Andrew. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods 8* (2011), 443–444.

[116] Tillmann, Christoph, and Xu, Jian-ming. A simple sentence-level extraction algorithm for comparable data. In *NAACL/HLT, Companion Volume: Short Papers* (2009), pp. 93–96.

[117] Tombros, Anastasios, Villa, Robert, and Van Rijsbergen, C. J. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing Management 38*, 4 (2002), 559–582.

[118] Topsøe, Flemming. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory 46*, 4 (2000), 1602–1609.

[119] Ture, Ferhan, Elsayed, Tamer, and Lin, Jimmy. No free lunch: Brute force vs. locality-sensitive hashing for cross-lingual pairwise similarity. In *SIGIR* (2011), pp. 943–952.

[120] Ture, Ferhan, and Lin, Jimmy. Why not grab a free lunch?: mining large corpora for parallel sentences to improve translation modeling. In *NAACL/HLT* (2012), pp. 626–630.

[121] Uszkoreit, Jakob, Ponte, Jay M., Popat, Ashok C., and Dubiner, Moshe. Large scale parallel document mining for machine translation. In *COLING* (2010), pp. 1101–1109.

[122] Voorhees, Ellen M. The cluster hypothesis revisited. In *SIGIR* (1985), pp. 188–196.

[123] Wallach, Hanna M. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

[124] Wallach, Hanna M., Mimno, David, and McCallum, Andrew. Rethinking LDA: Why priors matter. In *NIPS* (2009), pp. 1973–1981.

[125] Wallach, Hanna M., Murray, Iain, Salakhutdinov, Ruslan, and Mimno, David. Evaluation methods for topic models. In *ICML* (2009), pp. 1105–1112.

[126] Wang, Xuerui, and McCallum, Andrew. Topics over time: A non-Markov continuous-time model of topical trends. In *KDD* (2006), pp. 424–433.

[127] Wang, Xuerui, McCallum, Andrew, and Wei, Xing. Topical N-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM* (2007), pp. 697–702.

[128] Wei, Xing, and Croft, W. Bruce. LDA-based document models for ad-hoc retrieval. In *SIGIR* (2006), pp. 178–185.

[129] Xue, Xiaobing, and Croft, W. Bruce. Transforming patents into prior-art queries. In *SIGIR* (2009), pp. 808–809.

[130] Yang, Cheng. Efficient acoustic index for music retrieval with various degrees of similarity. In *MULTIMEDIA* (2002), pp. 584–591.

[131] Yi, Xing, and Allan, James. A comparative study of utilizing topic models for information retrieval. In *ECIR* (2009), pp. 29–41.

[132] Zeng, Jia, Cao, Xiao-Qin, and Liu, Zhi-Qiang. Residual belief propagation for topic modeling. In *ADMA* (2012), pp. 739–752.

[133] Zhou, Dong, and Wade, Vincent. Latent document re-ranking. In *EMNLP* (2009), pp. 1571–1580.