INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

SCHOOL OF ENGINEERING AND INFORMATION TECHNOLOGIES

GRADUATE PROGRAM IN MECHATRONICS AND INFORMATION TECHNOLOGIES

DOCTOR OF PHILOSOPHY

IN

INFORMATION TECHNOLOGIES AND COMMUNICATIONS MAJOR IN INTELLIGENT SYSTEMS

LARGE SCALE TOPIC MODELING USING SEARCH QUERIES: AN INFORMATION-THEORETIC APPROACH

BY:

EDUARDO H. RAMIREZ RANGEL

MONTERREY, N. L.　　DECEMBER OF 2010

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

SCHOOL OF ENGINEERING AND INFORMATION TECHNOLOGIES

GRADUATE PROGRAM IN MECHATRONICS AND INFORMATION
TECHNOLOGIES

DOCTOR OF PHILOSOPHY

IN

INFORMATION TECHNOLOGIES AND
COMMUNICATIONS MAJOR IN INTELLIGENT SYSTEMS

LARGE SCALE TOPIC MODELING USING SEARCH QUERIES: AN INFORMATION-THEORETIC APPROACH

BY:

EDUARDO H. RAMIREZ RANGEL

MONTERREY, N. L. DECEMBER OF 2010

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

SCHOOL OF ENGINEERING
DIVISION OF MECHATRONICS AND INFORMATION TECHNOLOGIES
GRADUATE PROGRAMS

DOCTOR OF PHILOSOPHY
in
INFORMATION TECHNOLOGIES AND COMMUNICATIONS
MAJOR IN INTELLIGENT SYSTEMS

Large Scale Topic Modeling Using Search Queries: An information-theoretic approach

By

Eduardo H. Ramírez Rangel

DEC 2010

Large Scale Topic Modeling Using Search Queries: An information-theoretic approach

A dissertation presented by

Eduardo H. Ramírez Rangel

Submitted to the
Graduate Programs in Mechatronics and Information Technologies
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Information Technologies and Communications
Major in Intelligent Systems

Thesis Committee:

Dr. Ramón F. Brena	-	Tecnológico de Monterrey
Dra. Alma Delia Cuevas	-	Escuela Superior de Cómputo, IPN
Dr. Jose Ignacio Icaza	-	Tecnológico de Monterrey
Dr. Leonardo Garrido	-	Tecnológico de Monterrey
Dr. Randy Goebel	-	University of Alberta

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey
Dec 2010

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Division of Mechatronics and Information Technologies
Graduate Program

The committee members, hereby, certify that have read the dissertation presented by Eduardo H. Ramírez Rangel and that it is fully adequate in scope and quality as a partial requirement for the degree of **Doctor of Philosophy in Information Technologies and Communications**, with a major in **Intelligent Systems**.

Dr. Ramón F. Brena
Tecnológico de Monterrey
Principal Advisor

Dra. Alma Delia Cuevas
Escuela Superior de Cómputo, IPN
Committee Member

Dr. Jose Ignacio Icaza
Tecnológico de Monterrey
Committee Member

Dr. Leonardo Garrido
Tecnológico de Monterrey
Committee Member

Dr. Randy Goebel
University of Alberta
Committee Member

Dr. José Luis Gordillo
Director of Ph.D. Program on ITC
School of Engineering

Copyright Declaration

I, hereby, declare that I wrote this dissertation entirely by myself and, that, it exclusively describes my own research.

Eduardo H. Ramírez Rangel
Monterrey, N.L., México
Dec 2010

Acknowledgements

I'd like to acknowledge and express my gratitude to some of the many people that have contributed in some way to this work. Very especially I thank my advisor, Dr. Ramon Brena for his continuous support and confidence during the course of my studies. Also, I'd like to thank my dissertation committee members: Alma Delia Cuevas (ESCOM IPN), Leonardo Garrido (ITESM); Jose I. Icaza (ITESM) and Randy Goebel (University of Alberta) for the time devoted to review and their valuable suggestions to improve this work.

Thanks also to the researchers that have sponsored me for internships and mentored me while there: Yi-Min Wang and Li-Wei He from Microsoft Research, Ricardo Baeza-Yates, Hugo Zaragoza and Carlos Castillo from Yahoo! Research. Particularly, I thank Yi-Min for the feedback related to my overall work methodology and for helping me better understand the value of good metrics. Besides being extremely helpful to improving my research skills those internships provided me with chances to interact firsthand with leading researchers like Susan Dumais, who suggested me to study PLSI in the first place. Also, thanks to the Yahoo! Research internship I could learn about the co-clustering algorithm from the work of Fabrizio Silvestri, which later became a key component of the presented framework.

Also, I say thanks to the colleagues with whom I engaged in discussions about this work, provided suggestions and troubleshooted me in topics of their own specialities: Sara Garza, Francisco Guzman, Marco Peña, Emmanuel Martinez (ITESM), Shirish Tatikonda (Ohio State), Alejandro Figueroa (DFKI), Iva Kozakova (Vanderbilt University) and Eric Nichols (Indiana University). Also I thank the members from ITESM staff: Edith Villasana, Dora Elia Juarez and Isabel Cerda for always kindly helping me get all my scolarship, travel and administrative work done in time.

Finally and very especially, I thank Davide Magatti and Fabio Stella from Università degli Studi di Milano-Bicocca for initiating and keeping up with the productive collaboration we have done in the area of topic model validation metrics, the metrics we have developed and published together were fundamental to round up the results presented in this work.

Large Scale Topic Modeling Using Search Queries: An information-theoretic approach

by

Eduardo H. Ramírez Rangel

Abstract

Creating topic models of text collections is an important step towards more adaptive information access and retrieval applications. Such models encode knowledge of the topics discussed on a collection, the documents that belong to each topic and the semantic similarity of a given pair of topics. Among other things, they can be used to focus or disambiguate search queries and construct visualizations to navigate across the collection. So far, the dominant paradigm to topic modeling has been the Probabilistic Topic Modeling approach in which topics are represented as probability distributions of terms, and documents are assumed to be generated from a mixture of random topics. Although such models are theoretically sound, their high computational complexity makes them difficult to use in very large scale collections.

In this work we propose an alternative topic modeling paradigm based on a simpler representation of topics as freely overlapping clusters of semantically similar documents, that is able to take advantage of highly-scalable clustering algorithms. Then, we propose the Query-based Topic Modeling framework (QTM), an information-theoretic method that assumes the existence of a "golden" set of queries that can capture most of the semantic information of the collection and produce models with maximum semantic coherence. The QTM method uses information-theoretic heuristics to find a set of "topical-queries" which are then co-clustered along with the documents of the collection and transformed to produce overlapping document clusters. The QTM framework was designed with scalability in mind and is able to be executed in parallel over commodity-class machines using the Map-Reduce approach.

Then, in order to compare the QTM results with models generated by other methods we have developed metrics that formalize the notion of semantic coherence using probabilistic concepts and the familiar notions of recall and precision. In contrast to traditional clustering metrics, the proposed metrics have been generalized to validate overlapping and potentially incomplete clustering solutions using multi-labeled corpora. We use them to experimentally validate our query-based approach, showing that models produced using selected queries outperform the ones produced using the collection vocabulary. Also, we explore the heuristics and settings that determine the performance of QTM and show that the proposed method can produce models of comparable, or even superior quality, than those produced with state of the art probabilistic methods.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

Now more than ever, many activities of life, work and business depend on the information stored on massive, fast-growing document collections. However, despite the impressive advances in Web retrieval technologies, searching the web or browsing extensive repositories are not simple tasks for many users. Very often users pose ineffective queries and need to reformulate them to better express their intent; unfortunately, producing effective query reformulations to achieve an information goal is not always a straightforward task [11], [33].

The very nature of human language counts among the causes of difficulty and dissatisfaction of users dealing with information access and retrieval systems like search engines or digital libraries. *Polysemy*, the language ability to relate a word to many senses, leads people to pose short and ambiguous queries. We as humans can easily detect ambiguity, infer the intended sense based on context and previous experience, or in case of doubt, ask for a clarification. However, retrieval systems are required to acknowledge the ambiguity [19] and adapt their response, whether by picking one of the senses of the query, or by presenting diversified results.

On the other hand, when queries get longer or users lack enough domain knowledge, it turns more likely that query terms may differ from the terms in the documents, thus making relevant documents less likely to be retrieved. This problem has been characterized by Furnas et. al [26], as the "vocabulary problem" or the "term mismatch problem" and is a consequence of *synonymy* in language, that is, multiple words with the same meaning. Currently, in order to deal with synonymy, the most skilled search users need to infer the words that may appear on their relevant documents and try different query variations using equivalent terms and expressions.

1.2 Computational Semantics Approaches

In general, there exist two major classes of solutions to the "term-mismatch" and the ambiguity problems reported in literature since the late 80's. One of the ultimate motivations and long-term goal of many of such developments, including the one presented in this work, is to evolve retrieval technologies from lexical matching towards semantic matching, that is, being able to retrieve documents that do not necessarily include the query terms but solve the information need.

The first class of works that we can broadly classify as "supervised", relies on human-crafted ontologies or linguistic databases like WordNet [40] to encode semantic relations between words and concepts. Such linguistic resources allow performing operations like computing semantic relatedness [43], disambiguating word-senses [64], [37] and expanding queries using synonyms [65], [39].

The second class of solutions is more general in nature and comprehends a family of unsupervised methods focused on the creation of statistical models of the collections. In general, they take as input the term and document co-ocurrence statistics and attempt to expose some form of hidden or "latent" semantic structure. An explicit general-purpose model of the semantic relations in the corpus can be used to answer queries based on semantic (rather than strictly lexical) similarity, but their applicability can be extended to different sort of semantically challenging problems, such as automatic translation [60], question answering [63] or semantic advertising [12], [16].

On this line of thought, one of the seminal works was *Latent Semantic Indexing* [20] (LSI). They proposed a vectorial representation of words and documents and used linear algebrà to create a spatial representation in which documents with similar terms appear close to each other. As LSI was criticized for lacking a theoretical foundation, Hofmann [32] proposed a probabilistic version of LSI, namely *Probabilistic Latent Semantic Indexing* (PLSI). PLSI and all subsequent methods like *Latent Dirichlet Allocation* [8] (LDA), work under the assumption that a document can be modeled as a mixture of hidden topics, and that those topics can be modeled as probability distributions over words. Then, some sort of parameter estimation algorithm (e.g. maximum likelihood estimation) is applied to the observed data to learn the parameters of the hidden topics. Authors like Griffiths and Steyvers have characterized this family of works as *Probabilistic Topic Models* [29] and [59].

The idea of modeling collections based on its topics and representing each topic as a probability distribution of terms is central to state of the art approaches and has additional benefits versus spatial representations. Also, it has been shown to be a good idea; by using probabilistic topic modeling methods it is possible to improve access to information in collections in different application scenarios, such as retrieval [35], [69], or collection browsing [7]. So, on the basis of such evidence, we may confidently state that creating a topic model of the collection is a necessary step towards more adaptive search engines and applications.

However, due to its high computational complexity the applicability of probabilistic topic modeling methods remains limited on large corpus. Moreover, as collection size increases unique scalability challenges emerge [4] [2]. For instance, algorithms require to be executed in parallel over large clusters of commodity-class machines and exhibit low computational complexity. In this operational scenario, scalability should be considered in design-time, not as an afterthought.

1.3 Discrete Topic Modeling

Therefore, in the aim of making semantic modeling feasible on large scale collections, in this work we propose an alternative topic modeling method based on a simplified representation of topics as freely overlapping sets or clusters of semantically similar documents. By simplifying the notion of topic, the problem of Probabilistic Topic Modeling can be reformulated as one of *"Discrete Topic Modeling"* and essentially transforming it into an overlapping clustering problem, allowing us to take advantage of a broad array of techniques existing in the literature.

Among the many clustering methods available, a fairly recent innovation has been found to be a very natural match for the goals of this research on virtue of its information-theoretic nature and algorithmic properties. The Co-Clustering method, proposed by Dhillon et. al [22] is a distributional clustering algorithm that simultaneously clusters data over a two-dimensional matrix while maximizing the preserved mutual information between the original and the clustered data.

In contrast to LDA, the Co-Clustering algorithm exhibits great scalability and has been used successfully to process very large collections [47] and parallel implementations exist [42]. For instance, Puppin et. al [46] proposed an alternative collection representation, the query-vector document model (QV), where each document is represented as a vector of the queries that it serves. They used a set of popular queries obtained from search engine logs to locate together the documents that will likely be retrieved by similar queries.

1.4 Topic Model Validation Metrics

However, when approaching the clustering problem from the collection modeling perspective we found that the evaluation of the quality of the models was still a challenging task. Validation of probabilistic topic models was done typically by user studies, which are costly and hard to reproduce [13]. On the other hand, the metrics used to validate traditional, non-overlapping, hard clusters are not well suited to validate an overlapping and potentially incomplete clustering [70], [21].

Therefore, we have defined a novel set of metrics inspired on the notion of "semantic coherence" interpreted as the probability of randomly sampling two documents from the same topic or class after having randomly selected a cluster. The proposed metrics use multi-labeled corpora to measure the quality of overlapping and incomplete topic models in an inexpensive and repeatable way. In fact, the proposed validation metrics as such represent one of the main contributions of this work.

1.5 Query-Based Topic Modeling

The key idea behind our approach to collection modeling is the assumption of the existence of a "golden" set of highly informative queries, defined as the *"topical-queries"*, such that when co-clustered along with the documents, they result in a model of maximum semantic coherence. We provide experimental evidence to show that a set of queries generated and selected using simple information-theoretic heuristics is superior to the collection vocabulary in terms of the semantic coherence of the resulting clusters. For that reason, we place a special emphasis on the analysis of the heuristics that can be used to generate, evaluate, and select this set of "topical-queries", and efficiently build collection models.

The Query-Based Topic Modeling (QTM) framework presented in this work, is a scalable, information-theoretic, discrete topic modeling method that uses search queries to incorporate semantic information into the modeling process. Generally speaking, the QTM method comprises two broad phases. In the first phase, the collection is processed and heuristics are used to generate and select a set of candidate topical queries. Then, using the query-vector document model, the queries and documents are co-clustered. The resulting clusters can be used to produce two kinds of collection models: a) non-overlapping models, that are essentially "hard" document clusters and b) overlapping topic models, similar to those produced with LDA, that rely on an estimation of document-topic probabilities.

In terms of model quality, we also show experimentally that by using the QTM approach it is possible to produce models of comparable quality to those produced by state of the art methods such as LDA. However, in contrast to other topic modeling methods, the QTM framework can produce the models with very high scalability through parallel execution, as QTM was designed using the Map-Reduce architectural style on each of its components.

1.6 Contributions

The main contributions of this work can be summarized as follows.

1. An alternative, discrete, formulation of the topic-modeling problem.
2. A set of probabilistic metrics to evaluate the quality of topic models in a repeatable and inexpensive way using multi-labeled corpora.
3. A set of information-theoretic heuristics that can be used to evaluate and identify topics in large collections.

4. A scalable query-based modeling framework that can produce models of comparable quality of state-of-the-art methods in very large collections.

1.7 Document organization

The rest of this document is organized as follows:

- In Chapter 2, we recall some required retrieval concepts and present the major unsupervised approaches to extract latent topic structures. Also we describe in more detail the family of distributional clustering techniques, such as those used by QTM.
- In Chapter 3, we present formal problem statements for the main tasks approached in this work and present some of our hypothesis and research questions
- In Chapter 4, we describe the information-theoretic basis of our approach to the task of topical-query identification and perform a quantitative comparison of several query evaluation functions.
- In Chapter 5, we present an integrated overview of the Query-Based Topic Modeling (QTM) framework and algorithms.
- In Chapter 6, we describe in detail the probabilistic "semantic coherence" metrics used to evaluate the quality of topic modeling and soft-clustering methods.
- In Chapter 7, we present our main experimental findings, an analysis on the effects of the parameters of QTM on semantic coherence and compare our results to those generated with LDA.
- In Chapter 8, we drive our main conclusions an a schedule for future work.

Chapter 2

Background

2.1 Web Information Retrieval: Foundations and challenges

For some authors, the fundamental problem of Information Representation and Retrieval Systems is "how to obtain the right information for the right user at the right time" [15]. In the following sections we discuss the aspects of the IR systems that are more relevant to the purposes of this work.

2.1.1 Measuring Search Quality

The goal of the ad-hoc retrieval task is to find relevant documents within a collection for any given query. The ad-hoc retrieval task in the web usually involves short and ambiguous queries. There are two standard metrics to measure search effectiveness: recall and precision. The recall metric compares the number of relevant documents retrieved against the total number of relevant documents. On the other hand, precision measures how many of the retrieved documents are actually relevant. Relevancy is determined subjectively by a human judge and indeed brings some evaluation challenges, as it can vary in every user and every query. Thus, in order to overcome the problem of subjective judgments, the IR community has built standardized measurement datasets that incorporate a set of test queries as well as the predefined set of relevant documents that should return. An example of such benchmark is the TREC [66] collection, that albeit small, provide examples of queries associated to their human relevance judgments and is used as a way to objectively and automatically compare retrieval algorithms without requiring the same human judge to evaluate the results every time.

Precision and Recall are not the only quality metrics for retrieval systems, for instance, the Normalized Discounted Cumulative Gain (NDCG) metric proposed by Jarvelin & Kekalainen [34] is gaining acceptance among the Web IR community, and although is also based on human judgments, it may be used to automate relevance judgments and to use machine learning techniques to tune the system parameters. In contrast to pure precision/recall, NDCG also takes into account the position of the relevant results and a degree of relevance, so, the best score is obtained by an algorithm that presents the relevant results in decreasing order of relevance. On the other hand, the discounting factor is a matter of debate, as it can place excessive emphasis on the quality of the very top results only.

2.1.2 Classic IR Models

With clear and unambiguous queries, a human may easily determine if a document in the collection is relevant to a query. However, for IR systems, relevance needs to be approximated using a similarity function that computes a score for a given query and document representations. The nature of the relevance function and document representations is the essence of the retrieval model or strategy.

The basic data structure of the classic approaches to IR is a term-document matrix. The term-document matrix, assumes that each document is a "bag of words". According to the strategy each of the elements of the matrix contains a weight that indicates the relevance of the term in the specific document.

In the boolean model, the weights were constrained to be 1 or 0, where a term weight of 1 was an indication that the term indeed existed in the document. In other words, documents are represented as a set of words. When a query was posed, the system retrieved the documents containing a weight of 1 for each of the query terms. A clear limitation of this system is that only the documents containing all the query terms will be retrieved, also, there was not any distinction on how important were the terms in each documents, thus it was not possible to rank.

In order to overcome the limitations of boolean retrieval, Salton et. al [51] introduced the vector-space model (VSM). In the VSM, each document is represented as a vector of weights and each query is represented as a vector of terms assuming that a query and a document are similar if their vectors point in similar directions. For that matter, some monotonic vector function is applied, such as the angle or the inner product. The VSM also considers a different specification of term weights in the document representation based on the frequency of the collection frequency and in the relative frequency of the term in the document.

The underlying reasoning is that a term that appears in all (or many) documents in the collections embodies little information about a document, while a term that is included only in a few documents contains more information about them, this metric is known as the *Inverse Document Frequency [58]*, and is formally defined as:

$$idf(i) = log(\frac{n}{n_i})$$

Where n is the total number of documents in the collection and n_i is the number of document in which the term i occurs.

Finally, the weight for each term i in a document vector is defined also taking into account the frequency of the term in the document j, and is expressed as:

$$w_{ij} = tf(i,j)idf(i)$$

The previous expression is also known in literature as *tf*idf*. Interestingly, given its computational efficiency, the fundamental ideas of the "classic" retrieval models are still in use these days, albeit combined with several web-specific ranking features.

2.1.3 Web retrieval challenges

Although the Information Retrieval Discipline (IR) has been studied since the second half of the 20th century, the size and nature of the web has created a new set of challenges to the IR systems, which have been surveyed by Baeza-Yates et al [3]. For instance, given the size and its hyperlinked nature, what works well on small plain-text collections is not guaranteed to work on the web, moreover, it is not guaranteed to work at all. Besides, as ranking documents in the Web affects a myriad of business interests, many webmasters have placed effort on developing techniques to win the first places for a considerable number of queries. The illegitimate techniques that have been invented to climb the ranks have been catalogued as "Web spam". Some of the first surveys of the field have been developed by Gyongyi and Garcia-Molina [30].

Web Search Engines have evolved to become heavily distributed systems. They are required to perform distributed crawling, distributed indexing and query processing [2], also they are required to perform well enough to serve thousands of queries per seconds. Current architectures embrace massive parallelism, where the leading companies operate clusters of several thousands of commodity-class machines that maximizes price/performance ratio [4]. From the algorithmic standpoint, what this means to any retrieval technology is that it should be fully distributable, both in order to index the web corpus on a regular basis and maintain "freshness" as well as to serve queries in fractions of a second using a distributed index.

2.2 Identifying and Modeling Topics in Collections

Broadly speaking, the abstract problem that concerns us is that of identifying the latent topic structure of a document collection in such a way that may be used to solve other computational problems such as classification, filtering, translation or retrieval.

The methods greatly vary in their computational strategy, which depends on the initial assumptions of what a topic is, how topics relate to documents and how topics relate to each other. However, all the referred methods, including ours, share the baseline assumption that corpus statistics contain enough information to produce useful results without requirements of expert knowledge.

The former requirement mostly complies with the "statistical semantics" paradigm, defined by G. Furnas [25] as the study of *"how the statistical patterns of human word usage can be used to figure out what people mean, at least to a level sufficient for information access"*. In contrast to lexicon approaches, the statistical semantics philosophy pushes towards the development of fully automated cross-language corpus-based algorithms.

2.2.1 Latent Semantic Indexing

The Latent Semantic Indexing (LSI) method was presented in [23], [20], [5] as a linear algebra based solution to the "term mismatch" problem in retrieval. The LSI method leverages the term co-occurence in the term-document matrix and applies Singular Value Decomposition in order to create a reduced matrix, in which semantically related documents appear closer. This reduced matrix is usually referred as the *latent semantic space*.

The method begins by creating a term-document matrix A of i terms by j documents, in which each $a(i, j)$ cell contains the frequency of term i in document j. The SVD factorization of A may be expressed as:

$$A = U \cdot \Sigma \cdot V^t$$

For the sake of efficiency, the SVD is usually truncated to a k number of dimensions. The values of Σ (singular values) are sorted by magnitude and the top k are used as the latent semantic representation of A, all other values are set to 0. After defining k the top k columns in a new matrix U_k are kept and the top k rows in the matrix V^t. Terms may be compared by computing the inner product of the rows in U_k and documents may be compared comparing the columns in V_k^t.

Some concerns regarding the LSI method are the lack of theoretical foundations [8], [32] and the computational costs involved in the SVD computation, which is in the order of $O(N^2k^3)$ where N is the number of documents in a collection and k the number of terms, in addition, the LSI computations cannot be optimized by using inverted indexes, which are the base of current web retrieval systems.

In conclusion, the LSI technique is not well suited to be applied on large web collections, although it has been used to semantically cluster the top result pages in order to improve user interface. However, the LSI method provided strong evidence towards the development of semantic capabilities using frequency-based metrics.

2.2.2 Probabilistic Topic Modeling

In the following years Hofmann [32] proposed a probabilistic version of LSI, namely *Probabilistic Latent Semantic Indexing* (PLSI). PLSI and all the methods that followed its approach work under the assumption that a document can be modeled as mixture of a number of hidden topics, and that those topics can be represented as probability distributions over words. Then, some sort of parameter estimation algorithm is applied to the observed data to estimate the parameters of the hidden topics. In the case of PLSI, the kind of estimation performed is maximum likelihood estimation (MLE).

Later on, Blei et. al [8] proposed the *Latent Dirichlet Allocation* (LDA) which was shown to be a generalization of PLSI by Girolami and Kabán [27]. The key innovation in LDA was the introduction of fully generative semantics into the model formulation and thus allowing the problem to be treated by Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling. In LDA each topic is represented as a multinomial distribution over words and each document is represented as a random mixture of topics, sampled from a Dirichlet distribution. In order to learn the model a topic mixture is sampled from a Dirichlet distribution, then a topic is sampled from this distribution and samples a word from that topic. The process is repeated for every word of every document until the full collection is generated. It is assumed that the words are generated based on the mixture of topic proportions. The reported complexity of the LDA procedure using Gibbs sampling is $O(IKN)$ where I is the number of sampling iterations, K is the number of topics and N is the total number of word occurrences in the training corpus [38].

The generative modeling approach introduced by LDA quickly became very popular and several models have been based on it. In these new models, topics are "first-class citizens" because they are now explicitly represented using a probability distribution and the task is not only to estimate its parameters, but also to learn the correlations between topics or to learn a structure that may be hierarchical, such as in hLDA [6] and non-parametric. Also of particular interest is in *Pachinko Allocation* [36] PAM, which can learn arbitrary topic correlations using a Directed Acyclical Graph. The PAM model also presents algorithmic improvements into the sampling procedure by challenging the assumption of the randomness of the mixing proportions and extending the topic representation schema.

Previously discussed models extract latent classes or aspects from the documents, however the number of classes is usually predefined and the resulting topic structure is a predefined parameter of the system, so that neither the nature of the underlying topic structure nor their hierarchical or semantic proximity relations were major concerns. Recently Blei [7] applied variants of the LDA model to obtain correlated topic models of the *Science* magazine archives, and Griffits and Steyvers [28] applied a similar method to the PNAS[1] database. In both cases, they usually predefine a fixed number of topics (say 100), so if they run the analysis truncating the database in time, they could analyze the relative topic "hotness" in time.

Regarding the application of discussed model to retrieval, Wei and Croft created a retrieval model for the ad-hoc task based on an LDA topic model. When applied to the ad-hoc retrieval task, the LDA outperformed Cluster-Based retrieval [69].

2.3 Distributional clustering

The concept of distributional clustering was first introduced by Pereira et. al [44] and was motivated by the word-sense disambiguation problem. The goal of the original algorithm proposed was to cluster words depending on their different "senses". One of the key ideas that define this family of methods was to represent each individual to cluster as a conditional probability distribution and then, assign each individual to the most likely cluster using existing information-theoretic distribution similarity measures such as the Kullback-Leibler divergence [18]. Conceptually, an important contribution of distributional clustering algorithms is that they replace the usage of arbitrary distance or similarity functions in favor of more objective information preservation criteria.

[1] Proceedings of the National Academy of Science

2.3.1 Information bottleneck (IB)

Tishby, et. al [62] generalized the original distributional clustering problem presented in [44] as the problem of self-organizing the members of a set X, based on the similarity of their conditional distributions over the members of another set Y, namely $p(y|x)$. That is, every element $x_i \in X$ is represented by a vector of real numbers that contains the conditional probability distribution of x_i over another random variable Y, that is $x_i = \{p(y_1|x_i), p(y_2|x_i)...p(y_j|x_i)\}$.

So, they reformulated the distributional clustering problem as a compression problem by expressing it as that of finding a compact representation of $X = \{x_1, x_2...x_i\}$, denoted by $\hat{X} = \{\hat{x}_1, \hat{x}_2...\hat{x}_k\}$, where every $\hat{x} \in \hat{X}$ is a cluster of elements of X, that maximizes the mutual information about Y, $I(\hat{X}, Y)$, subject to a constraint on the mutual information between X and $\hat{X}$. The solution of the stated problem required a balance between the compactness of the representation, that implied minimizing $I(X, \hat{X})$, and the preservation of the mutual information that implied maximizing $I(\hat{X}, Y)$. Given its double-optimization nature the problem was identified as the *"Information bottleneck"* (IB). From a pure information-theoretic perspective, the IB method could also be described as the problem of finding the relevant information that a signal X provides about another signal Y. In simpler terms, and considering the document-clustering problem, the IB method finds a partition of the documents that preserves as much as possible the mutual information about the words, each document represented by a conditional probability distribution over the words of the collection.

Interestingly, the authors found that the IB problem had an exact analytical solution, and proposed a greedy, locally optimal, agglomerative clustering algorithm, namely the *Agglomerative Information Bottleneck,* (aIB) [55]. The aIB method had the advantage of being non-parametric with respect of the number of clusters but it came at very high computational cost, specifically its time complexity was in the order of $O(|X|^3)$. Such high complexity of the aIB was due to the fact that it required pair-wise comparisons between elements in order to perform the best possible merge at every step. Moreover, the agglomerative procedure was not guaranteed to converge to a global optimum.

Thus, in later works, Slonim et. al [54] presented a remarkably more efficient algorithm, the *Sequential Information Bottleneck* (sIB) and showed that it outperformed aIB in both clustering performance and computational complexity. In contrast to aIB, the sIB method required a number of clusters to be found and then starting from an initial random partition, at each step one element was re-assigned to the best possible cluster until no further improvements were possible.

Finally, with respect to the document clustering application, the same authors proposed the *Double Information Bottleneck* method (dIB) [56]. Notably, the dIB algorithm introduced the idea of clustering both dimensions X and Y to improve the overall result. In the dIB method, first the words Y are clustered and a set of word clusters $\hat{Y}$ that preserve information about the documents is found. Then, documents were clustered in relation to the discovered word clusters. A very relevant finding of this work is that by clustering the two dimensions the results improved and the dimensionality of the problem could be reduced. The dIB method however was based on aIB and thus it suffered from the same performance drawbacks.

2.3.2 Co-Clustering

The authors of the dIB method described above were aware that a natural generalization of their work consisted of achieving a simultaneous compression of both dimensions of the co-ocurrence matrix, however, such generalization was proposed by Dhillon, et. al. [22] although under slightly different settings. The Co-Clustering algorithm benefited from a fresh perspective and approached the problem of simultaneously clustering rows and columns from an input matrix representing the joint probability distribution of the variables $p(X,Y)$ while maximizing the preserved mutual information (or minimizing the mutual information loss) along both dimensions.

In contrast to the IB methods, the Co-Clustering algorithm required the number of rows and column clusters to be specified beforehand. By doing so, it decoupled the clustering problem from the model selection problem and thus it created opportunities for achieving greater efficiency than preceding techniques. In that sense, co-clustering avoided the double-optimization issue present in IB methods and worked towards optimizing a unique global criteria, defined as the minimum loss of mutual information between the original joint probability distribution $p(X,Y)$ and the joint probability distribution of the clustered variables $p(\hat{X},\hat{Y})$. So, the optimal co-clustering solution is the one that minimizes $I(X;Y)-I(\hat{X};\hat{Y})$. The co-clustering algorithm is guaranteed to converge to a local minimum after a fixed number of steps.

On each step, the co-clustering algorithm re-assigns a row and a column to a different cluster. However, instead of performing element-to-element comparisons, it compares the evaluated element against the cluster-prototypes, which can be thought of as "centroids" in the context of k-means. As a matter of fact, the ability of algorithms such as k-means or co-clustering that only require performing comparisons against a cluster representative, is a defining feature to achieve good scalability. The computational complexity of co-clustering is given by $O(n_z \cdot \tau \cdot (k+l))$, where n_z is the number of non-zeros in the joint distribution $p(X,Y)$; τ is the number of iterations; k and l the number of row and column clusters respectively.

Very recently, alternative versions of the co-clustering algorithm have been proposed in order to produce soft-clusters. Among the methods that address such problem we can name the Bayesian Co-Clustering [52] and the Latent Dirichlet Bayesian Co-Clustering method [68].

2.4 Discussion

Probabilistic topic models are a flexible and theoretically sound approach to learn topics in collections, however in relation to our area of concern which is (very) large scale collection analysis, the main limitation of the approach is the computational complexity which is heavily dependent on the number of topics in the model. Although there have been interesting proposals on how to improve the efficiency of the sampling [45] or performing distributed inference [41], achieving greater scalability for very large corpora seems to imply a trade-offs in the quality of the estimations by limiting the number of topics below the optimal values and reducing the sampling iterations before the convergence zone. In other cases, improving scalability may require the implementation of model-specific optimizations that result on a loss of generality of the models [69].

We conclude that there exists a need for an alternative approach to the abstract topic-modeling problem, designed to work well in the scenarios where the following conditions are satisfied:

a) It is considered good enough to know the top-k most probable words for a topic without a specific ordering or probability value estimation (this simplified presentation is in fact a usual way to describe the results of the topic modeling process).

b) In terms of the topic proportions of documents it is considered good enough to know which topics are covered in the document, without requiring the assignment of specific probability values to each topic.

c) The size of the collection is in the order of millions of documents of variable length and the number of topics is unknown and presumably very large[2].

d) There exist state-of-the-art search engine technology, such as the ability to do parallel processing in map-reduce style and to serve thousands of queries per second using distributed indexes.

[2] Consider Wikipedia, if each article is considered a topic, the number of topics would be in the order of hundreds of thousands

Chapter 3

Problem Statement

In this chapter we attempt a formal treatment of the problems of interest of this work. First we will define some of the concepts used throughout the discussion in a formal way, then, we will specify the goals and the tasks that represent our problems and discuss the relations among them. Finally, once the formal elements are developed, we will move to discuss on a higher level our hypotheses, assumptions and research goals.

3.1 Preliminary definitions

- Let $D = \{d_1, d_2 ... d_n\}$ be the document corpus of size N,
- Let $W = \{w_1, w_2 ... w_M\}$ the vocabulary of all terms in the corpus and Q be the set of all boolean queries q_i that may be defined over the vocabulary W that will match at least one document from D. Also, given that each $w_i \in W$ is a query itself, then $W \subset Q$.
- Let $L(d_i) = \{l_{i1}, l_{i2} ... l_{ik}\}$ be the set of topic labels assigned to document d_i by a human judge. We say that two documents d_1, d_2 are *semantically similar* they have at least one label in common, that is, if $\{L(d_1) \cap L(d_2)\} \neq \emptyset$.
- A *topic* is $T_i \subset D$ is defined as a *set of documents*. From the practical point of view, it is desirable but not essential, that such sets are comprised of semantically similar documents in accordance to human judgment.
- A *topical-query* denoted by $q_i \in Q^*$, $Q^* \subset Q$ is a query that retrieves semantically similar documents, therefore, according to previous definition is a query that retrieves documents containing the same labels.

3.2 Formal task definitions

The definition of a topic as a set of documents allows producing an alternative, discrete formulation of the topic modeling problem that we have denominated "Discrete topic modeling" (DTM). However, the DTM task is a very generic one, so, we also define the Query-Based topic modeling task, a particular instance of the DTM task which embodies our baseline approach to the problem, as it specifically requires the usage of search queries to solve the DTM problem. In this section we will present the formal definitions of the previously introduced DTM and QTM problems as well as the topic-identification task, a subproblem of QTM that initially served as a motivation to our research.

3.2.1 Discrete topic modeling task

We define the general *discrete topic modeling* task (DTM) as the problem of finding a clustering of D in K clusters: $U = \{u_1, ..., u_K\}$ with maximum semantic coherence with respect to human judgment. As naturally occurring, topics may overlap across documents and a single document may belong to several topics, so, the clusters in U that model the "real" topics are also allowed to overlap each other. For the purposes of this work, semantic coherence is defined as the harmonic mean of two quantities:

- The probability of randomly selecting two documents from the same topic taken from a randomly sampled cluster, denoted by P_{PM}.
- The probability that a randomly selected document is included in at least one cluster, denoted by R_{U}.

Which can be expressed as:

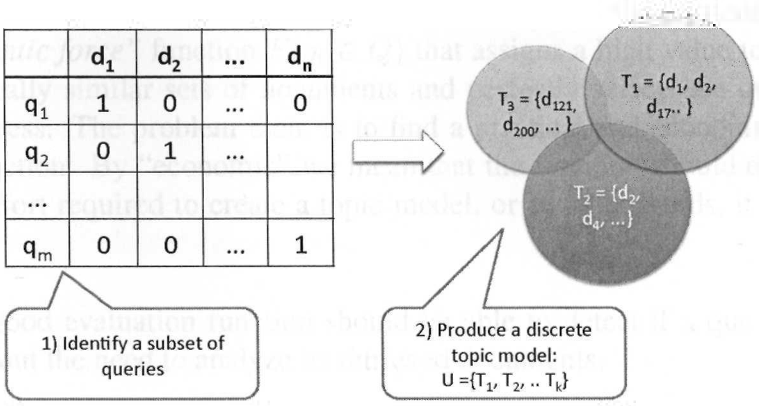
$$F_o(U) = \frac{2P_{PM}R_U}{P_{PM} + R_U} \tag{3.1}$$

The metric described in formula 3.1 is a form of the well-known F-score and as such, it accounts both for the completeness and for the coherence of the clustering under consideration. As simple as it may sound, the computation of the "coherence" probabilities is not trivial and deserves a detailed discussion that will be offered in chapter 7. However, the discrete topic modeling task along with the semantic coherence metric serve as the foundation to compare probabilistic topic modeling approaches with alternative soft-clustering and overlapping clustering algorithms such as the one presented in this work.

3.2.2 Query-based Topic Modeling task

The Query-Based Topic Modeling (QTM) problem is a particular instance of the Discrete Topic Modeling task and it can be defined as follows: Given a text document collection represented as a query-document co-occurrence matrix, where the rows of the matrix can be any subset of Q, find a:

1. A set of topical queries Q^* that maximizes the coherence of the clustering
2. A semantically coherent clustering of D in K potentially overlapping clusters,



	d_1	d_2	…	d_n
q_1	1	0	…	0
q_2	0	1	…	
…				
q_m	0	0	…	1

$T_3 = \{d_{121}, d_{200}, \ldots\}$
$T_1 = \{d_1, d_2, d_{17}\ldots\}$
$T_2 = \{d_2, d_4, \ldots\}$
1) Identify a subset of queries
2) Produce a discrete topic model: $U = \{T_1, T_2, \ldots T_k\}$

Figure 3.1: Schematic representation of the QTM task

Notice that in this case the rows of the matrix can be the vocabulary W (given that $W \subset Q$) or any other query set that contains information about the documents. For instance, in Puppin et. al [47] a set of popular search queries from a search engine has been selected, then the set was co-clustered along with documents to group together the documents that will likely be retrieved by similar queries.

From the collection modeling perspective, we acknowledge that both the vocabulary and the set of popular search queries are subsets of Q that define all the valid queries that could be constructed using the vocabulary terms. However, it is presumable that not every query set is equally useful for the purpose of modeling the collection topics. For instance, speaking of the extreme cases, if the query set is very small and does not retrieve all the documents of the collection, the missing documents will produce a penalization in R_U, by formula 3.1. On the other extreme, is a very large subset of Q is selected, it will quickly become unpractical to process.

So, the fundamental problem of QTM could be summarized as that of finding a set of topical queries Q^* and use it to build an semantically coherent clustering of D efficiently using the computational resources at hand. As we will show later, the resource constraints could play a role in the different QTM strategies used.

3.2.3 Topical-query identification task

The topical-query identification task is the first sub-problem of the QTM problem, and it could be defined as the problem of finding a set of topical queries $Q^* \subset Q$ that result in maximum semantic coherence of the model.

However, as exploring the combinatory space of all the possible query-sets is not computationally feasible, the practical solution of the problem involves the development of heuristics to locally evaluate and select highly informative queries.

So, in order to approach this problem, we need to assume the existence of a hypothetical, ideal *"semantic force"* function $F(q_i \in Q)$ that assigns a high value to the queries that retrieve semantically similar sets of documents and perfectly selects the queries to include in the QTM process. The problem then, is to find a practical and economic approximation to such ideal function. By "economic" we mean that the function should decrease the total computational effort required to create a topic model, or in other words, it should be worth computing.

Ideally, a good evaluation function should be able to detect if a query is likely to be informative without the need to analyze its retrieved documents.

3.3 Hypothesis and Research Questions

This work approaches a known general problem from an alternative perspective, so, broadly speaking, our hypotheses and research questions are developed around the feasibility of achieving similar results to state-of-the-art methods in scenarios where they are not suitable to be used, such as very large scale collections. Concrete hypotheses, presented below, are related to each of the building blocks of the proposed approach, such as the representation scheme, the underlying theory and the algorithmic solution.

3.3.1 Latent Semantic Structure Representation

Based on existing evidence we may state that some form of latent structure representation can be used to improve retrieval and other information access applications.

We have seen several approaches put to work based on different notions of what a topic is, such as LSI [5] where the notion of topic is implicit on the semantic distance of words and documents or as in Probabilistic Models, where topics are represented as probability distributions of terms [59].

Therefore, at this point we consider reasonable to hypothesize that our proposed representation, of topics as sets of documents associated to a topical-query would be at least as useful as the probabilistic representation in terms of retrieval performance.

3.3.2 Feasibility of the Information-Theoretic Approach

In 1972 Karen Spark Jones proved that the weighting of query terms based on its relative frequency can improve retrieval performance [58] and since then, the IR community have been successfully using and exploring such frequency-based weighting schemes, like tf-idf. It has been shown that tf-idf is closely related to the information-theoretic notion of information gain also known as Kullback-Leibler divergence [1]. Under this framework tf-idf can be interpreted as the decrease of uncertainty about the document relevance given a query. Although there are several other interpretations of tf-idf [50], the value of such measures is their usefulness to relate the term frequency to its specificity.

Regarding to the topic discovery problem, in works like Sista [53], tf-idf was used to select topic labels and topic support words. Their results included a user study, showing that the topic labels found by tf-idf weighting were in general acceptable to human readers. Moreover, as the computational cost of computing tf-idf is very low, it seems promising to apply in some form to large scale collections analysis.

Therefore, based on previous theoretic and experimental evidence, we hypothesize that there is a family of functions, similar in principle to tf-idf that can be used to identify good topical-queries. Those functions would assign higher values to queries that retrieve sets of semantically similar documents than to those queries that not. Provided with such functions, it would be possible to identify topics in the corpus by only looking at term frequencies and avoiding document-to-document comparisons.

Finally, assuming that the topic representation scheme contains enough information to provide useful results and that a "lightweight" function may be used to capture the topics, we need to presume the existence of a low complexity, parallelizable algorithm to create the topic model. Such an algorithm would be required to make the method available in large scale.

3.4 Objectives

At the end of this work we expect to accomplish the following objectives:

- *Metrics and theory.* To produce a fast and theoretically sound formula to measure semantic force of a query.
- *Algorithms.* To design a scalable algorithm that can be used to identify the topics in the collection and create a model based on the notion of topics as sets of documents. The complete modeling algorithm should include a discovery phase, in which many potential topics may exist and an optimization phase, in which the maximum semantic force per retrieved document is calculated and the final set of topical-queries is determined.
- *Benchmarks.* To define a robust benchmark for the topic modeling task, in order to measure the performance of a topic modeling algorithm against human judgment. That benchmark should include a pure "identification" task and may include a retrieval task.
- *Prototypes.* After having experimentally validated our approach, we should deliver prototype software that could be used to replicate the experiments and serve as a reference to more robust implementations.

Chapter 4

Topical-query identification

In this chapter we describe the information-theoretic basis of our approach to the problem of identifying a small set of queries that contain a significant amount of the semantic information of the collection. Recalling from previous chapter, for each given a query, we are interested in obtaining a fast estimation of the semantic similarity of its retrieved documents *without actually analyzing the documents.*

4.1 Estimating semantic force of queries using KL

When approaching the topic identification task as a search problem over a query-space, the fundamental challenge that arises is that of producing a low-cost approximation to the ideal function $F(q_i \in Q)$. While it would be almost unavoidable to execute the query, to perform document-to-document comparisons of the retrieved set would be prohibitive in terms of computational cost. So, as the basis of our approach we propose a sound way to perform simple query alterations and measure the amount of information in result sets without performing extensive comparisons.

Let q_i be a boolean query defined over a set of k words of the vocabulary and $W(q_i)$ the set of its terms, $W(q_i) = \{w_1, w_2..., w_k\}$ Now, we define two events for the experiment of selecting a random document of the corpus.

Let x be the event of retrieving a document with "any" of the terms of the query q_i. So, the probability of $P(x)$ is the probability of selecting a document retrieved by the query o_i, which is defined as a *disjunctive* (OR) query that matches any of the terms in $W(q_i)$, such that $o_i = \{w_1 \vee w_2... \vee w_k\}$.

Let y be de event of observing "all" the terms of the set $W(q_i)$. So, its probability $P(y)$ would be computed as the probability of selecting a document retrieved by a conjunctive query with all the terms of $W(q_i)$, like $a_i = \{w_1 \wedge w_2... \wedge w_k\}$.

As x and y are not independent events, we may notice that there exist the conditional event of observing *all* the query terms after having observed *any* of them with probability distribution $P(Y|x)$. So, we propose to approximate the semantic force $F(q_i)$ computing the

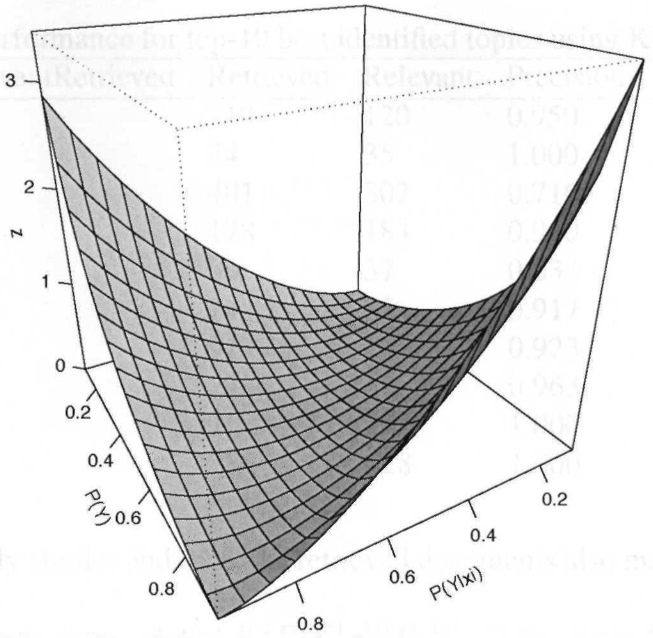


Figure 4.1: KL-divergence as semantic force

Kullback-Leibler divergence [18] or *information gain* over this two events. KL-divergence is formally defined as:

$$K(P(Y|x)||P(Y)) = \sum_{y_j \in Y} p(y_j|x) log \frac{p(y_j|x)}{p(y_j)} \tag{4.1}$$

Where $P(Y)$ is the probability distribution defined over the discrete random variable $Y = \{y, \neg y\}$. And $p(y_j)$ is the individual probability value for the event y_j as defined by the distribution $P(Y)$. In analogous way, $P(Y|x)$ is a discrete probability distribution that assigns probability values $p(y_j|x)$ to the conditional events $\{y \wedge x, \neg y \wedge x\}$.

The KL-divergence can be directly interpreted as how much more certain we are about the fact that our randomly selected document will contain all the words of the query (y), given that it has any of them (x). The equation 4.1 measures the divergence about probability distributions, so we must take into consideration the complements of our events of interest ($\neg y$). In addition, we introduce a minimum frequency threshold parameter z that causes the function to evaluate to 0 whenever the number of documents retrieved by query y, expressed by $n(y)$ is less than z. So by convenience we may express the proposed evaluation function as:

$$F(q_i) = \begin{cases} n(y) < z, & 0 \\ n(y) \geq z, & p(y|x) log \frac{p(y|x)}{p(y)} + (1 - p(y|x)) log \frac{(1-p(y|x))}{(1-p(y))} \end{cases} \tag{4.2}$$

The proposed function leverages the fact that semantically similar documents are more likely to have similar terms that those unrelated. So, if the query terms retrieve similar sets of documents whether executed as a conjunction or as a disjunction, we may infer that the query

Table 4.1: Performance for top-10 best identified topics using KL-divergence

Topic	RelevantRetrieved	Retrieved	Relevant	Precision	Recall	Fscore
soybean	113	119	120	0.950	0.942	0.946
sorghum	24	24	35	1.000	0.686	0.814
wheat	287	401	307	0.716	0.935	0.811
sugar	119	128	184	0.930	0.647	0.763
silver	25	30	37	0.833	0.676	0.746
rapeseed	22	24	35	0.917	0.629	0.746
coffee	84	91	145	0.923	0.579	0.712
rubber	26	27	51	0.963	0.510	0.667
rye	1	1	2	1.000	0.500	0.667
grain	286	286	628	1.000	0.455	0.626

terms are semantically similar and thus, the retrieved documents also may be.

In figure 4.1 we present a plot of $K(P(Y|x)||P(Y))$ in the range $[0, 1]$. From the graph behavior we can observe that it will favor queries whose terms are infrequent ($p(y) \approx 0$) and tend to co-occur with high probability ($p(Y|x) \approx 1$). As earlier noted, in practice this behavior require us to set a parameter $z \in \mathbb{N}$ to assign a value of 0 to the topics below a minimum size of interest. If $z = 1$, it will tend to favor document-specific topics.

4.2 Experimental validation

In this chapter we present two different experiments using the Reuters-21578 corpus to show the feasibility of the method and to provide a descriptive comparison of the properties of the presented evaluation functions. For this set of experiments, the Reuters-21578 corpus was splitted into 109,120 sentences from which 2,735,795 candidate queries were generated. We set $z = 2$ to accept all potential discovered topics with more than two documents. In order to show how the proposed function compares with alternative approaches we compare the proposed KL-Divergence based semantic-force with some alternative functions, defined as follows:

- **Query mutual information (MI).** This function is similar in nature to the described in eq. 4.2 as it uses the mentioned events $P(y)$ to denote the probability of occurrence of all the terms of the query and and $P(x)$ to denote the probability of any of them. The mutual information is a measure of the dependence of both quantities and is also interpreted as the expected information gain.

  $$QMI(X,Y) = \sum_{x_i \in X} p(x_i) KL(P(Y|x_i)||P(Y)) \tag{4.3}$$

  As it can be observed, the mutual information function holds a strong relation to the presented KL divergence function but it incorporates more clearly the frequency of the terms of the query by using $P(X)$.

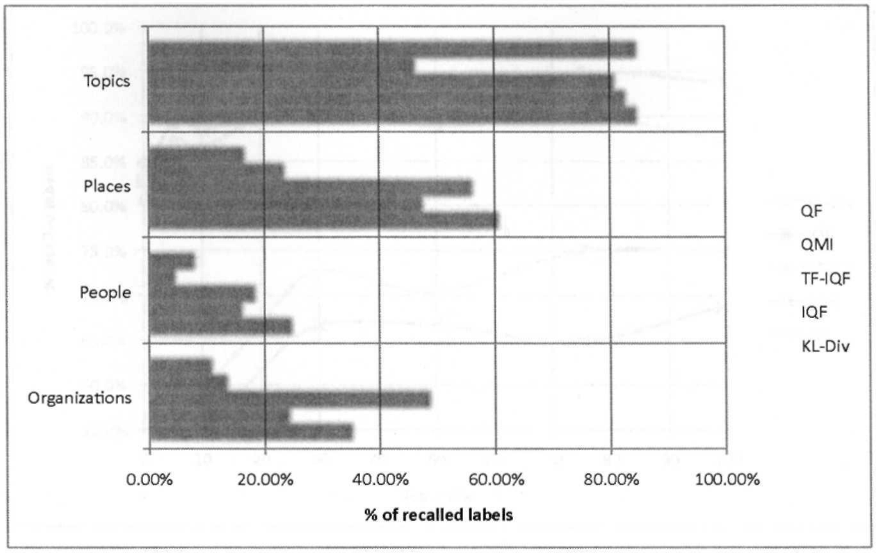


Figure 4.2: Topic label recall by concept class

- **Inverse Query Frequency (IQF).** For a given query q_i, the inverse frequency of a query is defined as the logarithm of the inverse of the number of documents that matches q_i.

$$IQF(q_i) = log\left(\frac{N}{n_d(q_i)}\right)$$

- **TF-IQF.** Analogous to TF-IDF, this function is defined here as the number of sentences that generated the query q_i within a document d_j, multiplied by the IQF as defined above. Formally,

$$\textit{TF-IQF}(q_i, d_j) = n_s(q_i, d_j) * IQF(q_i)$$

  Among the described metrics, a unique feature of the TF-IQF function is that it takes into account the document specific frequency of a given candidate query.

4.2.1 Topic label recall

In our first experiment we want to know to which extent the meaningful words in the corpus are captured by a topical-queries result set, where each query is the maximum evaluated candidate for every document. For each document one query was selected and the vocabulary of those query set was extracted. The set of meaningful words is given by the Reuters-21578 labels for four different concept classes, namely: topics, places, people and organizations. The label recall for a given concept-class, reported on figure 4.2 represents the percentage of “recallable” labels that were found on the extracted topical query set. The corpus contains 135 topic labels, from which we consider “recallable” the subset of 52 topics that exist in the vocabulary. In addition the corpus contains 175 place labels, 267 people names and 56 organizations, from which 37 exist in the vocabulary. We consider these words as a human provided list of the most meaningful description of the corpus.

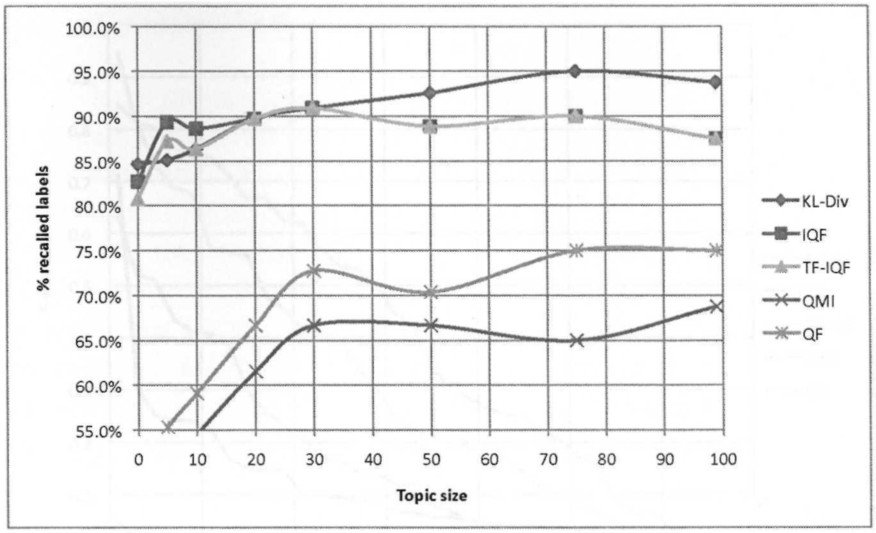


Figure 4.3: Topic label recall by topic size

In figure 4.2 we present the initial performance result of the label recall task. Topic labels were better recalled than the other concept classes by the majority of evaluation functions, achieving above 85% recall for topics comprising more than 5 documents. We hypothesize that the recall performance decreases as a function of the complexity of terms (i.e. people and organization names are more complex than topics and places). In this benchmark the KL-Divergence function slightly outperforms the frequency based ones for all the concept classes except for organizations names.

Also, in figure 4.3 we show how the different functions perform on identifying topic labels depending on the size of the topic, as given by number of documents that are labeled with them. One of the things than can be appreciated is how the functions are clearly divided in two different groups, namely the ones that favor term specificity such as KL-Div, IQF, TF-IQF which are good at identifying "small" topics, and on the other hands the functions that favor frequency, which are only marginally good at identifying broader topics, such as QF and QMI.

4.2.2 Topic identification performance

Table 4.2: Average topic identification performance

	Precision	Recall	F-Score
KL-Div	0.767	0.375	0.441
TF-IQF	0.788	0.239	0.312
QMI	0.929	0.014	0.028
QF	0.728	0.181	0.245
IQF	0.870	0.075	0.124

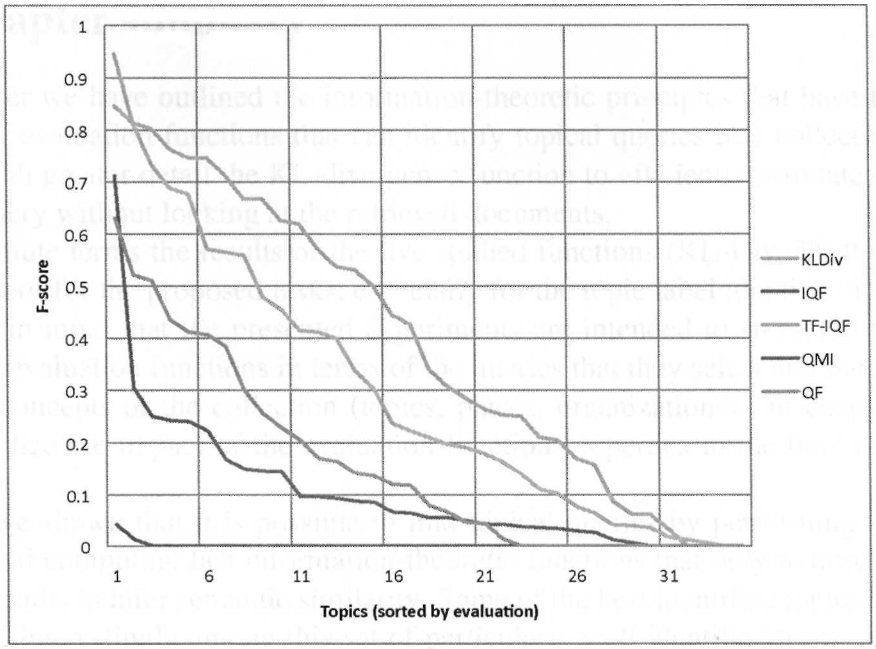


Figure 4.4: Topic identification performance

On the following experiments, we are concerned with doing an initial measurement of the similarity of the identified document with those produced by human labelers. For each topic label in the Reuters corpus we "executed" all the identified queries that contained the label word and created a "retrieved" result set. So, the retrieved result set was simply the union of the result set of all the queries that contained the label.

For each of the topic labels, its retrieved results were compared to the original labeled set, then, recall, precision and f-score were computed in the traditional way. The summarized results of the task are presented in figure 4.4 and table 4.2. For some topic labels no relevant documents were retrieved, so the results are splitted by total average and by the fraction of the topics for which at least 1 relevant document was retrieved, which are reported in the category "recalled". The curves from figure 4.4 show the recalled topics in decreasing order of f-score for each function. Thus, if a function appears "covering", another it means that it outperforms it. On the other hand, the more to the right a function intersects with the X-axis, the grater the number of recalled topics. So, in figure 4.4 we may observe that the KL-divergence semantic force function clearly outperforms the alternatives in terms of f-score being the function that clearly provides best recall, and overall balance between recall and precision. This may be due to the fact that the KL-div function is the only one that considers the "disjunctive" or independent frequency of the query terms. Table 4.2 presents the average scores for all the recalled topics and from its observation we can confirm that KL-Div is the only function delivering precision and recall above 75% and 35% respectively.

4.3 Chapter summary

In this chapter we have outlined the information-theoretic principles that have been used to design query evaluation functions that can identify topical queries in a collection. Also we presented with greater detail the KL-divergence function to efficiently estimate the semantic force of a query without looking at the retrieved documents.

In absolute terms the results of the five studied functions (KL-Div, TF-IQF, QMI, QF, IQF) were poor for the proposed tasks, especially for the topic label identification. However, please keep in mind that the presented experiments are intended to show the properties of the different evaluation functions in terms of the queries that they select and the relation with the labeled concepts of the collection (topics, places, organizations). In chapter 7 we will clearly visualize the impact of the evaluation function properties in the final outputs of the topic models.

We have shown that it is possible to find topical-queries by performing simple query alterations and computing fast information-theoretic functions that only require counting the number of results to infer semantic similarity. Some of the best identified topics are presented on table 4.2, interestingly among this set of particularly well identified topics we can count topics with more than 600 relevant documents such as “grain” as well as others containing only 2 documents, such as “rye”. Our intent is to present this results as preliminary evidence of the potential of the technique to match the human topic assignments. A more robust benchmark will be presented on chapters 6 and 7.

Chapter 5

The QTM framework

5.1 Introduction

In this chapter we propose the Query-Based Topic Modeling (QTM) framework, an information-theoretic approach to build semantic models of large text collections. In contrast to Probabilistic Topic Modeling (PTM) methods like LDA, QTM models topics as overlapping sets of semantically similar documents associated to "topical-queries". A simpler notion of topic drives computational advantages, for instance, in QTM a mixture of topic probabilities is estimated for each document, however, the estimation of term-topic probabilities is not performed at all. In addition, the QTM framework follows the map/reduce style of design, relying as much as possible in stream-oriented operations, which significantly decreases the amount of memory required to run the algorithm and allows operations to be performed in parallel.

The chapter is structured as follows. First, we will describe the main processes and components of the framework and the technical decisions justifying the approach. Then, we will discuss some theoretical aspects regarding specific component implementation that were explored in the course of the research, concretely regarding the evaluation functions. Finally, we will propose a model description notation that will be useful to walk trough the experimental aspects of the work.

5.2 The QTM process

The query-based topic modeling framework (QTM) illustrated in figure 5.1 summarizes our approach to collection semantic modeling. Given a raw document collection, the QTM process can generate two kinds of models. The simplest is a traditional clustering, with non-overlapping partitions where every document is guaranteed to belong to exactly one partition. The second type of model is an overlapping clustering, where a document may belong to zero or more partitions. Depending on the type of model that is to be generated the process could take 3 to 4 phases, described as follows.

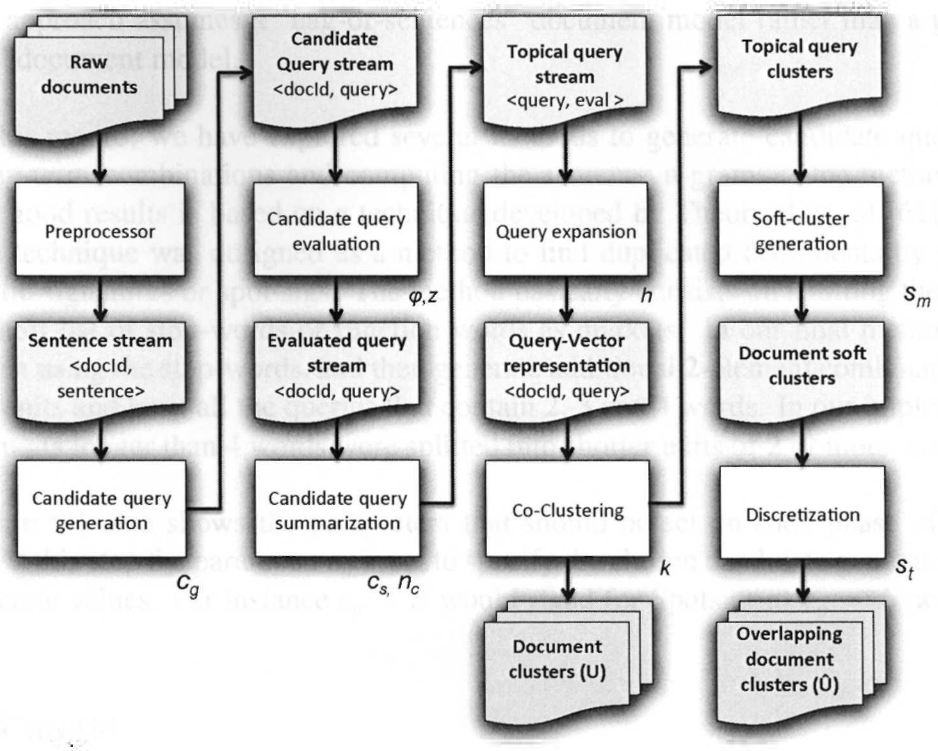

Raw documents
Candidate Query stream <docId, query>
Topical query stream <query, eval >
Topical query clusters
Preprocessor
Candidate query evaluation
Query expansion
Soft-cluster generation
φ,z
h
s_m
Sentence stream <docId, sentence>
Evaluated query stream <docId, query>
Query-Vector representation <docId, query>
Document soft clusters
Candidate query generation
Candidate query summarization
Co-Clustering
Discretization
C_g
C_s, n_c
k
s_t
Document clusters (U)
Overlapping document clusters (Û)

Figure 5.1: Map/Reduce version of the topic identification algorithm

5.2.1 Pre-processing

The pre-processing phase takes the raw documents of the corpus. The first process that is performed is the document indexation into an inverted index. For our implementation we have used Apache Lucene. In order to evaluate the candidate queries, the index is frequently accessed, so, it is required to provide fast-index serving of boolean queries to the evaluation process. Stop-words were not removed during the indexing process, as they are important to the evaluation.

The second operation in pre-processing involves the tokenization of the documents to create a stream of sentences. Each document is divided into sentences and a new line is produced for each. In order to perform the sentence-splitting we are using the Lingua::EN::Sentence Perl module. The sentence extraction component considers punctuation to determine sentence boundaries. In addition, when dealing with HTML markup, in absence of punctuation we consider some tag markers as sentence separators, such as *h1*, *p* and *li*.

5.2.2 Candidate query generation

The candidate query generator takes as input the sentence stream and for every sentence it produces candidate queries. Thus, implicitly assuming that the topics discussed in a document will affect the term co-occurrences at the sentence level. Also, it implies that the generated queries will capture some of the influence of lexical proximity. As a consequence, the proposed approach assumes a "bag-of-sentences" document model rather than a pure "bag-of-words" document model.

In this regard, we have explored several methods to generate candidate queries, such as creating term combinations and computing the sentence n-grams. One method that has produced good results is based on a technique developed by Theobald et. al [61]. Interestingly, the technique was designed as a method to find duplicated documents by generating its semantic signatures or spot-sigs. The method basically consists on splitting the sentences using a short list of stop-words or function words as markers. In our final method, we first split the list using the stop-words, and then generate additional 2-element combinations of the resulting units and keep all the queries that contain 2, 3 and 4 words. In our implementation, resulting units longer than 4 words were splitted into shorter units of 2 or more words.

Figure 5.1 also shows the parameters that should be set on each phase of the QTM process. At this step the parameter c_g is set to specify the chosen candidate generation method using discrete values. For instance $c_g = S$ would stand for Spotsigs as $c_g = N$ would for n-grams.

5.2.3 Candidate query evaluation

In general, the evaluator component could be any program that reads the candidate query stream, line by line, evaluates each query and writes to the evaluated query stream. The evaluation results are used to rank and select a subset of the candidate queries in latter phases of the process; thus, the evaluation function is an heuristic that should help deciding which queries to keep and which queries to discard. Good evaluation functions should be "cheap" to compute.

In the course of this research, we have explored functions that leverage the information-theoretical properties of the query terms to infer properties about the retrieved documents *without* actually retrieving them, by only considering the hit counts of some alterations of the original query.

Candidate queries are evaluated using one of the semantic force functions proposed in chapter 4. Each evaluation could require sending queries to an index server. Depending on the size of the collection, the index may be located on a single machine, distributed to the nodes in the map/reduce cluster or accessed remotely from an index-serving cluster. Each deployment has different implications that need to be further researched.

As shown in figure 5.1, the chosen evaluation function is represented by the discrete parameter φ that can take values such as $\varphi = KL$ to indicate that a QTM instance will use the Kullback-Lleibler divergence as evaluation function, or $\varphi = MI$ to specify that the queries will be evaluated according to its contribution to the mutual-information about the documents. In addition, the z parameter, a positive integer, is used to determine the number of retrieved results threshold below that all the queries evaluate to 0.

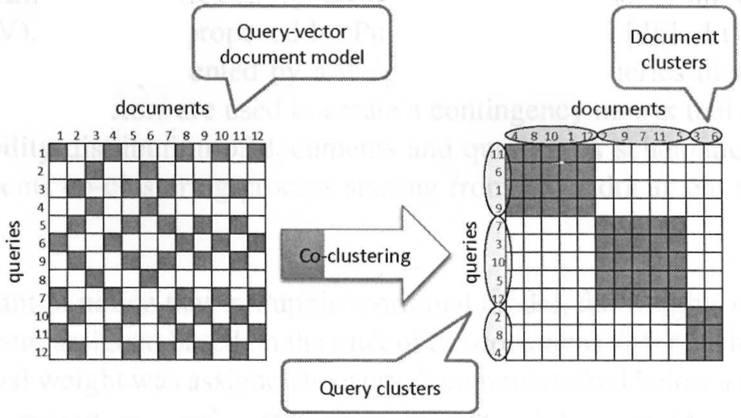


Figure 5.2: Schematic representation of the co-clustering process

5.2.4 Candidate query summarization

The summarization phase consist in determining the final query set of topical queries to be included into the model. Among the many candidate query selection strategies are possible we have analyzed two:

By-document selection Consists on selecting the maximum s_c evaluated queries of each document, where $s_c > 0$. Then duplicates are removed. This method guarantees that every document will be retrieved by at least one query in the final topical query set.

Global selection Consists on selecting the maximum s_c evaluated queries of any document. This method doesn't guarantee that every document is retrieved by a query in the final topical query set, however it can include queries with higher evaluations than the previously described method.

The candidate query summarization (or selection) method is formally specified using the discrete parameter c_s. For the purposes of this work, we use $c_s = D$ to denote by-document selection and $c_s = G$ for global selection, respectively.

5.2.5 Query expansion and co-clustering

At the beginning of this phase, we have a set of topical-queries that ideally should capture most of the semantic information of the collection. As this set is too large to be useful for human analysis, queries and documents should be clustered to obtain a smaller number of broader topics. Among several algorithms co-clustering, proposed by Dillon, et. al. [22] turned out to be a natural match to the QTM problem given that: a) we need to cluster bi-dimensional data, b) it is guided by information-theoretic principles and b) it has been shown to be scalable.

However, before running the co-clustering algorithm the collection have to be represented as a contingency matrix where each row is a query q_i and each column a document

d_j. So, we transformed our collection representation using a variant of the Query-Vector document model (QV), originally proposed by Puppin and Silvestri [46]. In the QV document model each document is represented by a weighted vector of queries that retrieve it. Then those query-document vectors are used to create a contingency matrix that can be interpreted as a joint probability distribution of documents and queries. A schematic representation of the query-documents co-clustering process starting from a QV document model is shown in figure 5.2 [1].

It is important to notice that in Puppin's original model, the weights of the vector terms representing d_j were assigned based on the rank of the document d_j for each query q_i, whereas in our case an equal weight was assigned to every document ranked below a specified cut value h. We decided to assign an equal weight to each retrieved document in order to better control the effects of the ranking function on the final model. Documents are ranked using the Apache Lucene's default scoring function, which essentially uses the tf-idf term weighting over a vector-space document model. We will show experimentally that our strategy preserves enough information to build the model, however, the effects of the ranking function and its potential to provide additional value by incorporating information such as linkage or term prominence is a problem that remains to be explored.

Finally, in our implementation, the co-clustering algorithm requires a single parameter k, to specify the number of row (queries) and column (document) clusters to be produced. If the application under consideration can work using non-overlapping partitions, the results of this phase can be considered the final output of the QTM process.

5.2.6 Soft-clustering generation and discretization

The final steps of QTM process deal with the problem of generating an overlapping partition of the document collection taking as input the non-overlapping query and document partitions generated by the co-clustering algorithm. The resulting overlapping partition of the corpus, identified by $\hat{U}$ will be considered the final representation of the topic structure.

In general, the strategy employed to generate the overlapping clusters is to first to generate a "soft" clustering based on an estimation of document-topic probabilities and then discretize the document-topic probabilities to create "hard" document-topic associations using a threshold value s_t.

The probability of the document d_j belonging to a given topic $\hat{U}_m$ is estimated by the number of queries matching d_j that belong to query cluster Q_m divided by the total number of queries that match the d_j.

$$p(d_j \in \hat{U}_m) = \frac{\sum_i (q_{ij} \in Q_m)}{\sum_i q_{ij}} \tag{5.1}$$

Where $\{q_{1j}, q_{2j}, ...q_{nj}\}$ is the list of queries that match document d_j.

[1] Figure 5.2 adapted from Puppin, et. al 2007

5.3 Model description notation

The QTM process shown above can be think of as "template" to create specific collection models, depending on the concrete method and parameter choices.

So, in order to describe individual instances we will need to introduce notation to formally describe its properties.

As shown above, the end result of the QTM process is a partition or clustering of the documents of the collection. Generated partitions could be complete, non-overlapping just as in a traditional clustering algorithm or can be overlapping and without completeness guarantee.

5.3.1 Model for non-overlapping partitions

Let Q_{TM} be an algorithm instance that produces a set of non-overlapping partitions denoted by U. A model instance will be specified by the expression:

$$U \sim Q_{TM}(c_g, \varphi, z, c_s, n_c, k, h)$$

Which can be read as the partition U produced by algorithm Q where the following parameters have been specified:

- c_g : Candidate query generation method.
- φ: Candidate query evaluation function.
- z: Minimum query results allowed.
- c_s: Candidate query selection method.
- n_c: Number of selected queries.
- k: Number of topics to construct.
- h: Number of hits per query to include in the query-vector representation.

5.3.2 Model for overlapping partitions

In analogous way, we will use the literal $\hat{Q}_{TM}$ to denote a QTM algorithm that produces a set of overlapping partitions $\hat{U}$, with a model formaly defined by the expression:

$$\hat{U} \sim \hat{Q}_{TM}(c_g, \varphi, z, c_s, n_c, k, h, s_m, s_t)$$

The parameters required to build an overlapping model c_g, c_s , n_c, k and φ are the same as in Q_{TM}, plus two additional parameters:

- s_m: Soft-clustering generation method.
- s_t: Soft-clustering inclusion threshold

5.4 Computational complexity and scalability

The computational complexity of a single node running QTM depends on the phase of the process. For the candidate generation and summarization phase, the time complexity is in $O(N)$, the evaluation methods considering that they require at least one lookup, would be in the order of $Nlog(W)$ where W is the size of the vocabulary. In analogous way, the co-clustering phase would take $O(2 \cdot I \cdot n_z \cdot K)$ steps, where I is the number of iterations, n_z is the number of non-zeros in the QV matrix, which is bounded by the hits and the number of selected candidates $O(h \cdot s_c)$ and K is the number of clusters, assuming that the number of query and document clusters is the same.

When analyzed on a single node, QTM's computational complexity can seem very similar to that of LDA, which, using Gibbs sampling has a complexity of $O(INK)$ where N stands for the total number of word occurrences in the corpus. However, the problems of LDA are more evident when the data grows and a parallel version of the algorithm is required to run. As reported by several authors like Liu [38] and Chen, et. al [14], parallelizing LDA under the Map-Reduce paradigm is hard as the algorithm presents memory and communication bottlenecks. For instance, Chen, et. al. report that *"parallel LDA can achieve approximately linear speedup on up to 8 machines. After that, adding more machines yields diminishing returns. When we use 32 machines, the communication time takes up nearly half of the total running time"*.

One of the main motivations to propose QTM was to obtain enhanced scalability with respect to existing probabilistic methods like LDA. So, despite the apparent similarities on single-node computational complexity, the key to QTM scalability is the simpler topic representation, that allows the usage of the co-clustering algorithm at the most complex phase of the QTM process. In contrast to LDA, co-clustering has been shown to be end-to-end map reduce scalable by Papadimitriou, et. al. [42].

Chapter 6

Topic Model Validation Metrics

6.1 Introduction

Evaluating the performance of techniques such as LDA, LSI or any other unsupervised modeling algorithm including QTM is a non-trivial problem, especially when it comes to drawing strong conclusions about the quality of the models. In this chapter we present a novel set of metrics that can be used to perform external validation of the semantic coherence of topic modeling and soft clustering methods using multi-labeled corpora such as Reuters-21578 or 20-Newsgroups.

The proposed topic modeling validation approach is based on transforming the topic model under analysis into a "hard" overlapping partition by discretizing the "soft" document-topic associations. The proposed metrics are based on alternative interpretations of widely accepted concepts such as "precision" and "recall". In addition, the presented validation approach has among its advantages:

1. An intuitive and explicit probabilistic interpretation.
2. Applicability to validate overlapping and incomplete partitions.

The rest of this chapter is organized as follows. In section 6.2 we establish the notation and the main objects involved in our analysis. In section 6.3 we briefly discuss the nature of the existing approaches to cluster validation and perform a dissection of the Fowlkes-Mallows Index in order to clearly expose its underlying probabilistic principles. Then, based on the previous analysis, we propose a generalized version of it that can be applied to overlapping and incomplete clustering solutions. However, by working on this generalization we became more aware of its limitations, so, in section 6.4 we propose some alternative metrics based on similar principles but of simpler interpretation. Finally, we show how the proposed metrics can be approximated using Monte-Carlo methods.

6.2 Notation

Once the soft-clustering solution, of a multi-labeled corpus, is discretized to obtain the corresponding hard-clustering, the problem to be faced with consists to correctly evaluate the

Table 6.1: Cluster-class contingency matrix

		Classes				
		c_1	c_2	...	c_s	Σ
	u_1	n_{11}	$n_{1,2}$	...	n_{1s}	n_{1*}
Cluster	u_2	n_{21}	$n_{2,2}$	...	n_{2s}	n_{2*}
	...	...	...	...	...	...
	u_k	n_{k1}	n_{k2}	...	n_{ks}	n_{k*}
	Σ	n_{*1}	n_{*2}	...	n_{*s}	n_{**}

quality of the resulting overlapping partitions. Let us first introduce the terminology and the notation which will be used in the rest of the paper. Every "hard-clustering" problem applied to a multi labeled document corpus involves the following elements:

- a dataset $D = \{d_0, ..., d_n\}$ consisting of n documents;
- a partition of D in K clusters: $U = \{u_1, ..., u_K\}$;
- a partition of D in S classes: $C = \{c_1, ..., c_S\}$.

Most of the existing validation metrics [70] can be expressed in terms of a $|U| \times |C|$ contingency (Table 6.1) where the content of each cell n_{ij} represents the number of documents belonging to cluster u_i and class c_j.

In the special case where clusters do not overlap and the document corpus is singly labeled, the following properties hold:

1. $\bigcup_0^K u_i = D$;
2. $u_i \cap u_j = \emptyset \; \forall i, j = 1, ..., K$ with $i \neq j$: there is no "overlap" between the elements of the cluster partition;
3. $c_i \cap c_j = \emptyset \; \forall i, j = 1, ..., S$ with $i \neq j$: there is no "overlap" between the elements of the class partition.

In this work we consider the case where the aforementioned properties cannot be assumed to hold. Indeed, in a realistic setting

- the thresholding procedure used to move from soft to hard clustering, may result in some documents being unassigned;
- a document can be assigned to more than one cluster;
- the document corpus is multi-label and thus in principle every document can be assigned to one or more classes.

6.3 Topic Model Validation Approaches

When it comes to validating topic models, two approaches are usually followed: The first consists of measuring the performance of a machine learning task such as classification or retrieval, which introduces new variables and interactions into the problem and as a consequence it makes hard to make inferences about the underlying quality of the models, unless of course the problem of interest is the performance of the retrieval or classification task as such. The second approach that has been used consists of user studies, which is valid in principle, but in practice suffers from lack of repeatability and it is difficult to compare anything against results reported elsewhere. Also, as employing human time is costly and time consuming, many of the reported results tend to be small, which hurts the significance of the results.

The problem of validating topic models, without relying on the performance of a task, was recently addressed by Chang et.al [13]. Their work, the first in that direction, shows by means of user studies that some problem exists when using supervised classification predictive metrics. Considering topic models as a particular form of soft-clustering models brings the attention to previous contributions concerning cluster validation. It is worthwhile to notice that most of the works on external clustering validation deal with the case of non-overlapping partitions of uni-labeled corpora. A comprehensive review of the traditional metrics used to validate non overlapping partitions can be found in [70] and [21].

6.3.1 Dissecting the Fowlkes-Mallows index

Among the existing cluster validation metrics, one of particular interest is the Fowlkes-Mallows index [24], [67] (hereafer referred to as "FM"). Using the contingency matrix notation from Table 6.1, the FM index is defined as follows:

$$FM = \frac{\sum_{ij}\binom{n_{ij}}{2}}{\sqrt{\sum_i\binom{n_{i*}}{2}\sum_j\binom{n_{*j}}{2}}} \tag{6.1}$$

To analyze the FM index the following events, associated with the random sampling of two documents d_1 and d_2 from a given documents corpus, have to be defined:

- S_{c_*}: d_1 and d_2 belong to the same class;
- S_{u_*}: d_1 and d_2 belong to the same cluster;
- $S_{u_*c_*}$: d_1 and d_2 belong to the same cluster and class.

To denote the event of d_1 and d_2 belonging to class c_j we write $S_{c_{*j}}$, whose probability is given by:

$$P(S_{c_j}) = \frac{\binom{n_{*j}}{2}}{\binom{n_{**}}{2}} = h(n_{**}, n_{*j}, 2, 2) \tag{6.2}$$

where $h(n_{**}, n_{*j}, 2, 2)$ represents the probability value, according to the hypergeometric distribution, to obtain 2 successes in a sampling without replacement of size 2, from a population of size n_{**} that contains n_{*j} successes. In a similar manner, we write S_{u_i} to denote that d_1 and d_2 belong to cluster c_i, where the corresponding probability value is given by:

$$P(S_{u_i}) = \frac{\binom{n_{i*}}{2}}{\binom{n_{**}}{2}} = h(n_{**}, n_{i*}, 2, 2) \tag{6.3}$$

The probability of two documents belonging to the same class can be computed from expression (6.2) to be:

$$P(S_{c_*}) = \sum_j P(S_{c_j}) = \frac{1}{\binom{n_{**}}{2}} \sum_j \binom{n_{*j}}{2} \tag{6.4}$$

while the probability of two documents belonging to the same cluster can be computed from expression (6.3) to be:

$$P(S_{u_*}) = \sum_i P(S_{u_i}) = \frac{1}{\binom{n_{**}}{2}} \sum_i \binom{n_{i*}}{2} \tag{6.5}$$

Finally, the probability of two randomly sampled documents, without replacement, to belong to the same class and cluster is:

$$P(S_{u_*c_*}) = \sum_{ij} P(S_{u_ic_j}) = \frac{1}{\binom{n_{**}}{2}} \sum_{ij} \binom{n_{ij}}{2} \tag{6.6}$$

Then, the conditional probability that two randomly sampled documents, without replacement, belong to the same class given they belong to the same cluster is:

$$P(S_{c_*}|S_{u_*}) = \frac{P(S_{u_*c_*})}{P(S_{u_*})} = \frac{\sum_{ij}\binom{n_{ij}}{2}}{\sum_i \binom{n_{i*}}{2}} \tag{6.7}$$

while the conditional probability that they belong to the same cluster given that they belong to the same class is:

$$P(S_{u_*}|S_{c_*}) = \frac{P(S_{u_*c_*})}{P(S_{c_*})} = \frac{\sum_{ij}\binom{n_{ij}}{2}}{\sum_j \binom{n_{*j}}{2}} \tag{6.8}$$

It is worthwhile to notice that the FM index (6.1) can be obtained by computing the geometric mean of the conditional probability that the pair of sampled documents belong to the same class given they belong to the same cluster ($P(S_{c_*}|S_{u_*})$) and the conditional probability that the pair of sampled documents belong to the same cluster given they belong to the same class ($P(S_{u_*}|S_{c_*})$). Therefore, expressions (6.7) and (6.8) allow us to write the following:

$$FM = \sqrt{P(S_{c_*}|S_{u_*})P(S_{u_*}|S_{c_*})}. \tag{6.9}$$

The previous formulations can also be expressed in terms of the hypergeometric distribution. This is helpful for computational purposes and to better understand the properties of the considered metric. For instance, by expressions (6.3) and (6.5), the probability to sample two documents from the same cluster, could be rewritten as follows $P(S_{u_*}) = \sum_i h(n_{**}, n_{i*}, 2, 2)$ while the probability to sample two documents from the same class becomes $P(S_{c_*}) =$

$\sum_j h(n_{**}, n_{*j}, 2, 2)$. In a similar fashion, we have $P(S_{u_*c_*}) = \sum_{ij} h(n_{**}, n_{ij}, 2, 2)$. Thus, the conditional probabilities expressed above can be rewritten as:

$$P(S_{c_*}|S_{u_*}) = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sum_i h(n_{**}, n_{i*}, 2, 2)} \tag{6.10}$$

and:

$$P(S_{u_*}|S_{c_*}) = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sum_j h(n_{**}, n_{*j}, 2, 2)} \tag{6.11}$$

Finally, by geometric averaging (6.10) and (6.11) the new expression for (6.1) is:

$$FM = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sqrt{\sum_i h(n_{**}, n_{i*}, 2, 2) \sum_j h(n_{**}, n_{*j}, 2, 2)}} \tag{6.12}$$

It is worthwhile to mention that equation (6.12) makes it easy to account the effects of overlapping clusters when computing the FM index.

Overlapping partitions

When validating using a multiply labeled corpus, such as Reuters-21578, the set of ground-truth classes result in overlapping partitions. In such a case the FM index cannot be computed by using equation (6.1), because the assumption of sampling without replacement does not hold. The main difficulty with overlap, when computing the FM index, is due to the use of the contingency matrix notation, that hides the probability being computed which easy results to make the wrong assumption that $n_{i*} = |u_i|$, $n_{*j} = |c_j|$ and $n_{**} = |D|$. The implications of such a wrong assumption are shown through the following example.

Example 1

Consider a non-overlapping partition consisting of 2 clusters and 2 classes.
Let $u_1 = \{d_1, d_2, d_3, d_4, d_5\}$ with $\{d_1, d_2, d_3\} \in c_1$ and $\{d_4, d_5\} \in c_2$, $u_2 = \{d_6, d_7, d_8, d_9, d_{10}\}$ with $\{d_6\} \in c_1$ and $\{d_7, d_8, d_9, d_{10}\} \in c_2$. The situation can be conveniently summarized through the following contingency matrix:

	c_1	c_2	$\sum$
u_1	3	2	$n_{1*} = 5$
u_2	1	4	$n_{2*} = 5$
	$n_{*1} = 4$	$n_{*2} = 6$	$n_{**} = 10$

Accordingly to (6.2) and (6.4) we can compute $P(S_{c_*})$ as follows:
$P(S_{c_*}) = \sum_j P(S_{c_j}) = \sum_j h(n_{**}, n_{*j}, 2, 2) = \frac{\binom{4}{2}}{\binom{10}{2}} + \frac{\binom{6}{2}}{\binom{10}{2}} = \frac{21}{45}$ and obtain a correct probability value.

The following class overlapping scenario, due to multi-labeled documents, is considered. Let c_3 be such that $\{d_1, d_4, d_8, d_9, d_{10}\} \in c_3$. The corresponding contingency matrix is:

	c_1	c_2	c_3	$\sum$
u_1	3	2	2	$n_{1*} = 7$
u_2	1	4	3	$n_{2*} = 8$
	$n_{*1} = 4$	$n_{*2} = 6$	$n_{*3} = 5$	$n_{**} = 15$

Intuitively, we expect the intra-cluster overlap to increase the value of $P(S_{c_*})$. However, equation 6.4 yields the incorrect result of $31/105$ which is smaller than the correct one $21/45$. This is due to the fact that sampling without replacement assumption no longer holds. Indeed, there are not $\binom{n_{**}}{2} = \binom{15}{2} = 105$ ways to select 2 documents as that would allow the possibility of selecting the same document twice. The right number of ways to select 2 elements is still 45 and it is given by $\binom{|D|}{2} = \binom{10}{2}$. However, the events S_{c_j} to sample two documents from the same class j are no longer independent. Therefore, they cannot be added as in (6.4). Basically, when class or cluster overlap exists, the contingency matrix bins do not represent mutually exclusive events. Thus, the value of $P(S_{c_*})$ when classes overlap exists is given by:

$$P(S_{c_*}) = \sum_j h(|D|, |c_j|, 2, 2) - J(C) \tag{6.13}$$

where $J(C)$ is the probability that a selected pair of documents belong to two classes simultaneously, defined by the expressions:

$$J(C) = \sum_j \sum_{j'>j} P(S_{c_j} \cap S_{c_{j'}})$$

and, using the hypergeometric distribution:

$$J(C) = \sum_j \sum_{j'>j} h(|D|, |\{S_{c_j} \cap S_{c_{j'}}\}|, 2, 2)$$

However, the above formulas deal with the case where the classes overlap is restricted to pairs. The case where general classes overlap is concerned is more complex from both the theoretical and computational point of view and will be presented in a different work. Formula (6.13) is a re-expression of (6.4) under the general addition rule of probability for non-independent events[1]. For instance, if any of the pairs $\{(d_4, d_8), (d_4, d_9), (d_4, d_{10}), (d_8, d_9), (d_8, d_{10}), (d_9, d_{10})\}$ is sampled, then S_{c_2} and S_{c_3} are both true and this results in a double count. The correct value of $P(S_{c_*})$ is obtained by subtracting the probability of the classes intersection:

$$P(S_{c_*}) = \left(\frac{\binom{4}{2}}{\binom{10}{2}} + \frac{\binom{6}{2}}{\binom{10}{2}} + \frac{\binom{5}{2}}{\binom{10}{2}} \right) - \left(\frac{\binom{4}{2}}{\binom{10}{2}} \right) = 0.55$$

Incomplete partitions

When hardening a soft-cluster solution generated by a topic model we potentially obtain overlapping and incomplete partitions, thus, the validation metrics should be sensitive to some form of "recall". In the FM index computation, the base assumption would be that the column

[1] Which states that: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

marginal totals correspond to the size of the classes, i.e. $n_{*j} = |c_j|$, and that the row marginal totals equals the size of the clusters, i.e. $n_{i*} = |u_i|$. As shown before, such an assumption is false when an overlapping exists with the same applying to cases where the clusters are incomplete. Measuring incomplete partitions with the FM contingency matrix is wrong. Indeed, it results to incorrectly reduce the number of successes inside the population by using n_{i*} instead of $|u_i|$. Furthermore, the possibility of cluster overlapping has to be taken into account. Therefore, the correct probability of selecting 2 documents from the same cluster will be given by:

$$P(S_{u_*}) = \sum_i h(|D|, |u_i|, 2, 2) - J(U) \tag{6.14}$$

where $J(U)$ accounts for the probability of selecting a pair of documents belonging to two or more clusters and it is given by adding up the probabilities of clusters intersections:

$$J(U) = \sum_i \sum_{i'>i} P(S_{u_i} \cap S_{u_{i'}})$$

and, by using the hypergeometric distribution:

$$J(U) = \sum_i \sum_{i'>i} h(|D|, |\{S_{u_i} \cap S_{u_{i'}}\}|, 2, 2)$$

It is worthwhile to notice that formula (6.14) is valid also in the case where clusters do not overlap. However, although FM can be corrected to take into account some of the effects of partitions incompleteness and/or overlap, we consider that its interpretation is more biased toward measuring partition similarity and thus we find it valuable to study new metrics that can serve better to estimate semantic coherence.

6.4 Proposed metrics

In this section, we introduce a version of the FM index adjusted for overlapping and incomplete clusters. Two overlapping "precision" metrics together with their probabilistic interpretations are given. Finally, we discuss the computation of a kind of cluster "recall" which can be used to achieve a single metric performance.

6.4.1 Generalized Fowlkes-Mallows Index (GFMI)

As discussed in previous section, if the FM index is expressed in terms of the contingency matrix it can not be used to validate neither overlapping nor incomplete clusters. The reason is that while its' addition terms are hypergeometric probabilities, they would use an incorrect population size in the case where cluster overlapping is concerned. However, we have shown that when re-expressing the FM index in terms of the hypergeometric distribution and by correcting its formula so as to use the cluster size $|u_i|$ and the class size $|c_j|$ the probabilities $P(S_{c_*})$ and $P(S_{u_*})$ are correct under the assumption that the maximum overlap equals two. Therefore, the last step required to obtain a generalized version of the FM index requires to generalize the computation of $P(S_{u_*c_*})$ in such a way that non-independent events $S_{u_ic_j}$ are correctly taken into account. This generalization requires to compute the probability of the events intersection. For the whole contingency matrix, the sum of the probabilities of the intersection between "bins" will be denoted by:

$$J(U,C) = \sum_{ij}\sum_{i',j'} P(S_{u_i c_j} \cap S_{u_{i'} c_{j'}})$$

where $i' > i$ and $j' > j$ and where by using the hypergeometric probabilities we obtain:

$$J(U,C) = \sum_{ij}\sum_{i',j'} h(|D|, |\{S_{u_i c_j} \cap S_{u_{i'} c_{j'}}\}|, 2, 2) \qquad (6.15)$$

Notice that the computation of $J(U,C)$ requires to create an additional "overlap matrix" consisting of $(|U| \times |C|)^2$ elements. Finally, the generalized result for $P(S_{u_* c_*})$ is given by:

$$P(S_{u_* c_*}) = \sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U,C) \qquad (6.16)$$

Thus, the generalized version of the metric can be defined as the geometric average of:

- the probability of 2 randomly sampled documents belong to the same class, given they belong to the same cluster, i.e.:

$$P(S_{c_*}|S_{u_*}) = \frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U,C)}{\sum_i h(|D|, |u_i|, 2, 2) - J(U)}; \qquad (6.17)$$

- the probability of 2 randomly sampled documents belong to the same cluster, given they belong to the same class, i.e.:

$$P(S_{u_*}|S_{c_*}) = \frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U,C)}{\sum_j h(|D|, |c_j|, 2, 2) - J(C)} \qquad (6.18)$$

In conclusion, the generalized version of the FM index, which will be referred to as GFM, is given by[2] :

$$\frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U,C)}{\sqrt{[\sum_i h(|D|, |u_i|, 2, 2) - J(U)]\,[\sum_j h(|D|, |c_j|, 2, 2) - J(C)]}} \qquad (6.19)$$

[2] We are aware that this formulation may not be accurate on extreme cases of *very* overlapped collections, however we will show that the hypothetical error, which is in fact an underestimation of actual probabilities is negligible in real-world corpora such as Reuters-21579. Such insights of pure theoretical interest will be presented in future works

6.4.2 Partial Class Match Precision

This metric is inspired from the notion of precision utilized in the IR field. The Partial Class Match Precision (PCMP) measures the probability of randomly selecting two documents from the same class taken from a randomly sampled cluster. In contrast to FM, where we are concerned with the random sampling of two documents d_1 and d_2 from the documents corpus, PCMP requires to first randomly sample a cluster and then randomly sample two documents from the sampled cluster. In order to clearly differentiate both random events, we use $\tilde{S}_{c_*}$ to denote the event of selecting two documents belonging to the same class sampled from a given cluster. Formally, the PCMP metric is defined as follows:

$$P_{PM} = P(\tilde{S}_{c_*}) = \sum_i P(\tilde{S}_{c_*}|u_i)P(u_i) \tag{6.20}$$

where the prior probability of selecting the cluster u_i is given by $P(u_i) = n_{i*}/n_{**}$.

PCMP measures the probability of the event $\tilde{S}_{c_*}$, i.e. to sample two documents from the same class, *after* having randomly selected a cluster. However, the computation of each individual $P(\tilde{S}_{c_*}|u_i)$ also needs to be generalized to the case of classes overlapping. Therefore, we need to add up the probability of selecting two documents from each class comprised within the cluster $P(\tilde{S}_{c_j}|u_i)$ under the general rule of the addition for non-independent events which implies discounting the probability of a success in two classes simultaneously. So, each individual $P(\tilde{S}_{c_*}|u_i)$ would be given by:

$$P(\tilde{S}_{c_*}|u_i) = \sum_j P(\tilde{S}_{c_j}|u_i) - J(u_i) \tag{6.21}$$

where $J(u_i)$, which represents the probability to sample two elements from two or more classes when selecting documents d_1 and d_2 which belong to cluster u_i, is given by:

$$J(u_i) = \sum_j \sum_{j'>j} P(\{S_{u_i c_j} \cap S_{u_i c_{j'}}\}|u_i) \tag{6.22}$$

The previous equation requires computing a half-matrix of class overlaps, and then computing the probabilities of selecting two from each "bin".

$$J(u_i) = \sum_j \sum_{j'>j} h(|u_i|, \{S_{u_i c_j} \cap S_{u_i c_{j'}}\}, 2, 2) \tag{6.23}$$

This metric is designed to work well with multi-labeled documents corpus. The name *"Partial"* comes from the fact that in a multi-label setting the two randomly sampled elements d_1 and d_2 can be associated with many classes. As long as one of their classes matches we will consider the result to be semantically coherent, thus as success. We consider that this property of the metric is a valuable feature to focus on measuring semantic coherence rather than mere partitions similarity. For instance, there is not a unique way to achieve the maximum evaluation. In fact, we can visualize two clustering solutions that will obtain the maximum evaluation under this setting.

a) Creating one cluster for every class, and assigning all the elements in c_i to u_i, so that $k = |C|$.

b) Creating clusters of elements that share the exact same class labels.

Finally, we should highlight that this metric can be easily approximated via Monte Carlo simulation. We will use this method to show the correctness of the metric.

6.4.3 Full Class Match Precision (P_{FM})

The P_{pm} metric described in section 6.4.2 allows objects to be clustered among others with whom they share only one label. Take for instance 2 objects: $x_1 \in \{c_1 \cap c_2 \cap c_3\}$, $x_2 \in \{c_3 \cap c_4 \cap c_5\}$. If they happen to be in the same cluster and they are randomly selected, for the P_{pm} case it will be considered a "success". So, we consider that in some cases it could be valuable to measure the probability that two randomly sampled objects have exactly the same class labels. $A_c(x_1, x_2)$: *True* iff x_1 and x_2 belong to the same class. So, the number of clusters would be given by the number of elements of the power-set of C containing more than one element. $k = \{C_* : (C_* \subset \wp(C)) \wedge (\sum_{c_j \in C_*} |c_j| > 1)\}$

6.4.4 Recall metrics

In the IR field the "recall" measure represents the probability that a relevant document is retrieved. Therefore, for the clustering scenarios under consideration, when the completeness of the partition cannot be assumed, it is critical to provide clear ways to measure the completeness of the clustering.

Let N_c be the total number of class assignments, given by the sum of the sizes of every class:

$$N_c = \sum_j |c_j|$$

In overlapping and incomplete clustering we must not to rely on the values of the contingency matrix to compute recall values, given that they can account for duplicates and that they do not consider elements not included in clusters.

Class recall

If we are interested in measuring which classes are better captured by the clustering it is straightforward to compute a class recall value. We define this "class recall" as the probability that a document d, randomly sampled from the class c_j, is included in any cluster.

$$R(c_j) = P([x \in \cup_i^k u_i] | c_j) = \frac{|\bigcap_i^k \{u_i \cap c_j\}|}{|c_j|} \tag{6.24}$$

In other words, equation (6.24) means dividing the number of documents of class c_j that were recalled by any cluster u_i by the total number of documents belonging to class c_j.

Gross clustering recall

From previous expression, considering that the probability of selecting a class would be given by $P(c_j) = |c_j|/N_c$ it is possible to derive an unconditional expression to measure the recall of the whole clustering.

$$R_U = P(x \in \cup_i^k u_i) = \sum_j P(x \in \cup_i^k u_i | c_j) P(c_j) \tag{6.25}$$

Where the probability of selecting any class would be given by $|c_j|/N$. So, it can be conveniently expressed as:

$$R_U = \frac{1}{N_c} \sum_j R(c_j)|c_j| \tag{6.26}$$

6.4.5 Semantic-Coherence F-Score

In retrieval and classification it is widely known that it is trivial to achieve high recall at the expense of precision. So, traditionally they are averaged into a single metric, the F-Score. The traditional F-Score is nothing but the harmonic mean between precision and recall. Almost any two probabilities can be averaged in that way, however, for the particular case of topic-model validation we are interested in balancing the best measure for semantic coherence with the best measure for completeness, so our proposed metric is defined by averaging (6.21) and (6.26):

$$F_o = \frac{2P_{PM}R_U}{P_{PM} + R_U} \tag{6.27}$$

Notice that the selection of (6.21) and (6.26) comes at the expense of not penalizing some clustering dissimilarities. So, if the ultimate performance criterion is the partition similarity, then the GFM may be a best metric of choice.

Both components of the F_o metric are micro-averaged so that every document has the same weight on the result. The micro-averaging effect is achieved by the marginalization step performed in eq. 6.21 and eq. 6.26 in order to work with unconditional probabilities.

6.4.6 Empirical approximation to the metrics

Among the properties of the probabilistic metrics is that they can be computed in 3 different ways:

Enumeration For small clusters and classes ($n < 100$) , specially in very overlapped cases it is possible to enumerate all the sample space, which includes all the ways to select two elements from the cluster. Then, count all the pairs that have a class in common to obtain P_{PM} or to have exactly the same classes to compute P_{FM}.

Formula The enumerative expression has the advantage of accuracy but has the inconvenience of the computational complexity. The number of pairs required to analyze is bounded by $O(n!)$, so, the hypergeometric formulas such as the ones described earlier in this chapter, make sense whenever the size of the cluster in question is large and the overlap of more than 2 classes is not significative.

Empirical In any other case, or whenever the actual user experience is to be simulated, an approximation to the probability values may be obtained by repeated random sampling. The main advantage of the sampling method relies on its implementation simplicity.

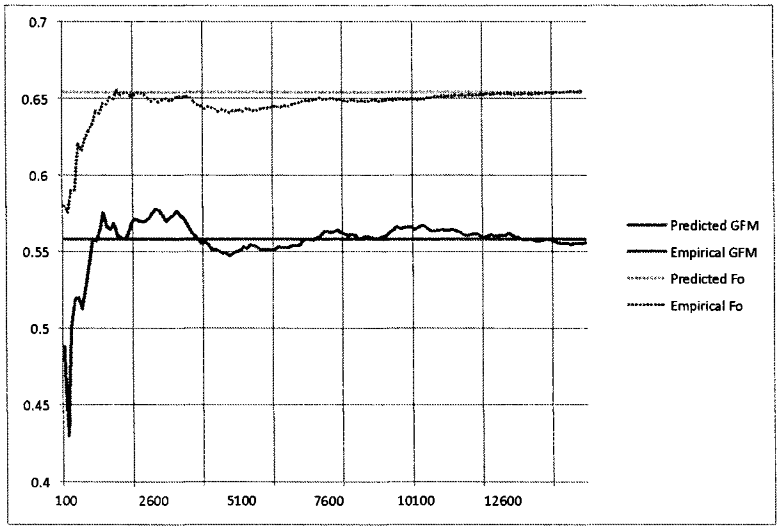


Figure 6.1: Monte Carlo approximation of GFM and Fo

First, in order to demonstrate the correctness of the GFM and Fo formulations, we performed some Monte Carlo simulations. To estimate the GFM metric the following procedure was performed:

1. Randomly sample a pair of documents.
2. Count if they belong to the same class.
3. Count if they belong to the same cluster.
4. Count if they belong to the same class and cluster.
5. Compute empirical values for $P(S_{u_* c_*})$, $P(S_{c_*}|S_{u_*})$, $P(S_{u_*}|S_{c_*})$ and GFM.

Then, in order to demonstrate the correctness of the Ppm and Fo formulations, the following simulation was performed:

1. Randomly select a cluster, based on its prior probability
2. Randomly select 2 documents from the cluster, count if they belong to the same class.
3. Randomly select a class based on its prior probability, then select a document for the class and count if it is included in the clustering.
4. Compute empirical values for $P(\tilde{S}_{c_*}|u_i)$, R_U and F_o.

Results of an individual simulation for K=90, t=0.2 are shown on figure 6.1 where it is shown how the empirical measurements converge to the predicted measurements.

6.5 Chapter summary

Generally speaking, it is not straightforward to validate topic models or soft-clustering algorithms using multi-labeled collections because many of the assumptions of the known metrics do not hold and usually are inadvertently ignored or oversimplified, leading to methodological flaws. So, in order to build a reliable evaluation and model comparison tool, we have worked on two mathematical artifacts:

1. We have extended a known clustering similarity metric, the Fowlkes-Mallows Index (FM), in order to make it work in the overlapping case.

2. We have proposed simpler metrics of based on familiar concepts of recall, precision and the notion of *semantic coherence*, interpreted as the probability of randomly selecting two documents belonging to the same topic after having randomly selected a cluster.

We strongly believe this metrics can help on avoiding common methodological issues that arise when attempting to validate soft and overlapping clustering solutions. Throughout the rest of this dissertation, we will use the P_{PM} metric combined with R_U as metrics of choice given their simplicity of interpretation and implementation.

Chapter 7

Experimental results

7.1 Introduction

In this chapter we will present an analysis of the QTM framework from the quantitative and experimental point of view. The evidence will be helpful to gain insights into issues like:

- The effects of the different heuristics and settings in performance, including candidate selection method, and evaluation function.
- Determining how the method stands against other alternatives such as LDA.
- The efficiency of the different strategies in terms of computational resource usage.

7.1.1 Notation review

Briefly recalling from section 5.3, there are two kinds of QTM models under consideration:

1. Models for non-overlapping partitions: $U \sim Q_{TM}(c_g, \varphi, z, c_s, n_c, k, h)$ and,
2. Models for overlapping partitions : $\hat{U} \sim \hat{Q}_{TM}(c_g, \varphi, z, c_s, n_c, k, h, s_m, s_t)$,

Where the model parameters are defined as follows:

- c_g : Candidate query generation method.
- φ: Candidate query evaluation function.
- z: Minimum query results allowed.
- c_s: Candidate query selection method.
- n_c: Number of selected queries.
- k: Number of topics to construct.
- h: Number of hits per query to include in the query-vector representation.
- s_m: Soft-clustering generation method.
- s_t: Soft-clustering inclusion threshold

7.2 Experimental framework

In order to produce our results we used the Reuters-21578 corpus, ModApte split, considering only the documents that are labeled with topics. Numbers were replaced by a unique symbol. After this pre-processing, documents with less than 10 unique words were removed. Both ModApte training and test sets were included in the corpus, making a total of 10,468 unique documents and 117 ground-truth classes. For comparison purposes, LDA was run with parameters $\beta = 0.01$, $\alpha = 50/K$ running 1,000 iterations of Gibbs sampling.[1]

As QTM models are subject to a number of choices that determine their performance an experimental framework was built in order to perform a semi-automated exploration of the parameter space. In total we produced 17,984 different models with unique parameters combinations, from which we have created 3 consistent and standardized experiments subsets.

Notice that although a QTM model is defined by 7 or 9 parameters, not all of them are equally interesting; for instance, the number of clusters k or the threshold s_t. In such cases we tried to "discount" their effects rather than analyzing them. So, for every combination of parameters $\{c_g, \varphi, z, c_s, n_c, h\}$, the value of k was automatically varied within 10, 30, 50, 70, 90, 117. In the same way, for overlapping models, the parameter s_c was varied within 0.05, 0.10, 0.15, 0.25. 0.30, 0.35. Finally, the value of F_o was averaged for every group of (k, s_c), assuming that the proper selection of k and s_c are more properly implementation concerns. The experimental subsets are defined based on the inclusion criteria defined below.

As a side remark it is important to mention that in the proposed model specification scheme there are two kind of parameters, the ones representing strategies like c_g or φ and the ones which are merely numerical settings such as k or s_c. In this regard, when we say that some parameters are not as interesting as others it doesn't mean that their effects are less significative, in fact, models can be very sensitive to such settings. However, purely numerical parameters only require being properly set and don't provide much insights about the underlying principles that make a particular model work. So, on one hand we are interested in analyzing the behavior of the parameters that embody specific principles and on the other hand we are interested in determining which heuristics reduce the model sensitivity to the numerical parameters, thus increasing model robustness.

7.2.1 Standard Set (ST)

This is the main experiment set, contains information of 15,617 models. The goal of this data set is to perform comparative analyses of the main evaluation functions, candidate generation methods and candidate selection methods as well as the measuring the impact of the numeric parameters n_c and h that determine the problem size. The models included in the set have the following properties:

- Two candidate selection methods: Global $c_s = G$, and by document $c_s = D$.

[1] LDA results were provided by Davide Magatti

- For $c_s = G$, two generation methods: spotsigs, $c_g = S$ and n-grams $c_g = N$.
- For $c_s = G$, selecting $n_c = \{5000, 15000, 20000, 25000\}$ total queries.
- For $c_s = D$, selecting $n_c = \{1, 3, 5, 10\}$ queries per document.
- By-document candidates were only generated using spotsigs. $c_g = S$.
- Three evaluation functions $\varphi = \{KL, QMI, QF\}$.
- For the functions KL and QMI, the value of $z = \{2, 10\}$.
- All the models were run for $h = \{10, 100, 500\}$.

7.2.2 Alternative Candidate Generation Methods (ACG)

The ACG data set contains 1455 instances and was designed to test alternative candidate generation methods, with particular interest on measuring the effects of using different subsets of the n-gram query set. The rest of inclusion criteria were:

- Candidate generation methods evaluated were: Spotsigs, $c_g = S$; N-grams, comprising unigrams and bigrams $c_g = N$; Unigrams only, $c_g = U$; Bigrams only, $c_g = B$.
- The evaluation function for all the cases was $\varphi = QF$.
- Candidate selection was $c_s = G$ for all the cases.
- Number of selected queries was $n_c = \{5000, 15000, 20000, 25000\}$ total queries.
- All the models were run for $h = \{10, 100\}$.

7.2.3 Alternative Force Functions (AEF)

The AEF data set contains 3456 instances and was designed to benchmark the previously introduced functions KL, QMI and QF against an hypothetical term-document mutual information function (MI). The experiment was motivated after observing the rather decent performance of QF, which lead to the hypotheses that QF works well because it is an approximation of a "wiser" mutual information function. The inclusion criteria for this dataset were:

- Four evaluation functions $\varphi = \{KL, QMI, QF, MI\}$.
- Candidate selection was $c_s = G$ for all the cases.
- Candidate generation was $c_g = N$ for all the cases.
- Number of selected queries was $n_c = \{5000, 15000, 20000, 25000\}$ total queries.
- All the models were run for $h = \{10, 100, 500\}$.

7.3 Model efficiency

As stated in our research goals, the resource usage efficiency is an important dimension of quality. So, in order to measure the efficiency of the models we have decided to observe the size of the last phase of the process, which is the co-clustering phase. Certainly it is not the only cost metric, but it is a variable that clearly provides a hint about the size of the problem being solved.

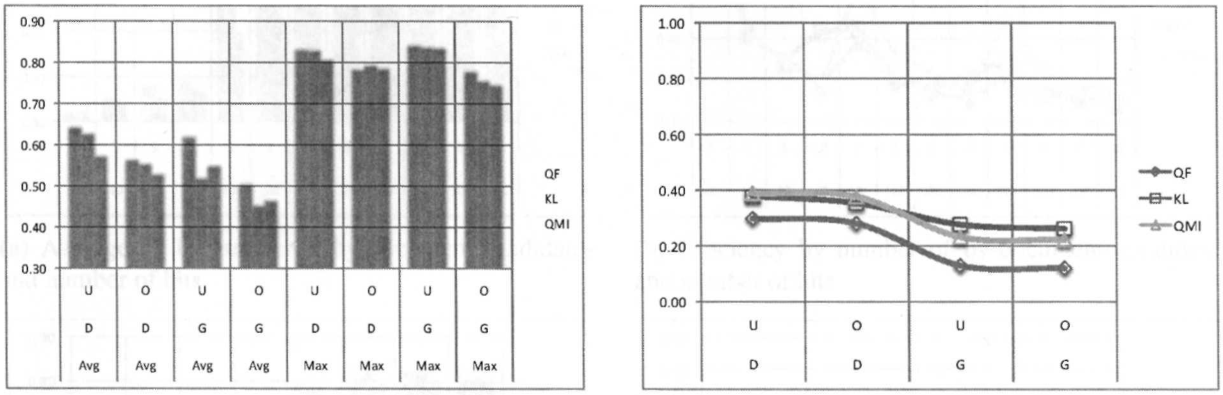
So, we are measuring efficiency as the performance obtained in relation to the size of the co-clustering contingency matrix, measured in number of non-empty cells. The concrete measurement performed is defined as the number of *f-score percentage points obtained by thousand of matrix cells* and it is computed as:

$$E_f(Q_{TM}) = \frac{100F_o(U)}{1000C} = \frac{10^5 F_o}{C} \tag{7.1}$$

Where U is the set of partitions obtained by model Q_{TM} and C is the number of non-empty cells in the co-clustering contingency matrix.

7.4 Evaluation of the main QTM heuristics

7.4.1 Candidate selection method



(a) F_o by candidate selection method, evaluation function and model type

(b) Efficiency by selection method, evaluation function, model type

Figure 7.1: Performance by candidate selection method, evaluation function and model type

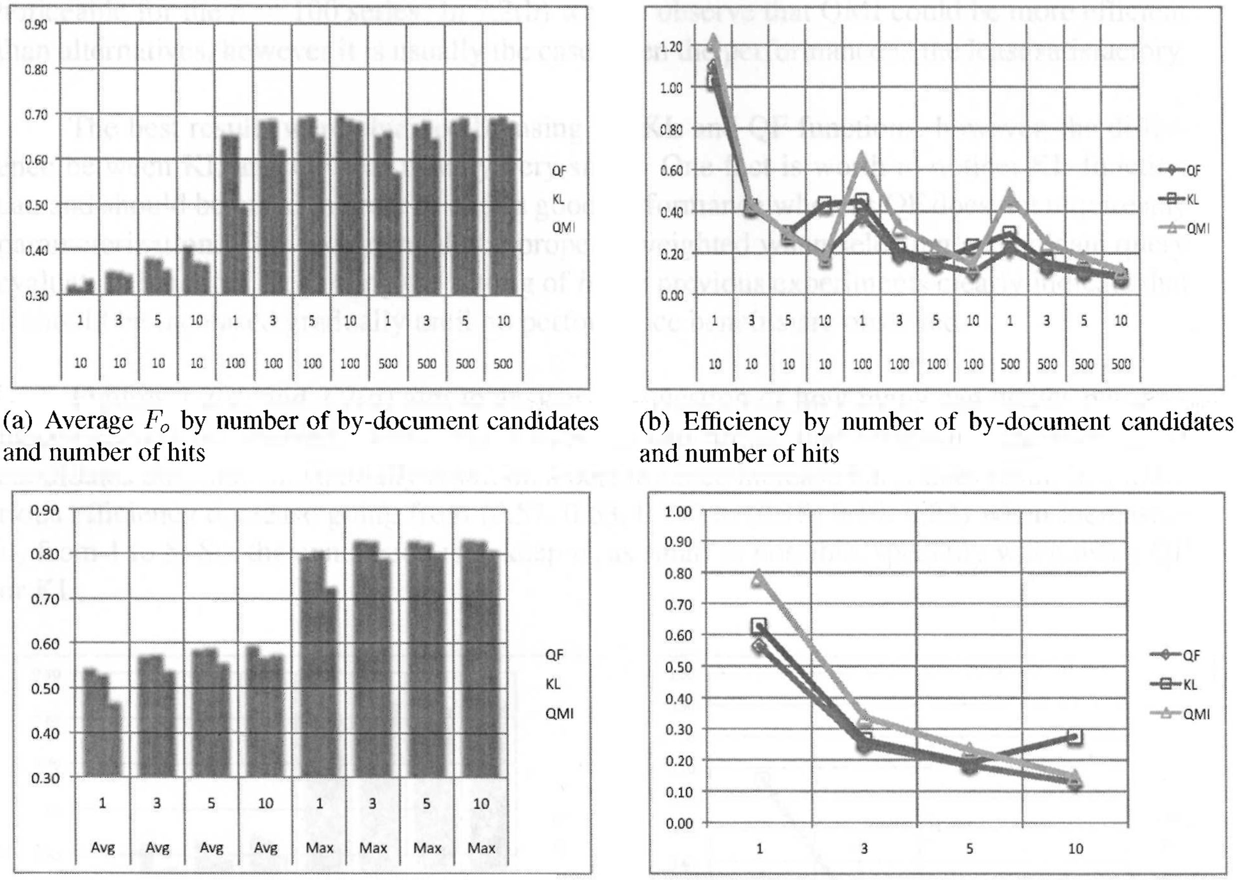
For this analysis we use the ST experiment set and we're interested in comparing two candidate query selection strategies:

- By-document Candidates (D). Using $n_c = \{1, 3, 5, 10\}$ queries per document.
- Global candidates (G). Using $n_c = \{5000, 15000, 20000, 25000\}$ total queries.

In figure 7.1(a) we report the average and maximum performance for the two strategies combined with the type of model generated, using the literal U to represent non-overlapping models and O to represent overlapping models. Given that the setting of n_c is somehow arbitrary, the obtained averages cannot be assumed to represent the efficiency of the method. Also, a different average and maximum was computed for the 3 evaluation functions described in chapter 4.

In general, from this experiment we can observe that the by-document selection method $c_s = D$ is in general helpful to obtain better average performance, specially when creating overlapping models, given that:

- For overlapping clusterings, the by-document selection $c_s = D$ showed better performance than $c_s = G$. Average performance of group O,D was: 0.57, 0.56, 0.53, while for group O,G was 0.51,0.46, 0.47, for functions QF, KL and QMI respectively.
- In terms of maximum performance, both strategies produced very similar models. The apparent differences can be perceived only between U and O groups, where the non-overlapping model (U) is usually superior to the overlapping by 3 to 5 points.



(a) Average F_o by number of by-document candidates and number of hits

(b) Efficiency by number of by-document candidates and number of hits

(c) F_o by number of by-document candidates

(d) Efficiency by number of candidates per document

Figure 7.2: Performance by hits and number of by-document candidates

Good performance can be obtained using any of the strategies, however, the average performance and the efficiency results on figure 7.1(b) show that it is somehow easier to find a good and more efficient model using by-document selection in combination with F or KL functions. Interestingly, the best function for $c_s = D$ under the current settings was clearly the pure query frequency QF. In terms of efficiency, $c_s = D$ nearly doubles $c_s = G$.

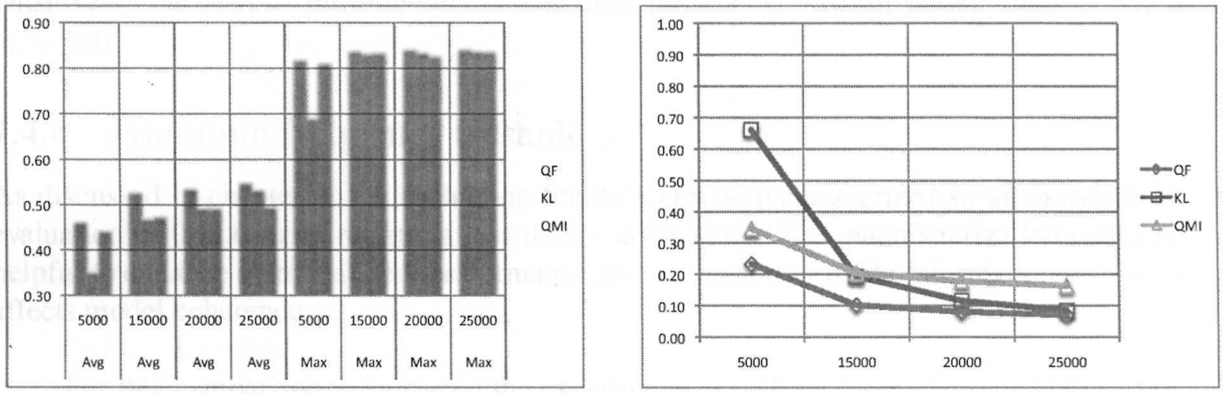
7.4.2 Hits and by-document candidates

Previous results are encouraging to analyze a bit deeper the behavior of the by-document selection method, this time in combination with the number of hits selected for the final clustering phase. The results of the experiment are summarized in figures 7.2. In order to obtain the previous results, we used dataset ST, now grouping by the value of $h = \{10, 100, 500\}$.

In figures 7.2(a) and 7.2(c) it is evident that using $h = 10$ hurts the average performance dramatically. On the other hand, increasing h from 100 to 500 does not improve the performance, moreover, it seems slightly reduced and the efficiency drops from the mid 30's to the 10's. The QMI evaluation function exhibits a greater sensitivity to the value of n_c. This is very noticeable for the $h = 100$ series. In 7.2(b) we can observe that QMI could be more efficient than alternatives, however it is usually the case when the performance is the least satisfactory.

The best results were obtained by using the KL and QF functions, however, the difference between KL and QF was usually very small. One fact is worth to notice; KL function can and should be tuned in order to obtain good performance whereas QF does not require any parameterization. This factor should be properly weighted when selecting a candidate query evaluation function. Regarding the setting of h, the previous experiments clearly indicate that it should be increased gradually until no performance benefits are observed.

Figures 7.2(c) and 7.2(d) aim to answer the question of how many candidates per document should be selected. From the results we can notice that increasing the number of candidates does not substantially result on a performance increase but it does result in a notorious efficiency decrease going from (0.57, 0.63, 0.79) to (0.18, 0.19, 0.23) when increasing n_c from 1 to 5. So, the conclusion is to keep n_c as small as possible, specially when using QF or KL.



(a) F_o by number of global candidates and evaluation function

(b) Efficiency by number of global candidates and evaluation function

Figure 7.3: Performance by number of global candidates

7.4.3 Number of global candidates

Although it has been shown that by-document candidate is a more elegant option in terms of performance versus efficiency trade-off, it requires more careful tuning of the n_c parameter and evaluation function. So the global candidate selection strategy is still worth analyzing, as it is simpler to implement and tune.

In figures 7.3(a) and 7.3(b), generated from ST data set we can observe how by gradually increasing the number of global candidates, the performance is improved until adding more candidates results in no-improvement, which happens around $n_c = 20,000$. On the other hand, it is also worth to notice that the KL function is the only one that appears to be more

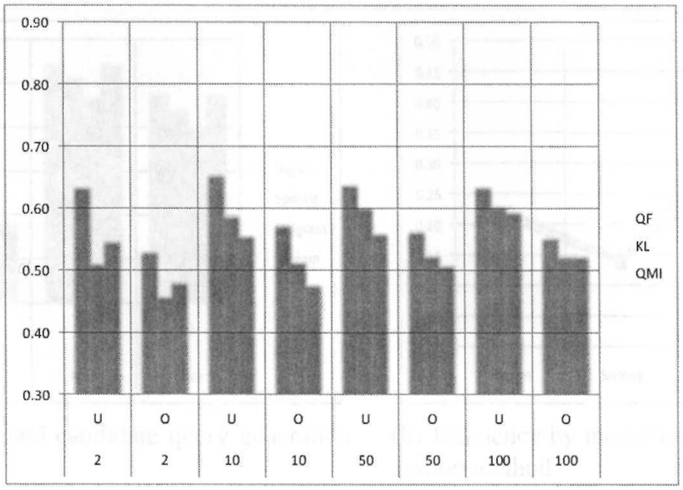


Figure 7.4: Average performance by model type, function and min frequency treshold

sensitive to the lack of information, as it dramatically underperform the alternatives when $n_c = 5000$.

7.4.4 Minimum frequency treshold z

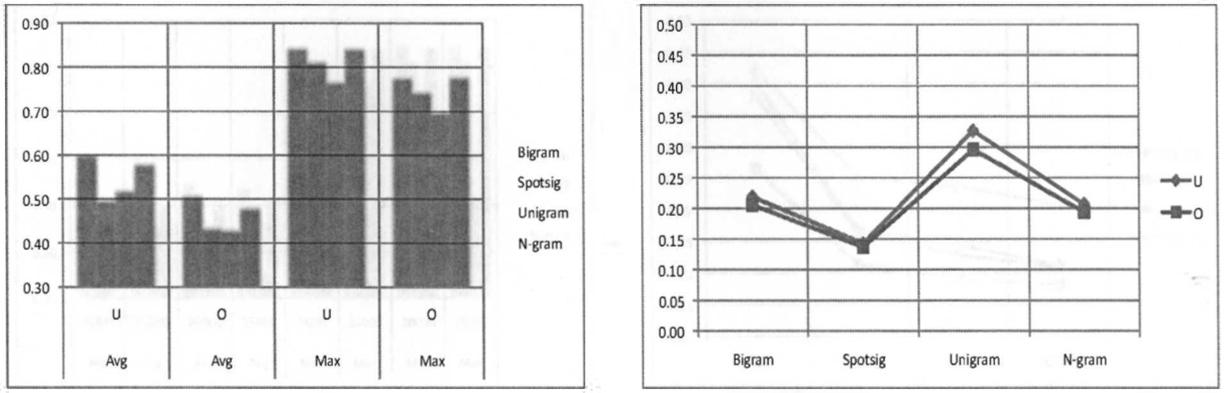
As discussed in chapter 4, the evaluation function can be parameterized so as to assign an evaluation of 0 to queries returning less than z documents. This parameterization could be helpful to enhance computational performance, but we need to provide information on how it affects model coherence.

For this comparison we grouped the experiments on ST set by model type U,O and by the distinct values of z. As shown in figure 7.4, the KL function is the one that exhibits the lowest sensitivity to the setting of z and being the one performing better on average for all cases. When $z = 100$ the functions KL and QMI are closer to QF.

Please notice that this result should not lead us to conclude that QF is "better" than KL and QMI only that the setting of z. However, it sheds some light on the sensitivity to parameterization of KL and QMI, that is somehow avoided by QF.

7.5 Alternative candidate query generation methods

In this section we're concerned with analyzing alternative query generation methods. We're interested in comparing the spotsigs (S), n-grams (N) and also considering two subsets of the n-grams: Unigrams and Bigrams, as defined in data set ACG. All the performance results of this section were evaluated using the frequency function.



(a) F_o by model type and candidate query generation method

(b) Efficiency by model type and candidate query generation method

Figure 7.5: Performance by model type and query generation method

7.5.1 Performance by model type and query generation method

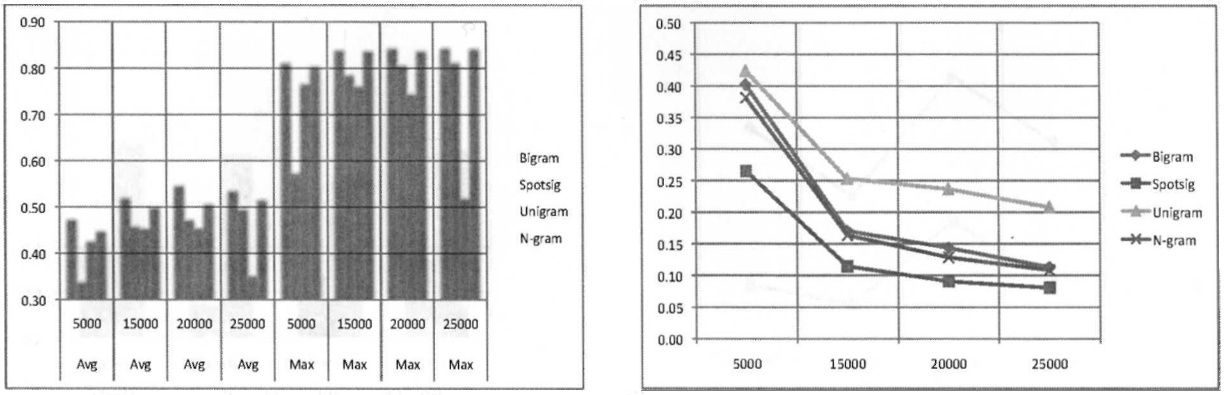
Figure 7.5 shows average performance and efficiency of the models grouped by model type and candidate generation method. The most interesting result of this set is that Bigrams perform better on average than n-grams, from which they are a subset. So, the inclusion of the Unigrams (the most frequent words) does not add value to the model, moreover, it can decrease it. Notice that the usage of Unigrams makes the model comparable to a traditional term-by-document clustering algorithm. The performance of the SpotSigs heuristic is lower on average although it is possible to find good models.

This experiment supports a key conclusion of this work: *selecting a good set of candidate queries is important to create good models and is better than selecting only the vocabulary (unigrams) of the collection.* From the theoretical standpoint, the result clearly shows how different subset of queries, produced by different rules can contain different amounts of semantic information. This fact contrasts with pure term-by-document clustering approaches that do not provide a mechanism to leverage the term proximity or sentence co-occurrence information.

7.5.2 Candidate generation method and number of selected candidates

A more detailed analysis on the effects of the candidate generation method together with the number of global candidates is shown on figures 7.6. Two key facts can be observed from charts:

- Spotsigs perform dramatically bad if n_c is low.
- Performance of Bigrams is consistently better than N-Grams as n_c increases.
- Performance of Unigrams drops sharply after increasing the number of selected candidates to 25,000, which suggests that the inclusion of unfrequent terms is the cause of the performance penalty for $c_g = U$ and $c_g = N$.



(a) F_o by candidate query generation method and number of selected global candidates

(b) Efficiency by candidate query generation method and number of selected global candidates

Figure 7.6: Performance by alternative query generation method and number of selected candidates

In summary, the spotsigs candidate generation method should be avoided particularly if the number of selected candidates is required to remain low. So far, the best global candidate generation heuristic to use in combination with the frequency evaluation function was the use of corpus bigrams. This result suggests that more experimentation is required to determine the performance of several other orders of n-grams in combination with other selection and evaluation heuristics.

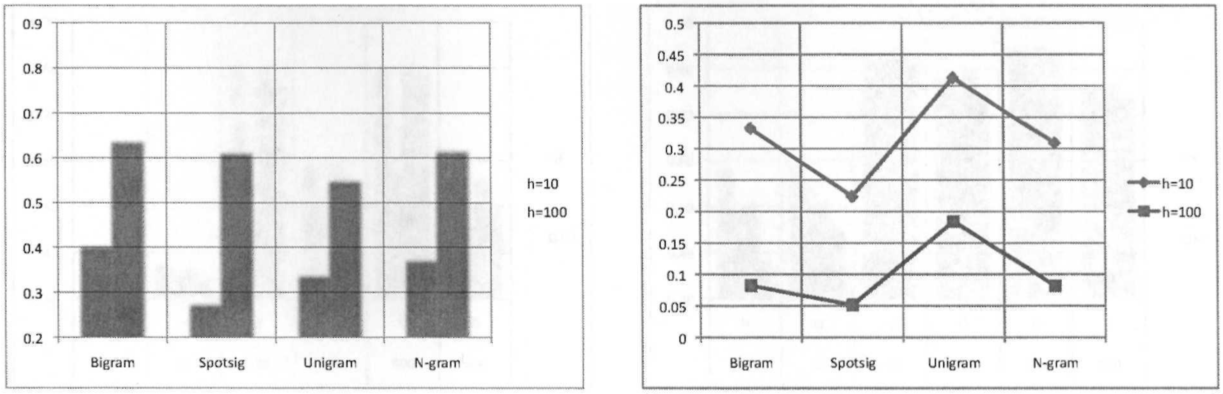
7.5.3 Candidate query generation method and hits

Figure 7.7 show the behavior of the different candidate generation strategies in relation to the hits parameter. The results are in line with the findings that the bigram outperforms n-grams method from which is a subset. In addition, we are also able to point out the greater sensitivity of the spotsigs method with respect to the hits parameter. As shown in figure 7.7(a), while the performance of bigrams goes from 0.40 to 0.63 when increasing the number of hits, the performance of spotsigs method goes from 0.27 to 0.61, which is more than double. So, besides the number of candidates, the spotsigs method should be avoided if h is required to be low.

7.6 Alternative evaluation functions

7.6.1 Semantic force as query contribution to MI

One of the most intriguing results from the analysis presented in section 7.4 was the superior performance of the frequency function in contrast to more complex functions, such as KL and QMI. The evidence lead us to revise the underlying working hypotheses which pointed out to explore functions based on query alterations measurements, motivated by the topic identification results from chapter 4.



(a) Average F_o by candidate query generation method and number of hits

(b) Efficiency by candidate query generation method and number of hits

Figure 7.7: Performance by candidate query generation method and number of hits

Results from 7.4 clearly indicate that the topic modeling problem works best under different heuristics than the topical-query identification problem discussed in chapter 4. As a result, going back to information-theoretic fundamentals, a new working hypothesis was proposed suggesting that the good performance of the QF function is due to its ability to capture mutual information between queries and documents, resembling the principles of the Information Bottleneck (IB) method [62], [54].

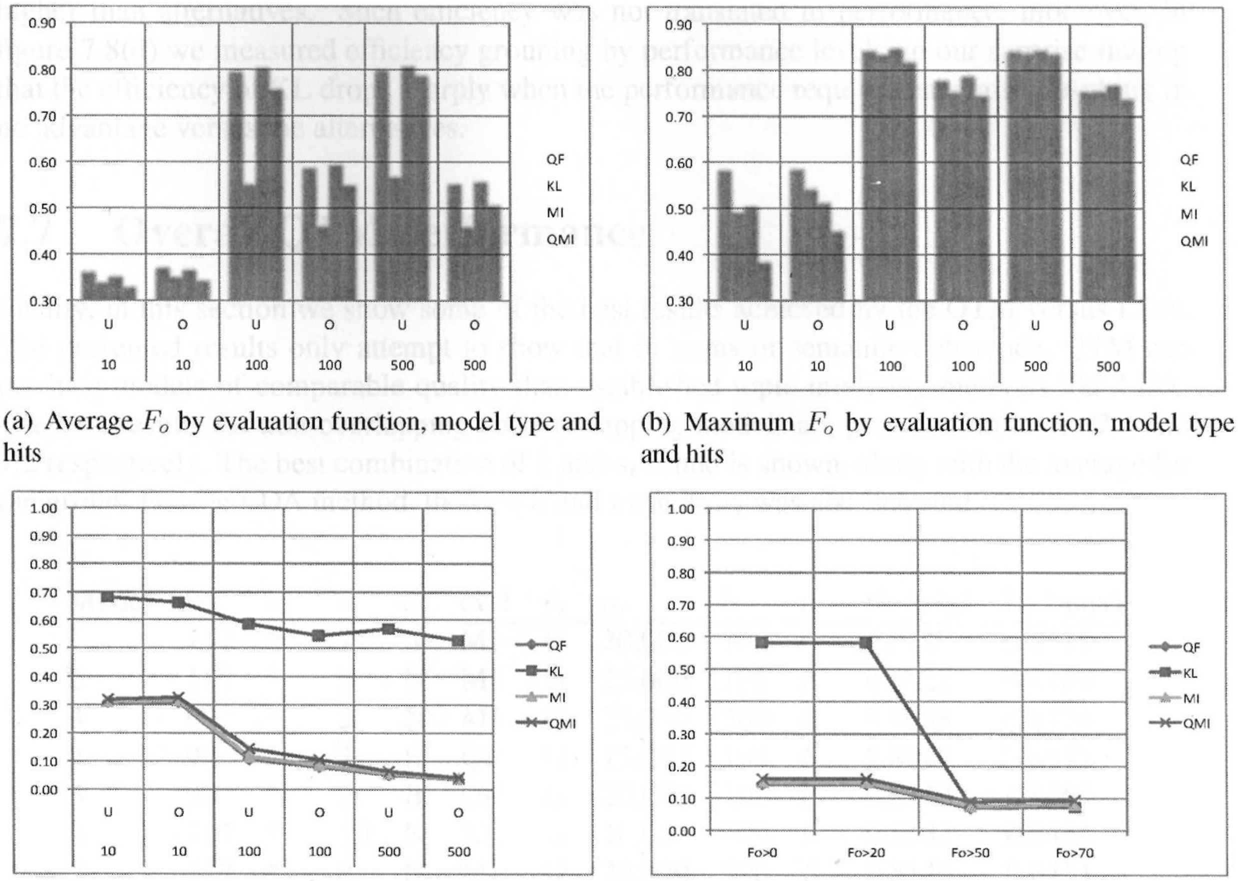
In the IB method, the documents of the collection are modeled as a probability distribution of terms. In our case, the model was adapted to let each document be represented by its conditional probability distribution of candidate topical-queries, defined by:

$$p(q|d) = \frac{n(q|d)}{\sum_{q' \in Q} n(q'|d)} \tag{7.2}$$

Where $n(q|d)$ is the number of ocurrences of the query q in the document d. So, we could define a candidate query evaluation criteria based on those queries in Q that contribute the most to the mutual information (MI) about documents, a quantity that may be expressed as follows:

$$MI = I(q; D) = p(q) \sum_{d \in D} p(d|q) log \frac{p(d|q)}{p(d)} \tag{7.3}$$

Where the probabilities in equation 7.3 were estimated using as sample space the full set of generated candidate queries Q_c which includes as many occurrences of every query as the number of times it was generated during the query generation process. So, the probability of selecting a document given a query was given by $p(d|q) = \frac{n(q|d)}{n(q)}$. The prior probability of a query: $p(q) = \frac{n(q)}{|Q_c|}$. The prior probability of randomly selecting a document: $p(d) = \frac{\sum_{q' \in Q_c} n(q'|d)}{|Q_c|}$. The denominator $|Q_c|$ represents the total number of candidate queries



(a) Average F_o by evaluation function, model type and hits

(b) Maximum F_o by evaluation function, model type and hits

(c) Efficiency by evaluation function, model type and hits

(d) Efficiency at minimum performance level

Figure 7.8: Performance of alternative evaluation functions

generated, which may include several occurrences of each query associated to different documents. In our case, $|Q_c|$ was simply the number of lines of the file emitted by the candidate query generator component (1,647,773).

7.6.2 Discussion

The experiments in figures 7.8 resume our findings concerning the relation of function QF and MI. In general, from what we can observe is that QF and MI in general exhibit very similar behavior, with QF being sligthly superior to MI for a low value of h and MI being sligthly superior to MI in any other case. We believe however that the results were so similar to make any strong conclusion, however, such similarity opens a promising line of improvements over the baseline results. So far, the best model we could produce resulted from the MI function described above.

In terms of efficiency, these last results also serve to establish some basic conclusions. First of all, from 7.8(c) we can observe that the efficiency of the KL function was substantially higher than alternatives. Such efficiency was not translated to performance, moreover, in figure 7.8(d) we measured efficiency grouping by performance levels, to our surprise finding that the efficiency of KL drops sharply when the performance requirements raise, resulting in no advantage versus the alternatives.

7.7 Overall QTM performance

Finally, in this section we show some of the best results achieved by the QTM versus LDA. The presented results only attempt to show that in terms of semantic coherence, QTM can produce models of comparable quality than established topic modeling methods like LDA. The best results for non-overlapping and overlapping models are presented in tables 7.1 and 7.2 respectively. The best combination of k and s_t found is shown, along with the average for the group. For the LDA method, the best k and treshold s_t was also selected.

Model	k	s_t	z	c_g	eval	c_s	n_c	h	n	Fo (avg)	Fo (max)
1	117	*	2	N	MI	G	20,000	100	6	0.8147	0.8517
2	117	*	2	N	MI	G	25,000	100	6	0.8122	0.8480
3	90	*	2	N	MI	G	25,000	500	6	0.8188	0.8474
4	90	*	2	B	QF	G	25,000	100	6	0.8012	0.8472
5	90	*	2	B	QF	G	20,000	100	6	0.8142	0.8469
6	117	*	10	N	KL	G	10,000	500	6	0.8144	0.8465
7	117	*	2	N	MI	G	20,000	500	6	0.8140	0.8461
8	117	*	2	N	QF	G	25,000	100	6	0.8157	0.8461
9	90	*	2	N	MI	G	15,000	100	6	0.8131	0.8459
10	117	*	2	N	QF	G	20,000	500	6	0.8121	0.8454
LDA	30	0.2	0	*	*	*	*	*	30	0.6711	0.7728

Table 7.1: Top-10 non-overlapping QTM models, by max Fo

Model	k	s_t	z	c_g	eval	c_s	n_c	h	n	Fo (avg)	Fo (max)
1	70	0.15	2	S	KL	D	10	100	42	0.7008	0.7978
2	117	0.1	2	N	MI	G	20,000	100	42	0.6012	0.7918
3	117	0.1	2	N	MI	G	25,000	100	42	0.6021	0.7910
4	70	0.15	10	S	KL	D	5	100	42	0.6957	0.7909
5	70	0.15	100	S	KL	D	10	100	42	0.6928	0.7897
6	50	0.2	50	S	QMI	D	10	500	42	0.6883	0.7896
7	70	0.15	2	S	QF	D	10	100	42	0.6938	0.7875
8	117	0.15	2	S	QF	D	10	500	42	0.6878	0.7861
9	70	0.15	10	S	QF	D	10	100	42	0.6899	0.7860
10	70	0.15	100	S	QMI	D	10	100	42	0.6859	0.7854
LDA	30	0.2	*	*	*	*	*	*	30	0.6711	0.7728

Table 7.2: Top-10 overlapping QTM models, by max Fo

From the analysis of the presented tables, some interesting facts can be observed and deserve further consideration. Although, those statements cannot be considered conclusive given that the presented table contains data from multiple data sets.

- The global candidate selection method seems to hold a relation with the type of model. For instance, in the non-overlapping models, the best results were obtained using global candidates, in contrast, for the overlapping models, the best results were obtained using by-document candidates. This finding cannot be clearly noted from the pure observation of figure 7.1(a).
- The KL function plus Spotsigs can deliver top performance on overlapping models, although requires more tuning.
- The recently presented MI contribution function clearly dominates the non-overlapping top positions, despite that MI has not been run using $c_g = B$ which delivered the best results for QF than $c_g = N$.

7.8 Chapter summary

In this chapter we have presented a quantitative analysis of the main variables and heuristics influencing the performance and efficiency of QTM models. Some of the findings have led us to review initial working hypotheses and to establish promising lines of future work. So far, our most relevant findings can be summarized as follows:

- **Model type should determine the candidate selection strategy.** Both of the candidate selection strategies proposed can achieve good results. However global candidate selection performed better when producing non-overlapping models, and by-document candidates resulted more useful to produce overlapping models. Both kinds of models perform well in comparison to LDA in terms of semantic coherence.
- **More queries do not guarantee more performance.** In terms of candidate generation, the usage of bigrams outperformed the unigrams and the extended n-grams set, defined as the union of bigrams and unigrams. The striking finding was that the inclusion of unigrams in the model hurts performance. This result indirectly supports a key assumption of the QTM approach: That more information exists in a good set of candidate queries than in the vocabulary of the collection, thus, finding a good set of candidate topical-queries is a challenging problem.
- **The best strategies from identification task are not the best for modeling tasks.** Our initial approach that resulted from the topical-query identification results was to use the KL evaluation function in combination with Spotsigs and By-Document selection was validated. This approach was shown to be able to deliver top-performance on overlapping models, however, it requires extensive parameter tuning and enough data to perform well. In addition, its efficiency advantages quickly vanish when the performance requirements are raised.

- **Query contribution to MI was established as the new baseline evaluation function.** All the analyzed evaluation functions were found to be sensitive to basic model parameters such as the number of candidates n_c and number of query hits h, but not all of them were sensitive in the same degree to the minimum frequency threshold z or the number of selected candidates n_c. In particular, functions such as QF and contribution to MI, were found to be less sensitive to parameterization, which is an important practical aspect, given that less tuning is required to find good models.

Finally, we would like to emphasize that a key advantage of the validation approach developed in previous chapters is that the usage of labeled corpora to validate models have allowed us to automate the process of exploring the model variants. This validation automation capability has an additional methodological benefit as it allows quickly iterating between hypothesis-experimentation-theory. So, it was possible to try several alternative strategies without theorizing a lot, then, once finding the ones that work one can try to understand them and produce new hypotheses. So, in contrast to the currently accepted scenario in which a single model validation has to be performed by the means of a user study (in the course of days), this method allowed us to validate over 15,000 models better understand the underlying principles that drive model performance.

Chapter 8

Conclusions and Future work

8.1 Contributions

In this work we have proposed an alternative approach to the problem on constructing semantic models of text collections, formally defined as Discrete Topic Modeling (DTM). The DTM approach takes advantage of a simplified notion of topic, interpreted as a set of semantically similar documents, to essentially transform the Probabilistic Topic Modeling problem, which requires the usage of sampling methods, into an overlapping clustering problem.

Then, based on the hypothesis that there exist a set of search queries that can capture a great deal of the semantic information about the documents of the collection in the form of co-occurrence and proximity relations, we have proposed the Query-Based Topic Modeling framework, a discrete, information-theoretic method based on heuristics to generate, select and evaluate a set of candidate topical-queries which are then co-clustered along with the documents of the collection and finally used to estimate topic-document probabilities and overlapping clusters. All the steps of the QTM process have been designed following the map-reduce style and thus can be executed in parallel over clusters of commodity class machines.

We have shown that queries selected using information-theoretic heuristics can produce better models than the set of vocabulary terms, and as a consequence, the heuristics that can be used to generate and select those queries were studied. Through the study of the topic-identification task, in which the goal was to select the queries with the most meaningful words according to the labeled corpus data, we could conclude that heuristics, such as the KL-divergence between the conjunctive and disjunctive versions of any given query were among the best options. However, when the same function was applied to the full query-based topic modeling task, we found that its usage only resulted convenient if very specific conditions are met, thus, leading us to favor simpler heuristics focused on measuring the mutual information between queries and documents.

A key enabler of many of the results produced over this research was the set of probabilistic, semantic-coherence metrics presented in Chapter 6. The proposed metrics, inspired on the familiar concepts of recall and precision were designed to inexpensively validate overlapping and incomplete topic models using multi-labeled corpora, thus making an ideal tool to compare the QTM results to those produced using LDA. As a result of the benchmarks, we concluded that QTM can produce models of comparable and in many cases superior performance, with the added benefit of an algorithmic design that offers greater scalability. The proposed set of metrics were also of great value in order to develop a semi-automated experimental platform, that allowed us explore the effects of many different parameters and strategies on the final model quality. In this way, we could enhance the validity and strength of the results presented in this work.

In, summary, the contributions of this work were:

1. An alternative, discrete, formulation of the topic-modeling problem.
2. A set of probabilistic metrics to evaluate the quality of topic models in a repeatable and inexpensive way using multi-labeled corpora.
3. A set of information-theoretic heuristics that can be used to evaluate and identify topics in large collections.
4. An scalable query-based topic modeling framework (QTM), that can produce models of comparable quality of state-of-the-art methods in very large collections.

8.2 Future work

8.2.1 Theoretical aspects of QTM

The first line of future work is the one driven by the need of a better understanding of the probabilistic and information-theoretic principles and their interactions among the different phases of QTM.

After our experimental results concerning the query-evaluation functions, we found that one of the most promising lines of research in the QTM approach is to explore new candidate query generation heuristics. In general, try to provide richer and better answers to the question of what is the query generation method that can produce the best models. In this regard we have shown that a query set generated using bi-grams was superior to the collection unigrams (or vocabulary terms), and in fact was superior to the union of both sets. Also, we have shown that signature-calculation methods like SpotSigs [61], offered promising performance in combination with other heuristics, leading us to conclude that more research is required in order to understand the different types of co-ocurrence information captured by each of the methods.

Among the aspects of model building that we consider worth of further research we can name the effects of different ranking functions when constructing the query-vector document model of the collection. The issue can be specially critical when clustering web documents as the ranking could be used to introduce additional information such as the linkage structure of the collection. The effects of the document ranking when produced query-based topic models was in general omitted in the current scope. Also, the effect of introducing usage-based queries is still to be explored.

The last phase of the QTM process, which involves producing an overlapping document clustering from a hard partition, is subject to many improvements. Recently, some "soft" versions of the co-clustering have been presented proposing alternative ways to define probabilistic document-cluster associations, like the Bayesian Co-Clustering technique [52] and the Latent Dirichlet Bayesian Co-Clustering technique [68]. Being theoretically sound, those alternative co-clustering methods lead us to hypothesize that they may convey performance improvements in the construction of overlapping-clusters, however, the potential improvements should be evaluated in terms of their computational costs and their parallel processing ability.

8.2.2 Online QTM

The dynamic nature of some collections creates a series of challenges to be addressed by large scale algorithms. It is clear that if the collection receives constant updates or additions, the topic models need to be periodically rebuilt to ensure satisfactory performance. For the previous scenario it is important that a model rebuild process could be performed within an acceptable time window without significantly impacting user's experience, or more interestingly, that the model could be updated incrementally or "online". Therefore, exploring the feasibility and implications of producing an online, incremental version of QTM is a promising and of potential high impact line of future work.

The main challenge that arises in building an online version of QTM is given by the current architecture of the Map-Reduce programming paradigm, for which QTM has been designed. The problem is that current Map-Reduce implementations are designed to operate on large batches of data without access to the final result until the whole process has been completed. In order to deal with the limitations of Map-Reduce, Condie et. al [17] have proposed a "pipelined" version of Map-Reduce, named Hadoop Online Prototype (HOP). By using HOP authors have shown that is possible to allow downstream Map-Reduce processes to start consuming output of previous phases. Also, as reducers start processing data as soon as it is produced by the mappers, they can work on an approximation of their final solution by using a Map-Reduce version of the "online aggregation" technique initially presented in [31].

Given the previous work on HOP, the main challenges to be addressed in QTM would be to:

1. Produce running versions of the evaluation functions such that could be used within a pipelined Map-Reduce environment, as well as determining the best QTM strategies for the online scenario.

2. Implement online aggregation reducers for the summarization steps. Some analysis on the confidence intervals of the anticipated results may be required.

3. Produce an online version of the co-clustering algorithm. As noted by Böse, et. al [9],

this is important as it turns out that multi-step algorithms such as k-means or co-clustering are the hardest to implement as incremental or online algorithms. Therefore, performing a previous feasibility analysis is important as well as to compare the performance of the online version versus the "batch" version.

8.2.3 Topic model validation Metrics

Full metric generalization

The work on metrics presented here and in derived publications is still unfinished. Among the major aspects that require further analysis is the full generalization of the metrics to work under any degree of inter-class overlap, avoiding the need to fall back to enumerative computations that could quickly become impractical. Also, the applications of the Full-class match precision metric need to be further explored so as to determine the circumstances under which it provides better information.

Corpora sensitivity analysis

On the other hand, the presented probabilistic evaluation metrics have been only analyzed using Reuters-21578 corpus, which has particular properties in terms of the class structure, topic overlap and document length, thus, we believe that more analysis is needed to fully understand the corpora sensitivity of the measurements and at the same time determine the inherent "hardness" of a given corpus as well as the relative advantages of one modeling method in a specific corpus. In that regard, an experiment that can be performed would consist on a factorial analysis, having the corpora and the modeling methods as factors, and the semantic coherence score as response variable. The obtained results would be useful to draw and test hypothesis about the properties of the corpora that led to significative differences on the evaluations.

Wikipedia as external validation corpus

Another promising line of future work in the area of metrics is the usage of Wikipedia as validation corpus besides Reuters-21578. For instance, if one considers Wikipedia's categories as the topic-labels of a multi-labeled collection, then it is feasible in principle to compute some sort of semantic coherence metrics similar to those presented earlier on chapter 6. However, one important issue has to be addressed theoretically in order to perform robust measurements arises from the fact that Wikipedia's categories are not overlapping sets as in Reuters-21578, rather they are overlapping trees. Therefore, answering the question if a pair of documents belongs to the same class is not as straightforward and requires incorporating the notion of proximity within the tree. e.g. For instance, if after randomly selecting a pair of documents from a cluster, we found that one belongs to category *Algebra* and the other belongs to category *Geometry*, both (hypothetical) subcategories of *Mathematics*, under the current metrics our result would be that those documents are not semantically similar despite the fact that their categories share a common ancestor one level up in the tree. So, in order to use Wikipedia as validation corpora, metrics should be extended in such a way that we could take into account the tree-semantic similarity and then, measure the probability of obtaining a pair of documents similar by at least n-degrees. This approach could lead to the development of a discounted metric similar in principle to NDCG [34].

8.2.4 Deployment and application issues

Multi-language QTM

Among the aspects that need to be better understood before the deployment of QTM in industrial application contexts is the language sensitivity. In terms of language sensitivity we can say that most of the QTM process is essentially language independent, however, specific QTM components actually deal with text at the syntactic level and thus can be sensitive to human language features.

In particular, the processes that could exhibit greater language sensitivity are the candidate query generation and the query expansion process. In the case of candidate query generation, the most obvious issue to deal with is the Spotsigs strategy, which requires having a preconfigured stop-word list for each target language. However, the n-gram based query generation techniques could introduce language sensitivity in more subtle ways in languages that make intensive use of *declension*, *conjugation* or *agglutination* by making use of a greater number of variations of the same words. In those cases, we would be required to integrate stemming and de-compounding processes in the query generation phase in order to prevent that the evaluation and other downstream processes operate on each of the different query variations as if they were completely independent. In analogous way, the query expansion process, that requires the execution and retrieval of query results, should be modified to take into account the query filtering operations performed in the generation phase.

Semi-supervised QTM

Also, although the QTM framework is designed to build models in a completely unsupervised way, when dealing with industrial applications what matters the most is to obtain the best performance/cost ratio. Therefore, another line of future work is the development of a semi-supervised version of QTM that can be fed some training information provided that such training data can be gathered at a reasonable cost. In this regard, the main problem to solve would be how to incorporate training information into the QTM process. So, among the relevant works in relation to this problem we can point to Zhang et. al [71], where they proposed a co-clustering based knowledge-supervised learning algorithm or CoCKSL. The CoCSL algorithm uses Open Directory Project (ODP) as knowledge source for the document classification task. On the other hand, the work by Song et. al [57] explores the usage of human-provided category labels and automatically extracted named entities[1] to introduce "constraints" to the traditional word-document co-clustering, thus naming the approach Constrained Information-Theoretic Co-Clustering or CITCC. In CITCC, the constraints are introduced as a"must-link" condition for two documents if they share the same category labels.

So, based on the two previously presented approaches we can think of two possible enhancements to QTM in order to introduce low-cost training information.

[1] "A named entity is a collection of rigidly designated chunks of text that refer to exactly one or multiple identical, real or abstract concept instances. These instances can have several aliases and one name can refer to different instances", A. Nadeu 2008

- Enrich the original corpora with pre-labeled data, such as ODP or Wikipedia pages, and add must-link constraints to the training set. Still, some challenges would have to be addressed, such as which Wikipedia categories to include.
- Use named entities or other NLP supervised features as candidate query generation heuristics.

Machine learning task performance

Finally, let's recall that this research was motivated by the potential that semantic modeling offers to improve retrieval and other machine learning tasks, such as classification. A necessary step in that direction is the construction of effective retrieval models and classifiers based on the generated query-based topic models. During that process, it would be very valuable to assess the predictive power of the probabilistic metrics on the task performance, for which specific benchmarks usually exist. In this regard, it would be critical to explore and understand the relation of the semantic-coherence of the model and task specific performance.

8.3 Final remarks

In this work we have presented QTM, a large-scale, information-theoretic, topic modeling method that uses search queries and a simplified notion of topics to create semantic models of text collections. Also, we have presented a novel set of probabilistic metrics that not only were helpful to validate the results of QTM but also to bring some of the methodological advantages existing on other research communities to the topic modeling field. It is important to mention that most of the results of this dissertation have been peer-reviewed and published in international conferences. Among others, the main publications related to this work are: R. Brena and E. Ramirez [10]; E. Ramirez and R. Brena [49] and E. Ramirez, et. al [48].

We consider that this work is addressing relevant and open problems and proposing effective and innovative solutions. We hope the QTM framework to be of interest of industry practitioners working on large scale modeling problems and our work on metrics to be of help to anyone looking for an external and automated way to validate a topic model or other kinds of overlapping clustering algorithms.

Bibliography

[1] AIZAWA, A. An information-theoretic perspective of tf-idf measures. *Inf. Process. Manage. 39*, 1 (2003), 45–65. 17

[2] BAEZA-YATES, R., CASTILLO, C., JUNQUEIRA, F., PLACHOURAS, V., AND SILVESTRI, F. Challenges on distributed web retrieval. In *IEEE 23rd International Conference on Data Engineering, ICDE 2007* (2007). 3, 8

[3] BAEZA-YATES, R., AND RIBEIRO-NETO, B. *Modern Information Retrieval.* Addison-Wesley, 1999. 8

[4] BARROSO, L. A., DEAN, J., AND HÖLZLE, U. Web search for a planet: The google cluster architecture. *IEEE Micro 23*, 2 (2003), 22–28. 3, 8

[5] BERRY, M. W., DUMAIS, S. T., AND O'BRIEN, G. W. Using linear algebra for intelligent information retrieval. *SIAM Rev. 37*, 4 (1995), 573–595. 8, 17

[6] BLEI, D., GRI, T., JORDAN, M., AND TENENBAUM, J. Hierarchical topic models and the nested chinese restaurant process. *Advances in Neural Information Processing Systems 16* (2004). 10

[7] BLEI, D., AND LAFFERTY, J. A correlated topic model of science. 17–35. 2, 10

[8] BLEI, D., NG, A., AND JORDAN, M. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022. 2, 9

[9] BÖSE, J.-H., ANDRZEJAK, A., AND HÖGQVIST, M. Beyond online aggregation: parallel and incremental data mining with online map-reduce. In *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud* (New York, NY, USA, 2010), MDAC '10, ACM, pp. 3:1–3:6. 63

[10] BRENA, R., AND RAMIREZ, E. A soft semantic web. *Hot Topics in Web Systems and Technologies 0* (2006), 1–8. 66

[11] BRODER, A. A taxonomy of web search. *SIGIR Forum 36*, 2 (2002), 3–10. 1

[12] BRODER, A., FONTOURA, M., JOSIFOVSKI, V., AND RIEDEL, L. A semantic approach to contextual advertising. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2007), ACM, pp. 559–566. 2

[13] CHANG, J., BOYD-GRABER, J., GERRISH, S., WANG, C., AND BLEI, D. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems (NIPS)* (2009). 3, 35

[14] CHEN, W.-Y., CHU, J.-C., LUAN, J., BAI, H., WANG, Y., AND CHANG, E. Y. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th international conference on World wide web* (New York, NY, USA, 2009), WWW '09, ACM, pp. 681–690. 32

[15] CHU, H. *Information representation and retrieval in the digital age.* Information Today Inc, 2003. 6

[16] CIARAMITA, M., MURDOCK, V., AND PLACHOURAS, V. Semantic associations for contextual advertising. *Journal of Electronic Commerce Research 9*, 1 (2008), 1–15. 2

[17] CONDIE, T., CONWAY, N., ALVARO, P., HELLERSTEIN, J. M., GERTH, J., TALBOT, J., ELMELEEGY, K., AND SEARS, R. Online aggregation and continuous query support in mapreduce. In *Proceedings of the 2010 international conference on Management of data* (New York, NY, USA, 2010), SIGMOD '10, ACM, pp. 1115–1118. 63

[18] COVER, T. M., AND THOMAS, J. *Elements of Information Theory.* Wiley, 1991. 10, 20

[19] CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2002), ACM, pp. 299–306. 1

[20] DEERWESTER, S. C., DUMAIS, S. T., LANDAUER, T. K., FURNAS, G. W., AND HARSHMAN, R. A. Indexing by latent semantic analysis. *Journal of the American Society of Information Science 41*, 6 (1990), 391–407. 2, 8

[21] DENOEUD, L., AND GUÉNOCHE, A. Comparison of distance indices between partitions. *Studies in Classification, Data Analysis, and Knowledge Organization* (2006), 21–28. 3, 35

[22] DHILLON, I. S., MALLELA, S., AND MODHA, D. S. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2003), ACM, pp. 89–98. 3, 12, 29

[23] DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., DEERWESTER, S., AND HARSHMAN, R. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1988), ACM, pp. 281–285. 8

[24] FOWLKES, E. B., AND MALLOWS, C. L. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association 78*, 383 (1983), 553–569. 35

[25] FURNAS, G. W., LANDAUER, T. K., GOMEZ, L. M., AND DUMAIS, S. T. Statistical semantics: analysis of the potential performance of keyword information systems. 187–242. 8

[26] FURNAS, G. W., LANDAUER, T. K., GOMEZ, L. M., AND DUMAIS, S. T. The vocabulary problem in human-system communication. *Commun. ACM 30*, 11 (1987), 964–971. 1

[27] GIROLAMI, M., AND KABÁN, A. On an equivalence between plsi and lda. In *SIGIR* (2003), pp. 433–434. 9

[28] GRIFFITHS, T. L., AND STEYVERS, M. Finding scientific topics. *Proc Natl Acad Sci U S A 101 Suppl 1* (April 2004), 5228–5235. 10

[29] GRIFFITHS, T. L., STEYVERS, M., AND TENENBAUM, J. B. Topics in semantic representation. *Psychological Review 114* (2007), 211–244. 2

[30] GYÖNGYI, Z., AND GARCIA-MOLINA, H. Web spam taxonomy. In *1st International Workshop on Adversarial Information Retrieval on the Web, AIRWEB 2005* (2005). 8

[31] HELLERSTEIN, J. M., HAAS, P. J., AND WANG, H. J. Online aggregation. pp. 171–182. 63

[32] HOFMANN, T. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1999), ACM, pp. 50–57. 2, 9

[33] JANSEN, B. J., BOOTH, D. L., AND SPINK, A. Patterns of query reformulation during web searching. *J. Am. Soc. Inf. Sci. Technol. 60*, 7 (2009), 1358–1371. 1

[34] JÄRVELIN, K., AND KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst. 20*, 4 (2002), 422–446. 6, 65

[35] LI, R. M., KAPTEIN, R., HIEMSTRA, D., AND KAMPS, J. Exploring topic-based language models for effective web information retrieval. In *Proceedings of the Dutch-Belgian Information Retrieval Workshop (DIR 2008), Maastricht, the Netherlands* (Enschede, April 2008), E. Hoenkamp, M. de Cock, and V. Hoste, Eds., Neslia Paniculata, pp. 65–71. 2

[36] LI, W., AND MCCALLUM, A. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning* (New York, NY, USA, 2006), ACM, pp. 577–584. 10

[37] LIU, S., YU, C., AND MENG, W. Word sense disambiguation in queries. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management* (New York, NY, USA, 2005), ACM Press, pp. 525–532. 2

[38] LIU, Z., ZHANG, Y., CHANG, E. Y., AND SUN, M. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology, special issue on Large Scale Machine Learning* (2011). Software available at `http://code.google.com/p/plda`. 10, 32

[39] MANDALA, R., TOKUNAGA, T., AND TANAKA, H. The use of WordNet in information retrieval. In *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, S. Harabagiu, Ed. Association for Computational Linguistics, Somerset, New Jersey, 1998, pp. 31–37. 2

[40] MILLER, G. A., BECKWITH, R., FELLBAUM, C., GROSS, D., AND MILLER, K. J. Introduction to wordnet: An on-line lexical database*. *Int J Lexicography 3*, 4 (January 1990), 235–244. 2

[41] NEWMAN, D., ASUNCION, A., SMYTH, P., AND WELLING, M. Distributed inference for latent dirichlet allocation. In *Advances in Neural Information Processing Systems* (2007), vol. 20. 13

[42] PAPADIMITRIOU, S., AND SUN, J. Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 512–521. 3, 32

[43] PEDERSEN, T., PATWARDHAN, S., AND MICHELIZZI, J. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)* (2004). 2

[44] PEREIRA, F., TISHBY, N., AND LEE, L. Distributional clustering of english words. In *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics* (1993), pp. 183–190. 10, 11

[45] PORTEOUS, I., NEWMAN, D., IHLER, A., ASUNCION, A., SMYTH, P., AND WELLING, M. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2008), ACM, pp. 569–577. 13

[46] PUPPIN, D., AND SILVESTRI, F. The query-vector document model. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management* (New York, NY, USA, 2006), ACM, pp. 880–881. 3, 30

[47] PUPPIN, D., SILVESTRI, F., AND LAFORENZA, D. Query-driven document partitioning and collection selection. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems* (New York, NY, USA, 2006), ACM, p. 34. 3, 16

[48] RAMIREZ, E. H., BRENA, R., MAGATTI, D., AND STELLA, F. Probabilistic metrics for soft-clustering and topic model validation. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on 1* (2010), 406–412. 66

[49] RAMIREZ, E. H., AND BRENA, R. F. An information-theoretic approach for unsupervised topic mining in large text collections. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01* (Washington, DC, USA, 2009), WI-IAT '09, IEEE Computer Society, pp. 331–334. 66

[50] ROELLEKE, T., AND WANG, J. Tf-idf uncovered: a study of theories and probabilities. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2008), ACM, pp. 435–442. 17

[51] SALTON, G., WONG, A., AND YANG, C. S. A vector space model for automatic indexing. *Commun. ACM 18*, 11 (1975), 613–620. 7

[52] SHAN, H., AND BANERJEE, A. Bayesian co-clustering. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 530–539. 12, 63

[53] SISTA, S., SCHWARTZ, R., LEEK, T. R., AND MAKHOUL, J. An algorithm for unsupervised topic discovery from broadcast news stories. In *Proceedings of the second international conference on Human Language Technology Research* (San Francisco, CA, USA, 2002), Morgan Kaufmann Publishers Inc., pp. 110–114. 18

[54] SLONIM, N., FRIEDMAN, N., AND TISHBY, N. Unsupervised document classification using sequential information maximization. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2002), ACM, pp. 129–136. 11, 56

[55] SLONIM, N., AND TISHBY, N. Agglomerative information bottleneck. MIT Press, pp. 617–623. 11

[56] SLONIM, N., AND TISHBY, N. Document clustering using word clusters via the information bottleneck method. In *In ACM SIGIR 2000* (2000), ACM press, pp. 208–215. 11

[57] SONG, Y., PAN, S., LIU, S., WEI, F., ZHOU, M. X., AND QIAN, W. Constrained coclustering for textual documents. In *AAAI'10* (2010), pp. –1–1. 65

[58] SPARCK-JONES, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28* (1972), 11–21. 7, 17

[59] STEYVERS, M., AND GRIFFITHS, T. *Probabilistic Topic Models.* Lawrence Erlbaum Associates, 2007. 2, 17

[60] TAM, Y.-C., LANE, I., AND SCHULTZ, T. Bilingual lsa-based adaptation for statistical machine translation. *Machine Translation 21*, 4 (2007), 187–207. 2

[61] THEOBALD, M., SIDDHARTH, J., AND PAEPCKE, A. Spotsigs: robust and efficient near duplicate detection in large web collections. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2008), ACM, pp. 563–570. 28, 62

[62] TISHBY, N., PEREIRA, F. C., AND BIALEK, W. The information bottleneck method. pp. 368–377. 11, 56

[63] TOMÁS, D., AND VICEDO, J. Re-ranking passages with lsa in a question answering system. *Evaluation of Multilingual and Multi-modal Information Retrieval* (2010), 275–279. 2

[64] VOORHEES, E. M. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1993), ACM, pp. 171–180. 2

[65] VOORHEES, E. M. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)* (1994), W. B. Croft and C. J. van Rijsbergen, Eds., ACM/Springer, pp. 61–69. 2

[66] VOORHEES, E. M. Trec: Continuing information retrieval's tradition of experimentation. *Commun. ACM 50*, 11 (2007), 51–54. 6

[67] WALLACE, D. L. A method for comparing two hierarchical clusterings: Comment. *Journal of the American Statistical Association 78*, 383 (1983), 569–576. 35

[68] WANG, P., DOMENICONI, C., AND LASKEY, K. B. Latent dirichlet bayesian co-clustering. In *ECML PKDD '09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases* (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 522–537. 12, 63

[69] WEI, X., AND CROFT, W. B. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2006), ACM, pp. 178–185. 2, 10, 13

[70] WU, J., XIONG, H., AND CHEN, J. Adapting the right measures for k-means clustering. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2009), ACM, pp. 877–886. 3, 34, 35

[71] ZHANG, C., AND XING, D. Knowledge-supervised learning by co-clustering based approach. *Machine Learning and Applications, Fourth International Conference on 0* (2008), 773–776. 65