UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Informatica Sistemistica e Comunicazione
Dottorato di Ricerca in Informatica - XXIII Ciclo


# Graphical Models for Text Mining: Knowledge Extraction and Performance Estimation

**Davide Magatti**
Ph.D. Thesis

Supervisor: Prof. Fabio Antonio Stella
Tutor: Prof. Raimondo Schettini

Coordinator: Prof. Stefania Bandini

ANNO ACCADEMICO 2009/2010

# Acknowledgments

This thesis would not have been possible unless the constant support of my advisor Professor Fabio Stella. He gave me suggestions, advices and a constant incentive to learn and deeply understand how things works. Special thanks go to Professor Enrico Fagiuoli for its suggestions and its huge knowledge of probabilistic models.

A particular thanks to Siemens Corporate Technology Group, München: Dr. Volker Tresp, Dr. Markus Bundschus, Dr. Florian Steinke, Maximilian Nickel, Yi Huang. My internship has been a precious occasion to learn and discuss and to find new friends.

Special thanks go my colleagues at DISCo that shared the highs and lows of my Ph.D. In particular, i thank my laboratory colleague Elena Gatti for our discussion and friendship and all the other Ph.D students for the discussion and fun moments. I thanks all the administrative personnel at the department, without them a Ph.D student life is hell. Thanks to all the student who worked with me during this years, part of the work in this thesis has been developed thanks to their contribution.

It is a pleasure to thank those who made my Ph.D and thus my thesis possible: Docflow S.p.A. They gave me the possibility to get a sense of business world, their needs and to propose solutions.

A special thanks go to all who have tolerated my obsessions and complaints during this years: my family, my friends and Manoela.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The continuously increasing amount of text available on the WEB, news wires, forums and chat lines, business company intranets, personal computers, e-mails and elsewhere is overwhelming [38]. From 1990 to 2005 more than one billion people worldwide entered the middle class, get richer, become more literate and thus fueled the information market [109]. The effect of such an economic and social revolution, together with the improvements achieved by information and communication technologies, is called the *information explosion*.

Indeed, in the last five years the information created started to diverge from the storage capacity as reported by the International Data Corporation [41]. Data and information has gone from scarce to superabundant. While it is common opinion that this setting brings huge benefits it is also clear to everyone that it creates new challenges. In the next ten years the data available on the WEB will amount to forty times the current size [41]. The knowledge hidden in such a huge amount of data will heavily influence social behavior, political decisions, medicine and health care, company business models and strategies as well as financial investment opportunities.

The overwhelming amount of available un-structured data has transformed the information from useful to troublesome. Indeed, it is becoming increasingly clear that our recording and processing capabilities are growing much slower than the amount of generated data and information.

Search engines exacerbated this problem and although new paradigm of web-search are now being explored [6], they normally provide users with huge amount of un-structured results, which need to be pruned and organized to become useful and valuable.

In [51] the authors explain how *the unreasonable effectiveness of data* will be the pillar of the new WEB revolution. Their position originates from noticing that the biggest successes in natural language related machine learning have been statistical speech recognition and statistical machine translation. These tasks are much harder than document classification, but the availability of large training sets allows the algorithms to have higher performances with respect to other task like document classification, part-of-speech tagging, named-entity recognition, or natural language processing. The difficulty to obtain labeled corpora for such task, is the primary problem researchers have to face. The process of annotation of a corpus by human evaluator is difficult, slow and expensive. Human annotators are usually biased and thus it difficult and costly to obtain objective evaluations. The suggestion of the authors is to exploit the data available over the WEB rather than generating expensive annotated data.

It is increasingly recognized that useful semantic relationships can be automatically learned from the statistics of search queries and the corresponding results, as well as from the accumulated evidence of WEB-based text patterns and formatted tables [107]; in both cases no manually annotated data is required. A second lesson learnt from speech recognition and machine translation is that memorization offers a good strategy in the case where a lot of data is available. The statistical models used are based on huge databases of probabilities associated with *n-grams*, i.e. short sequences of words, which have been built by exploiting billions or trillions of examples. A third lesson is the following; all the experimental evidence from the last decade in machine learning suggests that throwing away rare events is almost always a bad idea. Indeed, much WEB data consists of individually rare but collectively frequent events. Finally, the authors observed that for many tasks, words and word combinations provide all the representational machinery we need to learn from text. [51] conclude their manuscript with the following recommendation *"Choose a representation that can use unsuper-*

*vised learning on un-labeled data, which is so much more plentiful than labeled data"*.

## Contributions

In this dissertation, we are mainly concerned with probabilistic graphical model for knowledge extraction and performance estimation. Text mining is a broad definition of a huge set of models, methods and algorithms that aim to distill information from textual data and discover valuable knowledge otherwise hidden.

In particular we are interested in novel, efficient and sound models for managing document repositories expressed in many different formats and offering to the users nuggets of information necessary to gain competitive advantage and rational decision making. It is commonly acknowledged that business companies spend a great deal of efforts in document management and organization with slightly sufficient results. Usually, users perceive this tools as a burden and thus the quality of their submission to the content manager lack of useful information. This dissertation presents a set of models that could improve information mining by relieving the user from boring duties and offering efficient ways to manage, classify, tag and retrieve documents.

Thus the contributions of this dissertation are: a model for automatic document tagging by means of topic extraction models and labeling algorithm; a probabilistic model for performance evaluation of topic models, and a combined model that exploits semantic information together with textual sources to offer efficient and informative way to retrieve informations. The chapters are organized as follows:

- Chapter 2 presents an overview of background elements used along the dissertation. In particular we will introduce Text mining and its main activities: document preprocessing, representation, classification and performances evaluation metrics. We will also introduce a set of similarity metrics useful for comparison of discrete sets and a common divergence used to compare probability distributions. We

will conclude the chapter with a brief description of semantic graphs and their representations.

- Chapter 3 presents the current state of the art for probabilistic graphical models and their applications to text mining problems. We will introduce topic extraction models describing their evolution in the last decade until recently proposed hierarchical topic models. We will also review different approaches for topic models evaluation and topic labeling.

- Chapter 4 describes a proposal for improving document management with probabilistic topic models. We will show a model assessment procedure for topic extraction, an algorithm for automatic labeling of topics that is able to exploit a user supplied taxonomy, and finally, we propose a multi-net Naive Bayes classifier that directly maps labeled topics in a learned model for document tagging. The experimental section will show the topic extraction procedure, the labeling process and the documents tagging according to a real world corpus. Main contributions offered in this chapter refer to [71,73,74].

- In Chapter 5 a novel approach to topic model evaluation is presented. The model interprets a topic model as a soft clustering procedure and compares the result with a given *gold standard*. In particular the proposed model offers a probabilistic interpretation of a known clustering evaluation index and extends such metric to deal with incomplete and overlapping partitions. Moreover two probabilistic metrics linked to the concept of precision and recall are presented. A montecarlo procedure to evaluate the proposed metrics is presented. The experimental section is devoted to the evaluation of a topic model, and shows the correctness of the montecarlo procedure. Main contributions offered in this chapter refer to: [95,96].

- In Chapter 6 we present a novel approach for exploiting textual sources linked to a semantic knowledge base. In particular we will present a model that is able to enrich the semantic graph with text documents and offers an effective and sound technique to query the

knowledge base. The proposed system integrates classical information retrieval tools with SPARQL query by means of a spreading activation algorithm. The experimental section will show how this model is able to retrieve facts not present in the semantic knowledge base, and entity related with such facts. Main contributions offered in this chapter refer to: [72]

- Chapter 7 summarizes the ideas proposed in the dissertation and points to directions of future works.

# Chapter 2

# Background Material

The chapter will define the boundaries in which collocate the contributions of this dissertation. The central theme of this work is Text Mining, that can be described as an ensemble of theories and techniques that aims to extract valuable knowledge from unstructured or semistructured text. In this chapter we will briefly introduce the Text Mining workflow in terms of preprocessing, document representation, document classification and performance estimation. Moreover we will give a description of similarity measures for comparing discrete sets of objects, like words lists, and a brief description of Kullback-Leibler divergence and its modification as distance metric. Finally, we will give a brief introduction to semantic graphs and representation methods.

## 2.1 Text Mining

Text mining [9,38], is an emerging research area which aims to solve the problem of information overload. Its typical tasks are: *text categorization, document clustering and organization, and information extraction*. Text mining exploits models and algorithms from machine learning, data mining, information retrieval and natural language processing to automatically extract knowledge from semi-structured and unstructured data. Among such methods, Support Vector Machines (SVMs) [57] have been shown to be effective to solve the text categorization problem. However, SVMs are en-

dowed by an implicit limitation: they rely on a set of labeled samples. This condition is extremely costly to be achieved and thus it is not easily met; whenever it is satisfied the sample labeling result cannot be guaranteed to be coherent. A lot of efforts have been oriented towards document clustering and organization: which do not require labeled samples and automatically group documents according to some similarity or distance measure. Classical algorithms like *K-means* [70] or *Nearest Neighbor* [35] represent documents as vectors in a metric space and compute pair-wise distance to generate documents clusters. Hierarchical clustering algorithms have also been described in the specialized literature [35, 58]. This class of algorithms returns a nested sequence of partitions in which higher clusters contain a set of similar documents obtained from couples of more specific partitions and recursively repeat this idea until the partitions consist of a single document. Drawbacks of classical clustering methods consist in the inability to capture the meaning of the different parts of the documents, forcing them inside a determined bin or subset of bins. Another problem is the labeling of clusters that often relies on humans. [64] proposed Latent Semantic Analysis as an efficient approach to document clustering and organization; the model has been later enriched with a probabilistic framework based on mixture decomposition via a latent class model, by [54]. These models capture the concept contained in a document and identified by set of related words [16].

### 2.1.1 Text Mining Workflow

#### Preprocessing

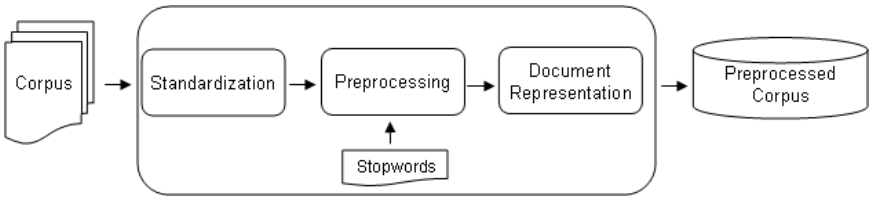



**Figure 2.1:** Preprocessing.

Given a textual source containing different types of documents (different formats, language registers, formatting) the first action that should be

taken is standardization. The *standardization* step consists in the conversion of the data in a common, shared and machine readable format. In the specialized literature the common destination formats are identified with eXtensible Markup Language (XML) [106] that allows a structured representation of the documents (i.e. it is possible to identify different sections in a document, like title, abstract, chapters, sections etc...) or TXT (ASCII or Unicode) that is preferred whenever the system needs a plain and straight format. Thus, the system should be able to deal with several formats (e.g. PDF, DOC, RTF, HTML) and generate the *corpus* as a set of TXT or XML documents.

Once the corpus has been generated, the successive step is the *document preprocessing*, in which different filters are applied to remove all the data that in the specific implementation are considered to be non informative. Hence the system applies a tokenization algorithm that identifies each word, removing the punctuation, numeric values and other standardization debris. Then, each token is matched against a stop-word list, containing articles, pronouns, common abbreviations that are usually considered to be non informative. The list is language dependent, and whenever the system has to deal with multi-language corpora, different stop lists should be applied. If the size of the corpus is huge, the dimension of the relative vocabulary (i.e. the set of all the words appearing in the corpus) can become quite large. One possible solution to reduce the number of words and to speed up the computation without loss of information is the removal of too frequent words that are normally language or corpus specific and of less frequent words that usually are typos or some other kind of errors. A possible approach consists in the computation of the distribution of the words sorted by frequency that usually follows a *zipf law* [78]. Then, two quantiles identifying the upper and lower tails are selected. Thus, the vocabulary is reduced and it will contain the words with frequency within a range defined by quantiles.

**Document Representation** The last component of the text preprocessing module is the *document representation.* Once the documents have been transformed and all the useless information are filtered out, the system should

transform the unstructured text in a highly structured format.

The documents are usually associated with a document vector $\mathbf{w}$. It is a $V$-dimensional vector where $V$ represents the cardinality of the vocabulary of the document corpus. However, the document vector can be represented in different ways, while the representation used plays a central role on the generalization ability achieved by the learning algorithm. The main representation schemes are the following; binary or $0/1$, term frequency and term frequency inverse document frequency. In *binary representation* each document is associated with a binary vector $\mathbf{w}$ whose $i^{th}$ component equals $1$, whether the document contains at least one instance of the $i^{th}$ word of the vocabulary and $0$ otherwise. The *term frequency* representation counts the occurrence of each word of the vocabulary. Therefore, the *ith* component of the vector $\mathbf{w}$ contains the number of occurrences of the *ith* word in the considered document. The last representation scheme, namely the *term frequency inverse document frequency* [101], consists of two components; the term frequency and the inverse document frequency. The *term frequency* component for the $i^{th}$ word $tf\,(i)$ is the same as in the term frequency representation. The *inverse document frequency* component of the $i^{th}$ word is the reciprocal of the number of document $df\,(i)$ where it occurs. Thus, the Term Frequency Inverse Document Frequency (TF-IDF) for the $i^{th}$ word is defined as follows:

$$tf - idf\,(i) = tf\,(i) \cdot \log\left(\frac{1}{df\,(i)}\right). \tag{2.1.1}$$

It is customary to normalize term frequency inverse document frequency to account for different document lengths.

Standard learning methods for text classification are; Naïve Bayes, Rocchio, K-nearest neighbors and decision tree. However, it has been recognized that Support Vector Machines (SVMs) are state of the art to solve the text classification problem [75].

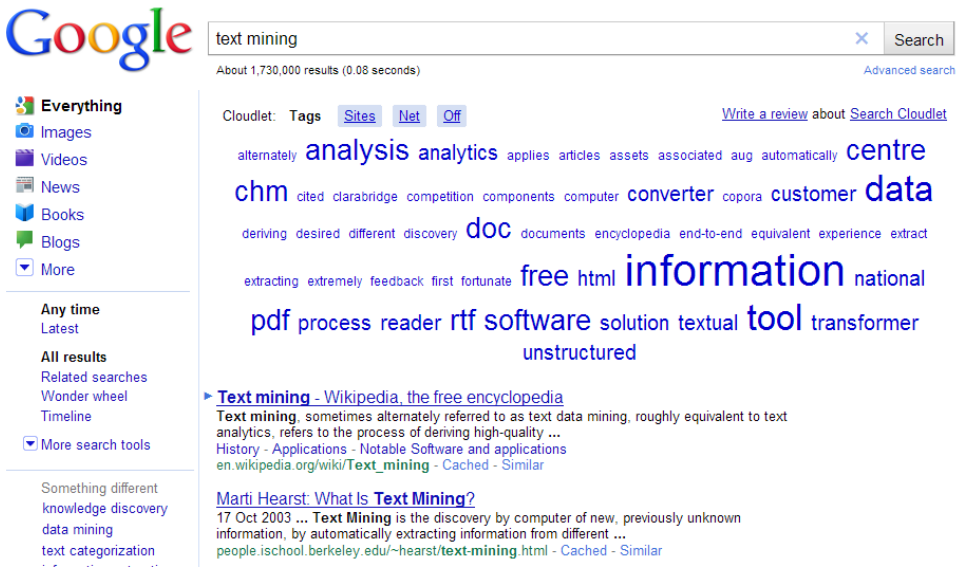



**Figure 2.2:** Example of Google results Tag Cloud

### Text Classification

The main scenarios in text classification are binary, multi-class and multi-label [57]. The *binary* scenario is the simplest and consists of learning a supervised classifier from two classes data; *multi-class* is the straightforward generalization of binary classification. It consists of documents coming from more than two classes. However, most of the text classification problems belong to the *multi-label* scenario. In such a scenario, there is no one-to-one correspondence between class and document. Given a fixed number $L$ of classes, each document can be in multiple, exactly one or no class at all. Classes are usually semantic topic identifiers used to tag documents, newswires, web pages, ... (Figure 2.2).

The multi-label text classification setting is modeled with an $L$-dimensional class vector $\underline{y}$ where each component can take value on $\{-1, +1\}$. Formally, the class vector is defined as follows:

$$\underline{y} = \{-1, +1\}^L . \tag{2.1.2}$$

[57] pointed out that in the multi-label setting it is not clear how clas-

sification errors have to be counted. Indeed, the $0/1$ loss function does not allow to model *close misses* while a reasonable distance metric is offered by the Hamming distance which counts the number of mismatches between the *class vector* $\underline{y}$ and the *classifier output* $\underline{\hat{y}}$. Therefore, whether the Hamming distance is used, the expected loss equals the sum of the error rates of $L$ binary text classification tasks. This means that the multi-label text classification task can be conveniently split into $L$ binary classification tasks. [57] motivated this approach by making the assumption that classes are independent, given the *document vector* $\mathbf{w}$ and thus using a Bayes argument concerning the optimality of the maximum posterior classification rule.

**Performance Evaluation measures**

Methods and algorithms for binary text classification are usually compared on the basis of the following performance measures; accuracy, precision and recall. In such a setting each document is labeled as a *positive* sample (+1) or a *negative* sample (-1). Let $y$ and $\hat{y}$ be respectively the true label and classifier forecasted label for the document described by the vector $\underline{x}$. It is customary to define as ***T**rue **P**ositive* (***T**rue **N**egative*) those documents where $y = \hat{y} = +1$ ($y = \hat{y} = -1$), while a document such that $y = +1$ and $\hat{y} = -1$ ($y = -1$ and $\hat{y} = +1$) is said to be a ***F**alse **N**egative* (***F**alse **P**ositive*). Then, given a document corpus consisting of $N$ elements the *accuracy* of a classifier algorithm or method for binary text classification is defined as follows:

$$accuracy = \frac{TP + TN}{N} \tag{2.1.3}$$

and measures the effectiveness of the classifier, i.e. its capability to provide reliable forecasts. *Precision* is defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{2.1.4}$$

and measures the reliability of the classifier to provide correct forecasts for the positive class. The *recall* measure is defined as follows:

$$recall = \frac{TP}{TP + FN} \tag{2.1.5}$$

and measures the fraction of true positive documents which are recalled from the classifier.

Finally, *F-measure* has been proposed as a single metric performance that incorporate the value of precision and recall by computing the harmonic mean as follows:

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.1.6}$$

## 2.2 Similarity Measures

A large number of similarity measures or pseudometrics have been presented in literature. The measures of interest to this dissertation should satisfy two distinct requisites: on one hand we are interested to measures that are able to compare sets of discrete objects like word lists, on the other hand we want to compute the similarity of given probability distributions.

### 2.2.1 Set Similarities

The comparison of discrete sets of object and in particular words lists can be performed with many different metrics. In this section we present a set of measures that can be applied for the comparison of sets of different cardinality. [48] presented, analyzed and compared a representative set of similarity measures that can be exploited to evaluate *concepts* similarity, where a concept is defined as a semantically related sets of words. In this section we present a subset of similarities that shows a coherent behavior applied to words sets comparison: namely cosine similarity, overlap similarity, mutual similarity, dice similarity, Tanimoto and Jaccard similarities.

Let $\underline{x}$ and $\underline{y}$ be two vectors while $\|\underline{x}\|$ be the Euclidean norm of vector $\underline{x}$, then the *cosine similarity* between vector $\underline{x}$ and vector $\underline{y}$ is defined as follows:

$$Cosine\left(\underline{x}, \underline{y}\right) := \frac{\underline{x}\underline{y}^T}{\|\underline{x}\| \cdot \|\underline{y}\|}. \tag{2.2.1}$$

It measures the similarity of the argument vectors through the cosine of the angle between them. The smaller the angle the greater the similarity between the argument vectors is. It is worthwhile to mention that vectors $\underline{x}$ and $\underline{y}$ are binary representations respectively for set $A$ and $B$. The dimensionality of $\underline{x}$ and $\underline{y}$ equals the cardinality of the union set $A \bigcup B$.

The *overlap similarity* measure is defined as follows:

$$Overlap\,(A, B) := \frac{|A \bigcap B|}{\min\,(|A|\,, |B|)} \tag{2.2.2}$$

where $A$ and $B$ are sets, while $|A|$ represents the cardinality of the set $A$.

The *mutual similarity* uses the degree of inclusion of set $A$ into set $B$ and the degree of inclusion of set $B$ into set $A$. It computes their average value as follows:

$$Mutual\,(A, B) := \frac{\frac{|A \bigcap B|}{|A|} + \frac{|A \bigcap B|}{|B|}}{2}. \tag{2.2.3}$$

The *dice similarity* is defined as follows:

$$Dice\,(A, B) := \frac{2 \cdot |A \bigcap B|}{|A| + |B|}, \tag{2.2.4}$$

and is related to the Jaccard coefficient, commonly used in information retrieval to measure the overlap between two sets.

The *Jaccard coefficient* is defined as follows:

$$Jaccard\,(A, B) := \frac{|A \bigcap B|}{|A \bigcup B|}. \tag{2.2.5}$$

It ranges from zero to one as the cosine similarity measure. Finally, the *Tanimoto distance*, commonly used to compute the similarity between sets with different cardinalities is defined as follows:

$$Tanimoto\,(A, B) := 1 - \frac{|A| + |B| - 2 \cdot |A \bigcap B|}{|A| + |B| - |A \bigcap B|}. \tag{2.2.6}$$

### 2.2.2 Probability distribution similarity

*Kullback-Leibler divergence* (*KL*) [69] is a similarity function commonly used to compare different probability distributions with the same support $S$ and is defined as follows:

$$KL(t_i, t_j) = \sum_{k=1}^{W} t_i^k \log\left(\frac{t_i^k}{t_j^k}\right) \tag{2.2.7}$$

where $t_i^k$ and $t_j^k$ are the probability $P_i(k)$ and $P_j(k)$. The quantity is equal to zero when for all $k$, $t_i^k = t_j^k$.

KL is not a proper metric due to its asymmetry. Thus, a symmetrized version has been proposed, which simply compute the average of both the divergences:

$$sKL(t_i, t_j) = \frac{1}{2}\left(KL(t_i, t_j) + KL(t_j, t_i)\right). \tag{2.2.8}$$

## 2.3 Semantic Repositories

In the last decades, the interest in efficient and sound tools for information sharing and knowledge management has steadily grown. Since 1972, ontologies have been introduced by computer scientists as a possible solution for knowledge engineering and representation, information integration, language modeling, database design. For each of this applications a specific definition of the term ontology has been given: for example it has been interpreted as a tool for domain modeling or as a way to solve lexical and semantic problems like sinonimity. In recent years also the Machine Learning and Text Mining communities has started to investigate possible connections [7, 37, 111] and applications.

While in [49, 50] the authors has discussed the notion of ontology and its implication and applicability, in [8] the authors has moved towards the definition of the Semantic Web by means of a set of properties that should be satisfied by a (web) resource:

> *The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and*

*people to work in cooperation. [8].*

In the last ten years, a great deal of effort has been spent in the definition of ontology languages that satisfy the above definition. A first step towards the creation of a structured information over the web was XML [106] that soon became too limited for richer ontology specification. Subsequently RDF and its accompanying schema language RDFs have became reference language for ontology definition and construction. In the recent years Ontology Web Language (OWL) has been proposed as an extension of RDF.

**Resource Description Framework**

RDF (Resource Description Framework) is a general purpose language for representing metadata over the World Wide Web. It guarantees metadata interoperability. The data model is characterized by triples consisting of $(object, properties, values)$. W3C has provided a standard syntax specification [65] and an associated schema specification [17]. The model consists of three basic data types:

- *Resources* identify objects named by Uniform Resource Identifier (URI) or by strings.
- *Properties* define a specific characteristic, attribute or relation to describe a resource.
- *Statements* are specific resources together with a named property and an associated value.

Formally each RDF statement is defined by a triple: $< S, P, O >$ where $P$ (Predicate) is an URI, $S$ (Subject) is an URI or a blank node and $O$ (Object) is a URI, a blank node or a literal. RDF offers a very basic syntax that defines web metadata like authors, creation date etc. RDFs extended such syntax making possible the definition of classes of resources and properties. In particular, according to RDFs syntax, an ontology is defined in terms of classes, subclasses, sub-properties, domain and range restriction of properties. A complete specification of RDF/RDFs *vocabulary description language* is provided by the W3C [17].

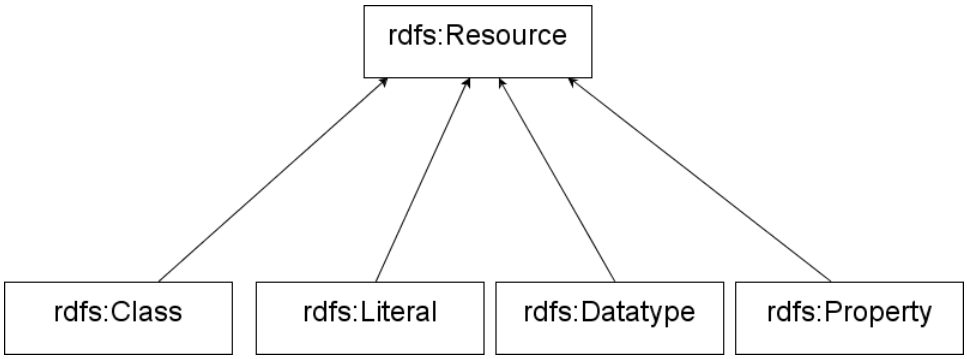



**Figure 2.3:** Classes definition in RDFs.

The basic elements of the RDFs are *classes* and *properties*: a class has the same interpretation of a class in an object-oriented programming language. The types of a class, depicted in Figure 2.3, are the following:

- *rdfs:Resource:* All things described by RDF are are instances of the class $rdfs : Resource$.
- *rdfs:Class:* The class of resources that are RDF classes. $rdfs : Class$ is an instance of $rdfs : Class$.
- *rdfs:Literal:* The class of XML literal values. $rdf : XMLLiteral$ is an instance of $rdfs : Datatype$ and a subclass of $rdfs : Literal$.
- *rdfs:DataType:* The class of datatypes. All instances of $rdfs : Datatype$ correspond to the RDF model of a datatype described in the RDF Concepts specification. $rdfs : Datatype$ is both an instance of and a subclass of $rdfs : Class$. Each instance of $rdfs : Datatype$ is a subclass of $rdfs : Literal$.
- *rdfs:Property:* is the class of RDF properties. $rdf : Property$ is an instance of $rdfs : Class$.

RDFs specifications for properties, depicted in Figure 2.4, are the following:

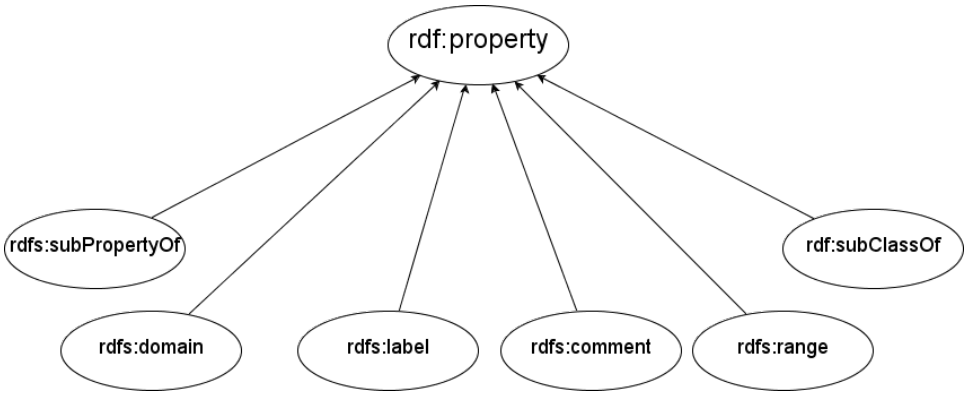



**Figure 2.4:** Properties specification in RDFs

- *rdfs:subClassOf:* The property rdfs:subClassOf is an instance of rdf:Property that is used to state that all the instances of one class are instances of another. The $rdfs : subClassOf$ property is transitive.
- *rdfs:subPropertyOf:* The property $rdfs : subPropertyOf$ is an instance of $rdf : Property$ that is used to state that all resources related by one property are also related by another. The $rdfs : subPropertyOf$ property is transitive.
- *rdfs:range:* is an instance of $rdf : Property$ that is used to state that the values of a property are instances of one or more classes.
- *rdfs:domain:* is an instance of $rdf : Property$ that is used to state that any resource that has a given property is an instance of one or more classes.
- *rdfs:label:* is an instance of $rdf : Property$ that may be used to provide a human-readable version of a resource name.
- *rdfs:comment:* is an instance of $rdf : Property$ that may be used to provide a human-readable description of a resource.

# Chapter 3

# State of the art

## 3.1 Probabilistic Graphical Models

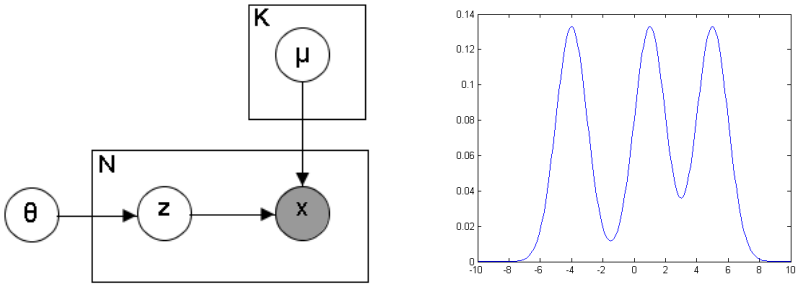
Statistical applications in Data and Text Mining require to deal with complex models involving thousand of interdependent random variables. Probabilistic Graphical models tries to merge two important fields of applied mathematic: probability theory and graph theory and offer an efficient and sound tool for solving inference and estimation problems.

A probabilistic graphical model (PGM) [59] is a family of probability distribution described by a directed or undirected graph. Undirected models commonly describes *Markov Random Fields* [63], while directed models are commonly used for the description of Bayesian Networks [87] or latent variable models.

Formally a PGM is defined as $G = (V, E)$ where each node $v \in V$ represents a random variable, each edge $\{(v_i, v_j) \in E\}$ describe dependencies among the variables and $plates$ describe replication of substructures of the graph.

PGM can be used to represent *latent variable models* that describe how observed data interact with *latent or unobserved* random variables. In the graphical representation, shaded nodes represent observed variables and unshaded ones the unobserved variables.

**Example 1.** *In Figure 3.1(a) is represented the graphical model of a Gaussian mixture model; the observed variable $x$ is shaded, the unobserved variable $z$ represents*




(a) Graphical Model. (b) Probability Density Function.

**Figure 3.1:** Gaussian Mixture.

*a sample from a multinomial distribution $\theta$ that selects the normal distribution from which the next value will be sampled and $\mu$ represents the random variable of the mean parameter of the Gaussian distributions composing the mixture.*

*In figure 3.1(b) is represented the probability density function of the mixture composed by three different normal distribution distribution with mean $\mu_1 = -4, \mu_2 = 1, \mu_3 = 5$ and equal variance $\sigma_1 = \sigma_2 = \sigma_3 = 1$. Posterior inference on such model is the main task we want to solve. In particular we may be interested in computing the value of $z$ given a data point $x$ i.e. $P(z|x = 1, \mu_1, \mu_2, \mu_3)$, or to estimate the value of the mean parameter $\mu_k$ for each component.*

Posterior inference on the desired latent variables conditional on observed data can be computed by *empirical Bayes* approaches which find the point estimates of the parameters based on maximum likelihood using, for example, the *expectation maximization* algorithm [31]. On the other hand, the problem can be solved using a fully Bayesian approach, placing prior on variables and computing proper posterior distribution over the model parameters. Such approach is defined as the *hierarchical Bayesian modeling* [44] and necessitate the specification of a distribution over the parameters, which are endowed by hyper-parameters.

### 3.1.1 Posterior inference

In latent variable models our goal is to compute the posterior distribution of latent variables conditioned on observed data. Exact computations

for this posterior is usually intractable except for toy models. In recent years with the increase of computational power and models complexity, several approximate approaches have been proposed in the literature. In this dissertation we are mainly concerned with *Mean field variational approach* [59] and *Gibbs Sampling* [24] that are the natural choice for the models presented in the following section in which variables are assumed to follow *exponential distributions* [12].

In a hierarchical mixture model with parameter $\eta$, observed variables $\mathbf{x} = x_{1:N}$ and latent variable $\mathbf{z} = z_{1:M}$ the posterior of the latent variable can be computed as:

$$p(z_{1:M}|x_{1:N}, \eta) = \frac{p(x_{1:N}, z_{1:M}|\eta)}{\int p(x_{1:N}, z_{1:M}|\eta) dz_{1:M}}.$$

This posterior is often intractable. Whenever in the denominator $z_{1:M}$ is a realization of one of K possible values (i.e. the mixture components), the integral is a sum over $K^M$ elements.

**Mean-Field Variational methods**

Let $(\mathbf{y}, \mathbf{z}) = (y_1, ..., y_n, z_1, ..., z_d)$ be a continuous random vector with values in $\mathbb{R}^{n+d}$ and for a given $\theta$ define the joint density of $(y, z)$ by $f(y, z|\theta)$. Suppose $y$ to be observed while $z$ is latent; the parameter $\theta$ is modeled with distribution $p(\theta)$. The posterior $p(\theta|y)$ can be defined as:

$$p(\theta|y) = \frac{\int g(y, z, \theta) dz}{m(y)} \tag{3.1.1}$$

where $g(y, z, \theta)$ is the joint density of $(y, z, \theta)$ and $m(y)$ the marginalized density. $p(\theta|y)$ is normally intractable in many real world problems, mean-field variational methods offer a deterministic methodology for approximating such posteriors. The posterior $p(\theta|y)$ can be computed by marginalizing out $z$ from $p(\theta, z|y)$. The variational approximation can be defined as follows:

- Let $q(\theta, z|y)$ to denote the variational density with same support $S$ of the true distribution.

- Assume that $\theta$ and $z$ are conditionally independent given $y$: $q(\theta, z|y) = q_1(z|y)q_2(\theta|y)$.
- Choose $q_1(z|y)$ and $q_2(\theta|y)$ to minimize the Kullback-Leibler divergence between $p(\theta, z|y)$ and $q_1(z|y)q_2(\theta|y)$.

The quantity to be minimized can be computed to be:

$$KL(p(\theta, z|y), q(z, \theta|y)) = log(p(y)) + \int_S q_1(z|y)q_2(\theta|y)log(\frac{q_1(z|y)q_2(\theta|y)}{g(y, z, \theta)})dzd\theta \tag{3.1.2}$$

The minimization must satisfy the following constraints:

- $q_1(z|y)$ and $q_2(\theta|y)$ are strictly positive
- $\int q_1(z|y)dz = 1$ and $\int q_2(\theta|y)d\theta = 1$

The minimization is solved through a mean-field algorithm that iteratively minimize the KL with respect to the unconstrained $q_1$ and $q_2$. In particular at each step the algorithm separately minimize $q_1$ and $q_2$ while holding the other fixed.

Further details on variational models can be found in [12, 60, 113, 118].

**Gibbs Sampling**

Gibbs Sampler is a particular type of *Markov Chain Monte Carlo (MCMC)*. MCMC are a class of algorithm in which the rationale is to build a Markov Chain which stationary distribution is the target distribution. Once the chain is collected it is possible to collect samples from the chain and approximate the desired distribution. In particular GS is derived from Metropolis-Hasting algorithm [52] in which the acceptance criterion for the candidate point is removed. The key of GS is to consider only univariate conditional distributions, hence all the random variables but one have assigned a fixed value. This approach implies that the vector of $n$ different random variables is computed in $n$ steps rather than generating the vector in a single

pass. The samples are collected after a **burn in** period that makes the distributions independent to the starting configuration.

To compute the posterior $p(\mathbf{z}|\mathbf{x}, \eta)$ each iteration of the GS draws each latent variable $z_i$ from $p(z_i|\mathbf{z}_{-i}, \mathbf{x}, \eta)$. When the chain is converged we collect $B$ samples and approximate the empirical distribution as follows:

$$p(\mathbf{z}|\mathbf{x}, \eta) = \frac{1}{B}\sum_{b=1}^{B} f(\mathbf{z}_b).$$

An extended description of the Gibbs Sampler can be found in [24,43,44]

## 3.2 Probabilistic Graphical Models for Text Mining

The following section describes the main models proposed in the literature for topic extraction. In particular we will describe the basic models proposed in the last decades together with their extensions.

### 3.2.1 Notation

The notation used in the dissertation follows the following schema:

- A corpus of documents is identified by a collection of $M$ documents: $\mathcal{D} = (d_1, d_2, \cdots, d_M)$.
- Each document $d_i$ is represented by a vector of $N$ words $\mathbf{w} = (w_1, w_2, \cdots, w_N)$.
- Each corpus is associated with a vocabulary $V = \bigcup_{d\in\mathcal{D}} \mathbf{w}_d$
- In each document $d_i$, each word $w$ is associated with a topic, thus each word vector $\mathbf{w} = (w_1, w_2, \cdots, w_N)$ is associated with a topic vector $\mathbf{z} = (z_1, z_2, \cdots, z_N)$.

### 3.2.2 Uni-gram model

The simplest latent graphical model for text can be identified with unigram model. The generative model described in Figure 3.2 assume that

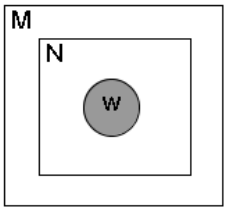



**Figure 3.2:** Uni-gram Model.

for each document $d \in \{1, M\}$ the words $w \in \{1, N\}$ are sampled from a single multinomial distribution. The probability of a document is defined as follows:

$$P(\mathbf{w}) = \prod_{n=1}^{N} P(w_n). \tag{3.2.1}$$

### 3.2.3 Mixture of Uni-grams

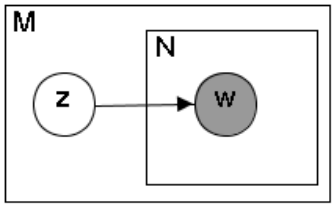



**Figure 3.3:** Mixture of Uni-grams Model.

[90] has extended the uni-gram model by including a latent topic variable $z$. The underlying generative model states that given a document we first choose a topic $z$ and then we sample $N$ words independently from the conditional distribution $p(w|z)$. The document probability is defined as follows:

$$P(\mathbf{w}) = \sum_{z} P(z) \prod_{n=1}^{N} P(w_n|z). \tag{3.2.2}$$

Hence, this representation states that each document is associated with only one possible topic. This assumption is limiting especially in large text collection where a more fine grained approach should better represents the complexity. Inference in conducted with an EM procedure.

### 3.2.4 Probabilistic Latent Semantic Analysis (pLSA) [54]

pLSA have been proposed by [54, 55] to give a statistical interpretation of Latent Semantic Analysis (LSA) [64]. LSA has its roots in linear algebra and, in particular, in dimensionality reduction algorithms. Given the *bag-of-words* representation of a corpus, i.e. a $\mathcal{D} \times \mathcal{W}$ term frequency matrix, Latent Semantic Analysis applies a *Single Value Decomposition (SVD)* to project the document in a new lower dimensional space that allows to better capture similarities between documents (LSA) and between documents and queries (Latent Semantic Indexing). One of the deficits of such model contested by [55] is the lack of solid statistical foundation and the difficulty of dealing with polisemy and synonymity.

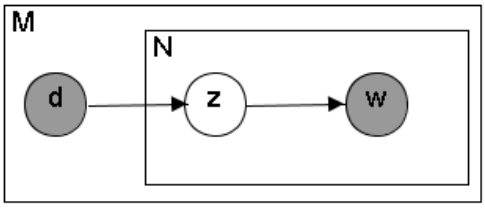



**Figure 3.4:** pLSA model.

The proposed model, tries to overcome the problem of LSA by formulating a latent class model specifying a precise generative model and offering an *empirical Bayes* algorithm for posterior inference.
In particular the proposed model tries to associate the co-occurrences of the words $w_i \in \mathbf{w}$ with a latent class $z_i \in \mathbf{z}$ for each document $d \in \mathcal{D}$. The graphical model is represented in Figure 3.4 and the relative generative model is described as follows:

1. select a document $d$ with probability $P(d) \propto length(d)$,
2. pick a latent class $z$ with probability $P(z|d)$,
3. generate a word $w$ with probability $P(w|z)$.

The observed variables are the pairs $(d, w)$, i.e. the presence of a word inside a document. The joint probability model can be written as:

$$P(w,d) = P(d) * P(w|d), \text{ where} \tag{3.2.3}$$

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z) * P(z|d). \tag{3.2.4}$$

The model makes two assumptions: the observation pairs $(d, w)$ are assumed to be independent (i.e. *bag-of-words assumption*), and it is assumed that the words $w$ are generated independently of the document $d$ given the latent class $z$.

**Inference** This model interprets documents as mixtures of multinomial distributions over words given the latent class variable $p(w|z)$ where the mixing proportion are given by $p(z|d)$. It is worthwhile to notice that *pLSA* is different form other *document clustering* algorithms, as a matter of fact, documents are described as mixtures of objects that are distribution over words.

The Log-Likelihood of the model is defined as:

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w) log P(d,w) \tag{3.2.5}$$

where $n(d, w)$ is the *term frequency* of the word $w$ in the document $d$. Maximum Likelihood estimation can be computed with the Expectation Maximization algorithm.
The $E$ step computes posterior probabilities for the latent variable $z$ based on current values of the parameters:

$$P(z|w,d) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}.$$

The maximization ($M$) step, updates parameters for the computed pos-

terior:

$$P(w|z) = \frac{\sum_d n(d,w)P(z|w,d)}{\sum_{d,w_i} n(d,w_i)P(z|w_i,d)}, \tag{3.2.6}$$

$$P(d|z) = \frac{\sum_d n(d,w)P(z|w,d)}{\sum_{d^j,w} n(d_j,w)P(z|w,d_j)}, \tag{3.2.7}$$

$$P(z) = \frac{1}{\sum_{d,w} n(d,w)} \sum_{d,w} n(d,w)P(z|d,w). \tag{3.2.8}$$

The author states that the EM algorithm [31] is prone to overfitting due to the numbers of parameters to evaluate: in particular the model consist of K multinomial distribution over V words plus the K multinomial mixture over each of the M documents. To overcome the overfitting author proposes a *Tempered* EM algorithm TEM, that at each iteration applies a smoothing over the parameters computation. Details of the TEM algorithm are described in [54].

### 3.2.5 Latent Dirichlet Allocation (LDA) [16]

LDA tries to improve the weakness of previous models. In particular the *mixture of uni-gram* model is heavily limited by assigning one topic per document, while *pLSA* suffers of overfitting due to the non well defined generative model. In pLSA $d$ is a dummy variable representing the index of the document currently considered in the corpus, and thus the associated random variable has as many possible values as the number of documents: so the model learns the topic mixture $p(z|d)$ considering only the documents in the current corpus. This particular set-up implies that there is no natural way to assign probability to a previously unseen document. Moreover, the number of parameters of the model grows linearly with the number of documents, due to fact that the distribution is indexed by the documents: e.g. in a $K$-topic model, the number of parameters is specified by $K$ multinomial over the vocabulary of size $V$ plus a mixture over $K$ topic for each of the $M$ documents resulting in $kV + kM$ parameters to estimate that grows linearly in the number of documents. [94] have shown that also the tempered version of the EM algorithm is prone to overfitting.

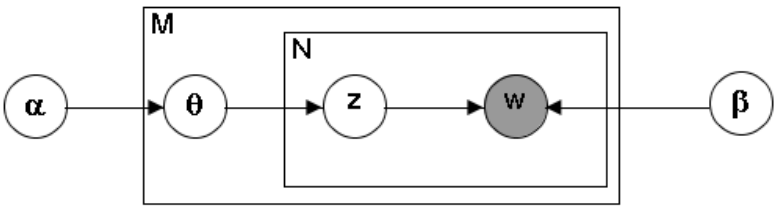



**Figure 3.5:** LDA model.

In LDA a document is formed by first deciding its number of words, the mixing coefficients associated with the topics and then by repeatedly choosing a topic and consequently sampling a word from the selected distribution. Thus the generative process proposed by LDA gives a fully probabilistic definition of document generation process:

1. choose the document length $N \sim \text{Poisson}(\xi)$,
2. choose the topic mixing proportion $\theta \sim \text{Dir}(\alpha)$,
3. For each of the N words $w_n$:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$,
   (b) Choose a word $w_n \sim p(w|z_n, \beta)$.

The model has some simplifying assumptions: the number of topics and thus the dimensionality of the Dirichlet distribution is assumed to be known. The Poisson assumption can be ignored: the number of words $N$ in the document is independent of the data generating variables $\theta$ and **z**, hence it can be interpreted as an ancillary variable and so ignored. Moreover, $\beta$ is interpreted as a $K \times V$ matrix such that $\beta_{ij} = P(w^j = 1|z^i = 1)$ is treated as fixed quantity to be estimated. Given the parameters $\alpha$ and $\beta$, it is possible to write the joint distribution of a topic mixture $\theta$, a set of words **w** and its associated topic **z** as:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta). \tag{3.2.9}$$

The marginal distribution of a document is then computed by integrating over $\theta$ and summing over $z$:

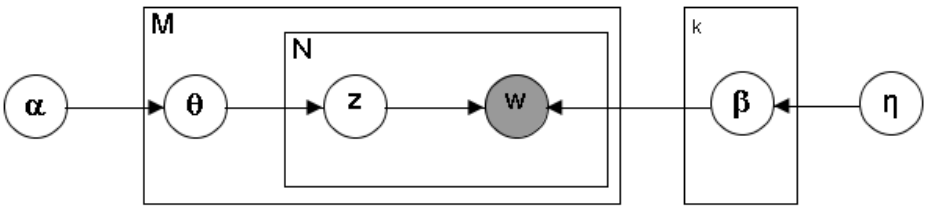



**Figure 3.6:** Smoothed LDA model.

$$p(\mathbf{w}|\alpha,\beta) = \int p(\theta|\alpha)(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta))d\theta. \tag{3.2.10}$$

**Inference** Given a LDA model, we are mainly interested in computing the posterior of the hidden variables given a document:

$$p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta) = \frac{p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta)}{p(\mathbf{w}|\alpha,\beta)}. \tag{3.2.11}$$

Unfortunately, this distribution is intractable to compute in general, due to the nature of the denominator (see Equation 3.2.10) in which $\theta$ and $\beta$ are coupled. The authors propose a convexity based variational inference approach for posterior approximation. We address the reader to the original paper for the details.

The parameter $\alpha$ and $\beta$ are estimated via a variational EM algorithm that maximize the Log-Likelihood of the data:

$$\mathcal{L}(\alpha,\beta) = \sum_{d=1}^{M} logp(\mathbf{w}_d|\alpha,\beta).$$

The authors state that in large corpora the evaluation of $\beta$ in the inference step can be problematic: when the size of the vocabulary tends to grow large the matrix $\beta$ with dimensions $K \times V$ becomes really sparse. In this set-up the model tends to associate zero probability to previously unseen words and thus giving zero probability to new documents. The authors suggest to insert a Dirichlet smoothing over $\beta$ with parameter $\eta$. The proposed model is depicted in Figure 3.6. The inference procedure is adapted

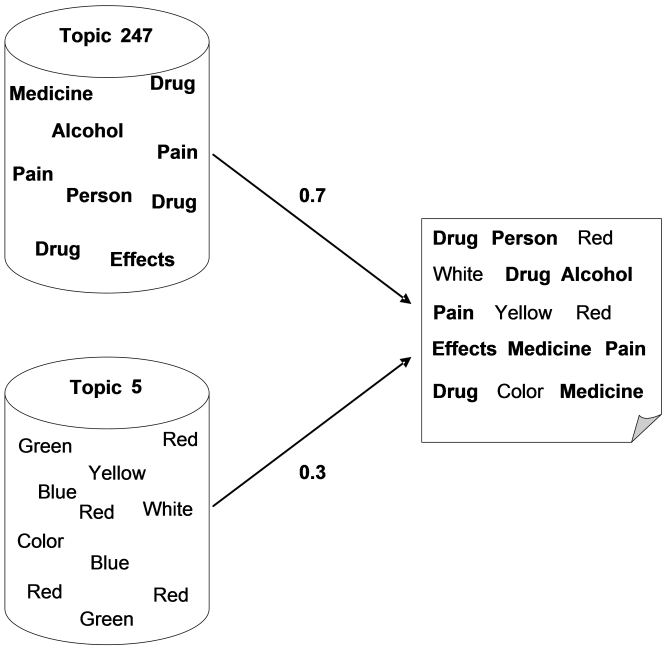



**Figure 3.7:** Document generation process. Mixing coefficient of $0.7$ and $0.3$.

according to this modification.

### 3.2.6 Probabilistic Topic Models (PTM) [103]

PTM endorse the main idea of topic extraction: a document can be interpreted as a linear combination of probability distributions over a given vocabulary, where each probability distribution, i.e. topic, is associated with a specific argument, idea or theme.

The generative model proposed by [103] adopt the smoothed generative model proposed in [16] (Figure 3.6) which associates a Dirichlet prior with hyper-parameter $\eta$, to the topic-word distribution $p\,(w|z)$. The authors suggest to interpret the hyper-parameter as prior observations on the number of times words are sampled from a topic before any word from the corpus is observed. This choice smooths the word distribution in every topic with an amount of smoothing determined by the value of the hyper-parameter. The authors suggest that a good choices for the hyper-parameters is dependent to the number of topics $K$ and vocabulary size $V$.

**Inference** The authors observe that many text collections contain millions of word tokens, and thus the estimation of the posterior over the topic requires the adoption of efficient procedures. While in [16], the the estimation $\beta$ is performed through variational approximation. [46] propose to directly estimate the posterior distribution $p(\mathbf{z}|\mathbf{w})$ and then obtaining the estimates of $\theta$ and $\beta$. Thus, they choose symmetrical Dirichlet priors for $\alpha$ and $\eta$, and computes the joint distribution of $p(\mathbf{w}, \mathbf{z})$ by integrating out $\theta$ and $\beta$. It is worthwhile to notice that the full joint distribution can be written as follows:

$$p(\mathbf{w}, \mathbf{z}, \theta, \beta|\alpha, \eta) = p(\beta|\eta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|\beta, \mathbf{z}). \tag{3.2.12}$$

Then integrating out $\theta$ and $\beta$ leads to:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \eta) = \int\int p(\beta|\eta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|\beta, \mathbf{z})d\theta d\beta. \tag{3.2.13}$$

Separating the integrals by pulling out the terms dependent on the variable being integrated:

$$p(\mathbf{w}, \mathbf{z}|\alpha, \eta) = \int p(\mathbf{z}|\theta)p(\theta|\alpha)d\theta \times \int p(\beta|\eta)p(\mathbf{w}|\beta, \mathbf{z})d\beta. \tag{3.2.14}$$

Hence it is possible to evaluate the posterior as:

$$p(\mathbf{z}|\mathbf{w}) = \frac{p(\mathbf{w}, \mathbf{z})}{\sum_{\mathbf{z}} p(\mathbf{w}, \mathbf{z})}. \tag{3.2.15}$$

However, the denominator requires to evaluate the distribution in a large sample space that is intractable. The authors then proposes to utilize a Gibbs sampling procedure, which is easy to implement and provides a relatively efficient method to compute such distribution. Details of the computations can be found in [46] and in [23].

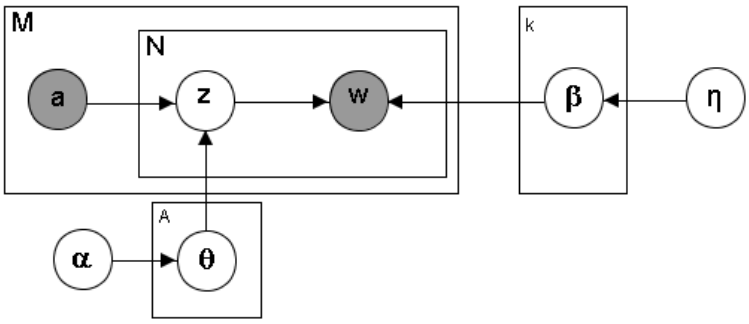



**Figure 3.8:** Author-Topic Model.

### 3.2.7 Extended Models

In the last years several models for topic extraction have been proposed. Each of them take as building block the LDA model and add other component that creates richer and specialized models. In this section we will briefly present the main and more interesting extended models. In particular we will describe the *Author-Topic*, the *HMM-LDA* models, and the hierarchical topic models.

#### Author-Topic Model [99]

The author-topic model (*AT-LDA*) extends LDA by incorporating the authorship of the documents in the corpus. In particular AT-LDA assumes that the observed variables are both the documents (i.e the words) and the set of associated authors; the inference procedure discovers the topics discussed in the documents and the set of authors associated to such topics. A multi-author document will be characterized by topics that are shared among the authors.

In the generative model each author $a_n \in \mathcal{A}$ is characterized by a probability distribution over topics ($\theta$), and each topic is described as a distribution over words ($\beta$): documents are generated by picking an author at random and then sampling a topic from the author-topic distribution and finally extracting a word from the specific word-topic distribution. Formally the generative process for a document is defined as follows:

1. For each author $a$, choose the topic mixing proportion $\theta_a \sim$ Dirichlet($\alpha$),
2. For each topic $z$, choose the word-topic distribution $\beta_z \sim$ Dirichlet($\eta$),
3. For each word in the document:
   (a) Choose an author $a_n \sim$ Uniform($\mathcal{A}$),
   (b) Choose a topic $z_n \sim$ Multinomial($\theta_{a_n}$),
   (c) Choose a word $w_n \sim$ Multinomial($\beta_{z_n}$).

The probability associated with the corpus can be defined as follows:

$$P(\mathcal{D}|\mathcal{A},\alpha,\eta) = \int\int p(\theta,\beta|\alpha,\eta)\prod_{d=1}^{M} P(\mathbf{w}_d|\mathcal{A},\theta,\beta)d\theta d\beta. \tag{3.2.16}$$

Equation 3.2.16 implements the same strategy as PTM in which the topic weight distribution $\theta$ and the word-topic distribution $\beta$ are treated as random variable and integrated out while $p(\theta,\beta|\alpha,\eta) = p(\theta|\alpha)p(\beta|\eta)$ are the Dirichlet priors.

**Inference** The inference strategy proposed by the authors exploits a Gibbs Sampler procedure to approximate the posterior $p(\theta,\beta|\mathcal{D},\alpha,\eta)$. In particular the authors use this posterior to derive other quantities used in different tasks, like information retrieval or as a tool to find the most surprising document from an author. The inference scheme is based on the observation that:

$$p(\theta,\beta|\mathcal{D},\alpha,\eta) = \sum_{\mathbf{z},\mathbf{a}} p(\theta,\beta|\mathbf{z},\mathbf{a},\mathcal{D},\alpha,\eta)P(\mathbf{z},\mathbf{a}|\mathcal{D},\alpha,\eta). \tag{3.2.17}$$

The Gibbs Sampler procedure is used to obtain an empirical sample estimation of $P(\mathbf{z},\mathbf{a}|\mathcal{D},\alpha,\eta)$, then $p(\theta,\beta|\mathbf{z},\mathbf{a},\mathcal{D},\alpha,\eta)$ is computed by exploiting the conjugacy between Dirichlet and multinomial distributions. Details of Gibbs Sampler derivation can be found in [99].

**Further extensions:** [88] propose some extension and differentiation of the basic author topic model that replace authors with different entities in the corpus; The proposed models are:

- *Conditionally-Independent LDA (CI-LDA)* that makes an explicit a priori differentiations of word tokens by means of common words and entities. The work is similar to the one proposed by [28].
- *Switch-LDA* is the fully generative model derived from CI-LDA.
- *CorrLDA-1* that tries to overcome the decoupling that implicitly affects Switch-LDA by first making inference about word topics and consequently associating entities to such topics.
- *CorrLDA-2* that modify CorrLDA-1 by allowing word topics including mixture of entity topics, i.e topics about sports is related to entity topics about soccer, football, and basket players.

[22] propose the User-Topic-Tag (UTT) model in which a collaborative tagging process in modeled. The generative process assumes that each user cites a document based on his interest (i.e. users are modeled as the authors in AT model) and then applies the tags to the documents according to its content. In particular, each topic is associated with a distribution over words (as in LDA), and moreover each topic is also associated with a multinomial distribution over tags. A comparison of different models for annotated documents is presented in [21]

**Integrating topic and syntax (HMM-LDA) [45]**

The words used in a document have two different purposes: semantic and syntactic. Syntactic words are functional words connecting the semantic ones which express the meaning of documents. This two classes of words have different behaviors in a text: semantic words have long range dependencies (i.e different sentences can have similar content) while syntactic constraints are normally sentence-dependent and have short range dependencies. The authors propose a composite model: an LDA model

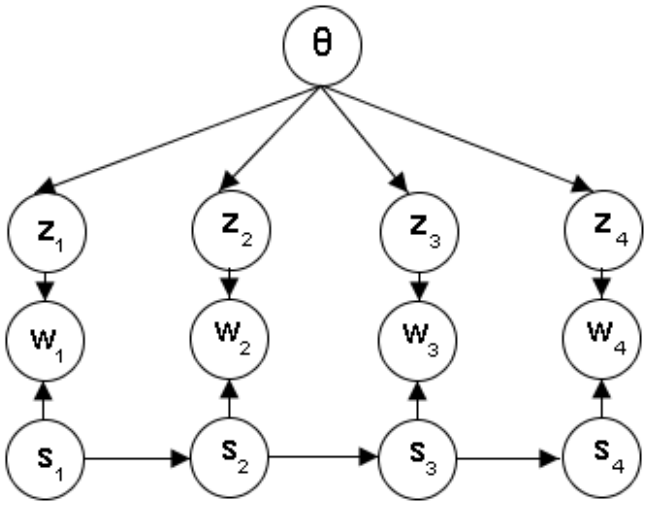



**Figure 3.9:** Simplified HMMLDA graphical model.

deals with semantic classes and a Hidden Markov Model (HMM) determines when to emit words from the syntactic class.

Thus, for each word $w_i$ in the document, the generative model takes into account the corresponding topic $z_i$ and also the particular state $s_i$ of the chain that emits the word class $c_i$. In particular, whenever the chain status selects a class $c = 1$ the word belong to the semantic class and then its topic $z$ is sampled from $\theta$ (i.e. topic mixing proportions), while if the status is greater then 1 a syntactic class $c$ is selected. Thus the word is sampled from $\phi_z$ when the chain is in a semantic state or from $\beta_c$ for syntactic states. Class transition follows a distribution $\pi_{(s-1)}$ that is a row of the transition matrix $\Pi$.

The generative model for a document in the corpus is the following:

1. Choose topic mixing proportion $\theta \sim$ Dirichlet($\alpha$),
2. Choose word-topic distribution for semantic class $\beta_t \sim$ Dirichlet($\eta$),
3. Choose a row of the transition matrix $\Pi$ as $\pi_{i,\cdot} \sim$ Dirichlet($\gamma$),
4. Choose word-class distribution for syntactic classes $\beta_c \sim$ Dirichlet($\delta$),
5. For each word in the document:

(a) Choose $z_i \sim \theta^d$,

(b) Choose $c_i \sim \pi^{(c-1)}$,

(c) **If** $c_i = 1$ **then** $w_n \sim \beta_t^{z_i}$ **else** $w_n \sim \beta_c^{c_i}$.

**Inference** The authors solve the inference problem via a two step Gibbs sampling procedure in which $\beta$, $\theta$ and $\pi$ are treated as parameters and consequently integrated out. In particular, in the first part of the Gibbs sampler a topic assignment $z_i$ is drawn conditioned on all other assignment $\mathbf{z}_{-i}$, on class assignments $\mathbf{c}$ and the observed words $\mathbf{w}$:

$$P(z_i|\mathbf{z}_{-i}, \mathbf{c}, \mathbf{w}) \propto P(z_i|\mathbf{z}_{-i})P(w_i|\mathbf{z}, \mathbf{c}, \mathbf{w}_{-i}). \tag{3.2.18}$$

Similarly the class assignment is drawn from:

$$P(c_i|\mathbf{c}_{-i}, \mathbf{z}, \mathbf{w}) \propto P(w_i|\mathbf{z}, \mathbf{c}, \mathbf{w}_{-i})P(c_i|\mathbf{c}_{-i}). \tag{3.2.19}$$

In the experimental section, however, the author states that to use a $3^{rd}$ order Markov Chain. This modification implies the inclusion of an additional term in $P(c_i|\mathbf{c}_{-i})$. Details of the computation can be found in [45] and [47];

**Hierarchical Topic Models**

A great deal of efforts have been focused on developing hierarchical topic models. Models like LDA and its extensions extract flat sets of topics, that sometimes does not capture the intrinsic hierarchical structure, and leave to the researcher the burden of the organization. Different approaches have been proposed in specialized literature: in particular it is possible to distinguish two different approaches.

On one hand there are post processing solutions like the one proposed by [119] that exploits particular metrics to evaluate the specificity of topics and build an ontology. Also [123] utilize topic models to build an ontology by means of topic models.

On the other hand several Bayesian approaches have been proposed: [14] exploit the *Chinese Restaurant Process* [2] by applying a nested structure

that defines an infinitely branching tree with chosen depth, and in which the nodes are the topics and documents are characterized by paths in the tree. In [13] the model is extended by allowing infinite branching and infinite depth topic trees that are described by two different stochastic processes: the *nested Chinese restaurant process* defines the branching characteristic, while the *stick braking process* [93] put weights on the nodes, i.e. the topics proportions, along the paths. In particular, this model describes documents as a path in the tree and force general topics to higher level of the trees. The stochastic processes described are both particular characterization of the Dirichlet Process [3].

[33] propose a model similar to [14] that exploits a nested *Hierarchical Dirichlet Process (HDP)* [108] to create more fine grained and compact topic tree.

[68] and [66] presented a model that exploits a directed acyclic graph (DAG) to capture nested correlation between topics: in particular the leaves of the DAG are individual words of the vocabulary, internal nodes represents correlation between its children (i.e. words or topics). [67] use a nonparametric Bayesian prior based on the HDP that allows the model to learn the number of topics and their relation.

## 3.3 Topic Labeling

Topic Labeling is a key issue in developing topic models, especially whenever the model is used as an intermediate step in application like information retrieval or document classification [83]. Nonetheless, research community has spent little attention on this problem and often it they relies on manual labeling.

This section presents two of the more interesting approaches to topic labeling: the first one is specifically devoted to the label assignment problem, the second utilizes a different approach to create more meaningful topic by means of a post processing step.

### 3.3.1 Automatic Labeling of Multinomial Topic [79]

The proposed model tries to overcome the weakness of human labeling for topic models. A manual labeling has to face many difficulties especially with larger models that normally are quite specific in the content and maybe overlapping. In this set up the labeling process is prone to errors due to the difficulty in the choice of meaningful, precise and unbiased labels.

According to the authors a topic labeling procedure should satisfy the following constraints:

- labels should be *understandable* by the user;
- labels should capture the meaning of topics i.e *relevant*;
- labels should distinguish a topic from others i.e *discriminative*;
- label should offer a *high coverage* of the labeled topic;
- labels should be *objective and unbiased*;

The main ingredients of the labeling problem can be formally described by:

- a *topic model* $\theta = \{z_1, z_2, \cdots, z_t\}$ describing a document corpus $\mathcal{D}$. where each $z_i$ is a probability distributions over words.
- a set of *topic labels* $\mathcal{L} = \{l_1, l_2, \cdots, l_t\}$ in which each $l_i$ is defined by a sequence of words semantically meaningful to the latent topic meaning;
- a *relevance score* $S(l_j, z_i)$ that measure the semantic similarity between a label and a topic.

The *Topic labeling problem* consists in finding the best set of labels $\mathcal{L}$ that match a topic model $\theta$ according to the relevance score adopted.

**Topic Labeling** The labeling process requires to find candidate labels by extracting relevant phrases form the document corpus: two distinct approaches has been identified, the simplest one requires to extract phrases using a Natural Language Processing (NLP) chunker and selecting the most frequent ones. The other approach, instead, requires to extract the best n-grams from the reference collection to build meaningful phrases. To select the n-grams it is possible to exploit measures like *mutual information* or particular statistic tests like $\chi^2$ or Student's $T - test$.

The authors propose two distinct relevance score. *Zero-order Relevance* can be simply computed by evaluating a label $l = w_1, w_2, \cdots, w_k$ against a specific topic:

$$Score_{0r} = log\frac{p(l|z_i)}{p(l)} = \sum_{j=1}^{k} log\frac{p(u_j|z_i)}{p(u_j)}. \tag{3.3.1}$$

where $p(u_i)$ is a normalization factor that smooths the score for short phrases, and it can be computed by some background collection or set to uniform. It is worthwhile to notice that this metric gives more weights to the higher ranked words inside a topic, limiting the coverage of the label on the specific topic. Another possible disadvantage of the metric is the absence of any contextual information from the reference collection $\mathcal{D}$.

*First-order relevance* tries to find labels with the best coverage with respect to the extracted topics. In this set up each label is defined as a multinomial distribution over words, and thus it is possible to compute the closeness between the label and the topic by means of Kullback-Leibler divergence (see section 2.2.2).
Each label is associated to a distribution $p(w|l, \mathcal{C})$ which takes into account the context $\mathcal{C}$ of the label. The score is defined as follows:

$$\begin{aligned} Score_{1r} &= -KL(z_i|l_j) = -\sum_{w} p(w|z_i) log\frac{p(w|z_i)}{p(w|l,)} \\ &= \sum_{w} p(w|z) PMI(w, l|\mathcal{C}) - KL(z_i|\mathcal{C}) + Bias(l, \mathcal{C}) \end{aligned} \tag{3.3.2}$$

where $Bias(l, \mathcal{C})$ acts as a prior over labeling context and $PMI(w, l|\mathcal{C})$ is the

pointwise mutual information between $l$ and the terms in the topic model given the context.

In order to increase the coverage of the labeling w.r.t. the topic model the authors propose to select labels that maximize the Maximal Marginal Relevance defined as follows:

$$\hat{l} = \arg\max_{l \in \mathcal{L} - \mathcal{S}} (\lambda Score(l, z_i) - (1 - \lambda) Sim(l', l)) \tag{3.3.3}$$

where $\mathcal{S}$ is the set of label already selected, $Sim(l', l) = -KL(l', l)$ and $\lambda$ is a parameter empirically set.

Finally, the last criteria to satisfy is that labels should be discriminative, i.e. a good label should have high semantic relevance to the target topic, and low relevance for all the others. The authors propose to modify the scoring function as follows:

$$Score' = Score(l, z_i) - \mu Score(l, z_{-i}) \tag{3.3.4}$$

where $Score(l, z_{-i})$ represents the semantic of the label against all the other topic but $z_i$ and $\mu$ is an empirically chosen parameter that controls the discriminative power.

The author tested this criterion against two different document corpus: *SIGMOD conference proceedings* and *Associated Press News dataset* showing the robustness of the proposed method. Further details, the experimental section and evaluation can be found in [79].

### 3.3.2 Turbo Topics [15]

Once a topic model has been inferred from the reference corpus, the next step consists in the visualization of the topics, and their interpretation. Topic models rely on uni-gram representation, and thus the users have to analyze list of words ordered by decreasing probability and try to intuit the "meaning" of the topic. Literature has provided model that tries to overcome the uni-gram representation [115, 120] loosing the computation advantage of uni-gram representation.

The authors propose a different approach that can be interpreted as a

sort of post-processing step over a topic models. In particular given a corpus, the model is fitted as usual, and the posterior is used to annotate each document with its most probable topic. Then, the most significant n-grams are extracted from each topic by a co-occurrence analysis. The selected n-grams are combined with the extracted topic words list to offer to the users the possibility to better understand the "semantic" of the topic.

The *Turbo topics* model works as follows:

1. estimate an LDA topic model with $T$ topic,
2. use the posterior over words and topic to annotate each word in the corpus with a topic assignment. The resulting corpus will contain an ordered sequence of words and topics pairs,
3. iteratively repeat for each given word (or phrase) $w$ with topic label $z$ an hypothesis testing procedure which identifies other words $v$ that are likely to precede or follow $w$ with label $z$, until no more significant phrases are added.

The n-grams generation procedure is based on an arbitrary language model such that the log-likelihood can be computed as follows:

$$\mathcal{L}_{\mathbf{w}} = \sum_{n=1}^{W} logP(w_n|w_1, \cdots, w_{n-1}). \tag{3.3.5}$$

where $W$ represent the number of words in the corpus.

A fully parameterized model is obviously intractable, while if word independence is assumed the model is transformed in the usual uni-gram representation. The proposed solution for bi-grams discovery distinguishes two different type of bi-grams, on one side *non-true* bi-grams are assumed to follow a general distribution $\pi$, on the other side *real* bi-grams instead are assumed to be heavily influenced by their previous history. E.g. suppose we encountered the word `new`: if the following word is `house` the probability $P(w_i = \text{“house”}|w_{i-1} = \text{“new”})$ is considerably lower than $P(w_i = \text{“york”}|w_{i-1} = \text{“new”})$.

The model is then recursively expanded through the analysis of the likelihood ratio of joined bigger n-grams. Best n-grams are selected by choosing the words that joined increase the likelihood ratios. Details about the recursive procedure and results are presented in [15].

## 3.4 Topic Model Evaluation

Topic model evaluation has gained much attention due to the specifity of this models. Topic models are unsupervised algorithms that on one hand can be interpreted as a kind of *Document Clustering and Organization* but on the other side they are mixture models which components describe the corpus. Thus, there are no *gold standards* to which we can refer and compare. In literature there are different approaches to evaluation: statistical measures to evaluate the quality of the inference, human judgment, or evaluation based on a different interpretation of the model like the one proposed in this dissertation in Chapter 5.

### 3.4.1 Perplexity and Other statistical Evaluation metrics

The classical approach to topic model evaluation relies on the computation of *perplexity* on held out documents. Perplexity is a standard measure for statistical models of natural language, and indicates the uncertainty in predicting a word given a model. It can be informally seen as an evaluation of how much we are surprised to find specific words in a document given a learned model. Perplexity is monotonically decreasing in the likelihood (i.e. lower values indicate better results) of the test data and can be computed as:

$$P(\mathcal{D}_{test}) = exp\left(-\frac{\sum_{d=1}^{M} log\ P(\mathbf{w}_d|\beta)}{\sum_{d=1}^{M} N_d}\right). \tag{3.4.1}$$

In [116] the authors present an extensive analysis of evaluation metrics, and propose two new metrics that are declared to be less biased. The evaluation process is performed upon an held out set of document $\mathcal{W}$. The paper describe *importance sampling* [44] and *harmonic mean method* as state of the art methods and discuss their limitation in evaluating high dimen-

sional distribution; moreover, *annealed importance sampling* [86] is described as capable to deal with this problem. The paper contribution is represented by two different measures: a *Chibb-Style estimation* [27] and *left-to-right evaluation algorithm*. The experimental section shows how the two proposed measures have good performances with respect to the others, and shows how the *left-to-right algorithm* [117] has better performance on real-world corpora. For the technical details and evaluation the reader should refer to the original papers.

### 3.4.2 Human interpretation of topic models

The approach proposed in [26] investigates the interpretability of topic models according to human judgment. In particular, the authors propose a first measure to evaluate the quality of the an extracted topic in terms of semantic coherence. The second measure evaluate the quality of the document-topic assignment. Both this task are conducted by human evaluators hired through Amazon Mechanical Turk [1].

The evaluation of the semantic coherence of a topic model is conducted as follows: given a topic $z$ and the associated words list, present to the judges a set of $6$ word in which $5$ are selected to be the most probable words of the considered topic and the sixth is chosen to be a high probability word from a different topic (i.e. the intruder). This setup guarantees that the word comes from another semantic area and it is not just a rare word in the corpus. The evaluators are asked to identify the intruder. With this kind of experimental set-up users should easily identify the intruder whenever the chosen topic contains semantically related words while in a bad topic the intruder will be chosen at random in the words list.

Document-topic assignment can be interpreted as a topic intrusion test. The user is presented with a given document snippet and a set of 4 topics each described by their 8 most probable words, one of the topic is taken at random from low probability topics for the given document. The evaluator is instructed to identify the intruder.

[1] `http://www.mturk.com`

Word intrusion for topic $k$ is evaluated by computing the model precision as follow:

$$MP_k = \frac{\sum_s \mathbb{1}(i_s^k = \omega_k)}{S}. \tag{3.4.2}$$

where $\mathbb{1}(i_s^k = \omega_k)$ is an indicator function that is equal to $1$ whenever the user selected intruder topic $i_s^k$ equals the real intruder $\omega_k$.

The topic intrusion of a topic $k$ for a document $d$ is defined as the topic log-odds (TLO) of the topic proportion for the considered document. Let $\hat{\theta}_{d,s}^k$ the point estimate of topic proportion selected by the user $s \in S$ and let $\check{\theta}_d^k$ the point estimation of topic proportion of real intruder:

$$TLO_d = \frac{\sum_s log\check{\theta}_d^k - log\hat{\theta}_{d,s}^k}{S}. \tag{3.4.3}$$

The evaluation has been compared with statistical metric usually applied to topic models like held-out likelihood and information retrieval based scoring. The author showed how this measures are negatively correlated to human judgment and indicate how topic model evaluation is still an open problem.

# Chapter 4

# Improving Document Management with Probabilistic Knowledge Extraction

Enterprise Document Management (ECM) is becoming a corner stone for every company that has to deal with huge repositories of unstructured textual sources. Many big players are challenging to offer solutions that best fit document management needs. ECM is a container of many different solutions and technologies. According to Gartner [42] ECM platforms offer solutions for:

- *Document management*: document organization, version control and security policy for business documents.
- *Document imaging*: digitalization of paper based document (i.e capturing, transforming and managing).
- *Records management*: long-term document archiving according to compliance policies.
- *Workflow*: business processes management.
- *Web content management*: integration of contents for web publishing.
- *Document-centric collaboration*: document sharing for project teams.

The prerequisites of a ECM solution are the ability to offer ease of use, integrability in preexistent IT environments, legal compliance for document storage and environments.

The proposed approach is intended to be integrated in a Document Management System to avoid the burden of manually categorizing, labeling and retrieving documents. The solution exploits statistical methods together with unlabeled documents to extract semantic knowledge to be used in document repositories and thus enabling the system to achieve efficient and effective management that allows users to dominate and organize data and to make rational decisions upon them.

The knowledge extraction process is implemented through an information processing pipeline which transforms documents from plain to tagged. The tagging process, which is performed by exploiting a set of extracted topics and a user supplied taxonomy, is capable of extracting and exploiting the semantic structure hidden in a given document corpus. The pipeline relies on probabilistic topic models to extract topics from the document corpus. Then, each topic is automatically labeled according to a user supplied taxonomy through the ALOT algorithm [71]. The learnt topics together with their labels allow to automatically tag the document corpus. Such tagging task is a multi-label document classification problem, and it is performed by exploiting a multi-net Naïve Bayes model that avoids the usual learining procedure and directly maps the output of the topic extraction process to the parameters of the multi-net Naïve Bayes model [62,74].

The chapter is organized as follows. Section 4.1 is devoted to introducing and describing the main ingredients of the information processing pipeline. In particular subsection 4.1.1 will describe the topic extraction and model assessment procedure; in subsection 4.1.2 the automatic labeling of topic algorithm is described and finally the multi-net Naïve Bayes supervised classification model is presented in subsection 4.1.3. Numerical experiments illustrating the functionalities of the information processing pipeline are described in Section 4.2. Future research directions, emerging trends, paradigms and conclusions close the chapter.

## 4.1 Knowledge Extraction components

The pipeline is intended to satisfy the needs of business companies with a little impact on their internal procedures. The system should be able to deal with the business company document collection and automatically extracts the relevant topics, labels them according to the company view and offers a reliable service that grants the labeling of new documents coherently with respect to the old ones.

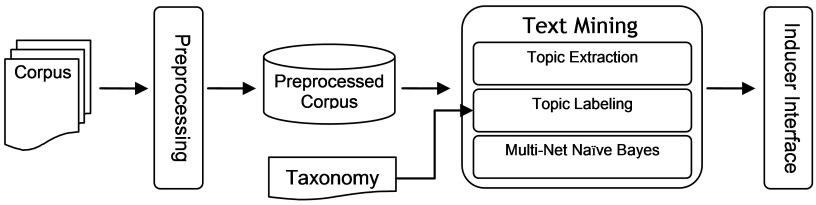



**Figure 4.1:** Information Processing Pipeline

The modules of the information processing pipeline, together with their interactions, are illustrated in Figure 4.1. In particular the modules of the pipeline are:

- *Preprocessing*: data manipulation and transformation,
- *Text Mining*:
  - *Topic Extraction*: topic extraction and model assessment,
  - *Topic Labeling*: topic labeling and organization according to a given taxonomy,
  - *Multi-Net Naïve Bayes*: multi-label document classification according to extracted topic.

### 4.1.1 Topic Model Assessment

Given a corpus of documents and a set of candidate topic models a key problem is to select the most meaningful and representative set of topics that explain the corpus. The approach adopted exploits a technique based on the evaluation of an infra-topic similarity according to a *symmetrized*

*Kullback-Leibler* measure. Given the symmetric matrix generated from the distance computation, a *hierarchical clustering algorithm* is applied and first creates clusters composed of pair of objects that are close together, then the algorithm recursively links each cluster to the others creating bigger clusters until all the objects of the original dataset are linked together in a hierarchical tree (i.e. the root of the tree is one single cluster containing all the initial data). The aggregation strategy computes a problem-dependent distance function between the content of each cluster. The proposed solution utilizes the *average linkage function*:

$$Link_{avg}(X,Y) = \frac{1}{|X| \cdot |Y|} \sum_{i}^{|X|} \sum_{j}^{|Y|} d(x_i, y_j) \tag{4.1.1}$$

where $x_i \in X, y_j \in Y$, $|X|$ is the cardinality of cluster $X$ and $d(x_i, y_j)$ is the distance between the argument objects.

The result of a hierarchical clustering is conveniently represented as a dendrogram ( Figure 4.2). The horizontal axis represents the distance at which successive clusters are joined.

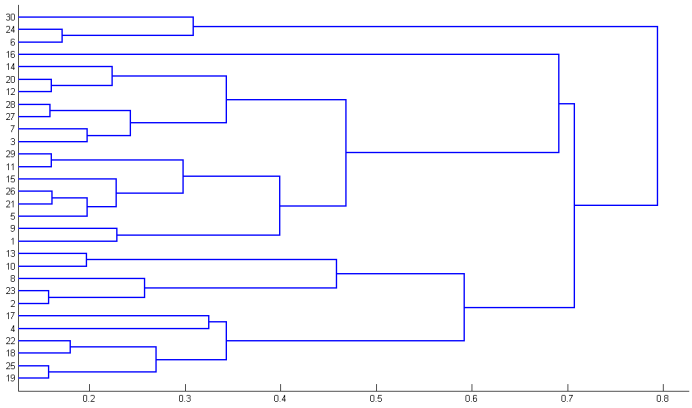



**Figure 4.2:** Dendrogram example

On the vertical axis all the data points are represented. After the algorithm has computed the cluster tree, the system chooses the "optimal" height at which the tree should be cut. [85] provides a stopping criterion

based on the heights at which the clusters are joined. Formally the rule states that the cut height is computed as:

$$cut_{height} = \overline{h} + \kappa\sigma_h \tag{4.1.2}$$

where $\overline{h}$ is the average of the heights for all the clusters, $\sigma_h$ is the standard deviation of the heights and $\kappa$ is a specified constant. Mojena suggested a value $\kappa$ satisfying $2.5 < \kappa < 3.0$. Successive analysis suggested a value equal to $1.25$ [82].

### 4.1.2 Automatic Labeling of Topic

As described in section 3.3 topic labeling is an open and emerging problem. The proposed algorithm tries to conjugate and reconcile human judgment with computer discovered solutions. The idea originated from the observation of business companies best practice. Indeed, many business companies organize their document collection through the use of specialized document management systems. These systems require the user to associate each document against a taxonomy or a given controlled vocabulary. Such taxonomies are built to exactly match the company needs and goals. They are context dependent and describe the full knowledge base of the business company. However, final users normally ignore such taxonomies and give bad or no labeling to their documents, and thus making useless the document management system. The proposed approach allocates the extracted topics inside the hierarchy, finds the associated label and organization.

#### Topic Tree

The taxonomy [56] is encoded in a particular structure defined as *topic tree*. A *topic tree* (Figure 4.3) is a pair $\Upsilon = \langle V, E \rangle$, where $V$ is a set of nodes indexed by non negative integers $j = 0, 1, ..., N$, while $E = V \times V$ is a set of arcs $(i, j)$ between nodes, $i, j \in V$. Each node $j$ is associated with a topic $T_\Upsilon(j) = \langle label, words\ list, infos \rangle$, where *label* is the topic label, *words list* is the topic list of positive words and *infos* is additional information associated

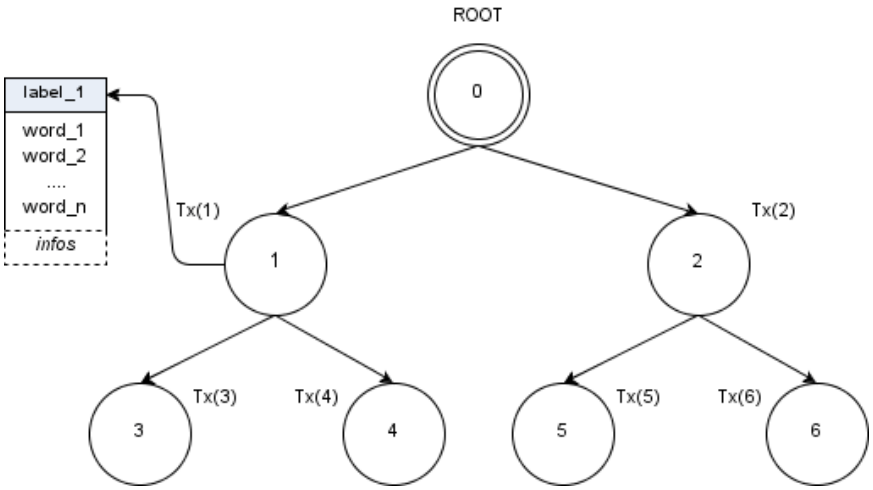



**Figure 4.3:** Topic Tree

with the topic. It is worthwhile to mention that the *root node* indexed by 0 is not a proper topic. It is introduced to ensure that the set of topics, which usually gives rise to a forest, forms a tree. Therefore, the *root node* can be interpreted as the most generic topic or *all-the-topics*. It is worthwhile to mention that the framework considered in this approach assumes that the world is described by a set of concepts (equivalently, topics) which are inserted into a light ontology [84]. The *topics tree* $\Upsilon$ describes how topics are linked in a taxonomic way by means of the usual IS-A relation. A concept $c$ IS-A concept $d$ iff $I(c) \subseteq I(d)$, where $I$ is an interpretation function $I : \mathcal{C} \mapsto U$ mapping a concept $c \in \mathcal{C}$ to a subset $I(c)$ of a given universe $U$. For instance, under the common-sense interpretation, *cat* IS-A *feline* since any real cat belongs to the set of felines (but not vice versa).

### The ALOT Algorithm

An *extracted topic* is a word list, obtained from the application of the LDA method. Given a topics tree $\Upsilon$ and a set of extracted topic $\top = \{T_e(1), ..., T_e(K)\}$, the algorithm for Automatic Labeling Of Topics (ALOT) aims to label each element $T_e(i)$, $i = 1, ..., K$, by means of labels associated with topics $T_\Upsilon(j)$, $j = 1, ..., N$ of the topics tree $\Upsilon$.

The main components of ALOT are the similarity measures and the la-

**Algorithm 1** Automatic Labeling Of Topics (ALOT)

**Require:** A topics tree $\Upsilon$, a topic $T_e(i)$ to be labeled.
**Ensure:** The label of $T_e(i)$.

Compute $j_r^* = \arg\max_j S_r(T_e(i), T_\Upsilon(j))\ \forall r, r = 1, ..., 6$, and set $L(i) = \{j_1^*, ..., j_6^*\}$
**if** $j_1^* = \ldots = j_6^*$ **then** {*case **TC***}
  Return the $T_\Upsilon(j_1^*)$ label
**else** {*case **TD***}
  Case **Path**: Find $j$, the shallowest topic in $\Delta(i)$
  Case **Subtree**: Find $j$, the deepest predecessor of nodes belonging to $\Delta(i)$
  **if** $T_\Upsilon(j) \neq ROOT$ **then** {*case **SA***}
    Return the $T_\Upsilon(j)$ label
  **else** {*case **NSA***}
    Compute $j^{max}$ which maximizes $depth(j^{max})$ and $|successor(j^{max}) \cap \Delta(i)|$
    **if** $j^{max}$ is unique **then** {*case **S-dtmap***}
      Return the $T_\Upsilon(j^{max})$ label
    **else**
      Apply subcase **M-dmatp** and return the computed label if unique or ROOT if not (subcase **R-dmatp**)
    **end if**
  **end if**
**end if**

beling rules. While similarity measures, introduced in section 2.2.1, are concerned with the *word list* component of topics, labeling rules exploit the topics tree to find the *optimal label* (w.r.t. the available topics tree $\Upsilon$) for each extracted topic $T_e(i)$. More in detail, given a topics tree $\Upsilon$, for each extracted topic $T_e(i)$ its nearest topic $T_\Upsilon(j_r^*)$, with respect to similarity measure $S_r$, is recovered by solving the following optimization problem:

$$j_r^* = \arg\max_j S_r(T_e(i), T_\Upsilon(j)).$$

That is, $T_\Upsilon(j_r^*)$ is the topic which has the greatest similarity $S_r$ with $T_e(i)$ and $j_r^*$ the index of this topic in $\Upsilon$. For each extracted topic $T_e(i)$, we collect all these indexes associated to the similarity measures $S_r$ in

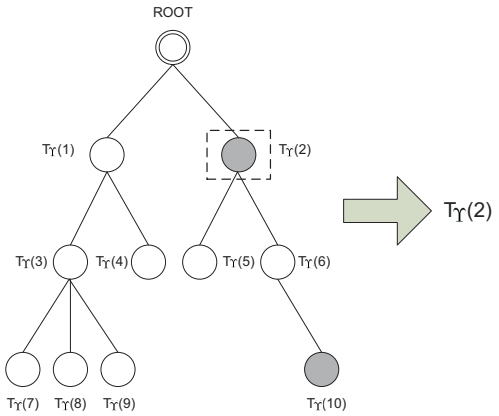
$L(i) = \{j_1^*, ..., j_6^*\}$, and the corresponding set of topics will be denoted by $\Delta(i) = \{T_\Upsilon(j_1^*), ..., T_\Upsilon(j_6^*)\}$. Both $L(i)$ and $\Delta(i)$ will be represented on the tree structure by coloring the corresponding nodes. For instance, in Figure 4.4(b), $L(i) = \{4, 7, 9\}$ (and this means that the six similarity measures give only three different results).

Given $T_e(i) \in \top$, the following cases can occur:

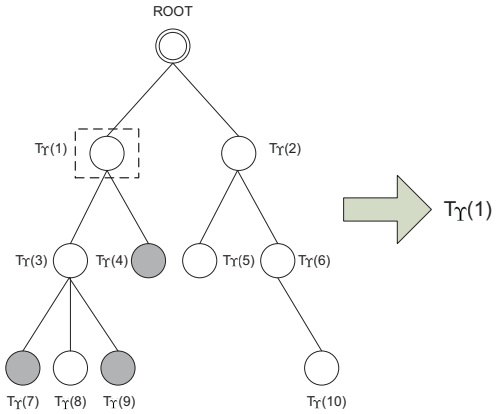
- *Topic concordance* (**TC**); $j^* = j_1^* = ... = j_6^*$, all similarity measures agree on which the nearest topic $T_\Upsilon(j^*)$ is. The ALOT algorithm labels $T_e(i)$ with the label of the corresponding optimal unique topic $T_\Upsilon(j^*)$.

- *Topic discordance* (**TD**); $\exists l, g : j_l^* \neq j_g^*$, at least two similarity measures disagree on which the nearest topic is. The ALOT algorithm labels $T_e(i)$ according to:

  1. **SA** (Semantic Association, topics belonging to $\Delta(i)$ share a predecessor, different from the root). The ALOT algorithm looks for a topic $T_\Upsilon(j)$, not necessarily belonging to $\Delta(i)$, which synthesizes all the topics in $\Delta(i)$. The following subcases can occur:
     (a) **Path**; all the topics in $\Delta(i)$ lie on the same path. ALOT labels $T_e(i)$ with the label of the shallowest topic in $\Delta(i)$, i.e., the topic $T_\Upsilon(j_r^*)$ which minimizes $depth(j_r^*)$ (Figure 4.4(a)).
     (b) **Subtree**; all the topics in $\Delta(i)$ belong to a common subtree. The ALOT algorithm labels $T_e(i)$ with the label of the topic $T_\Upsilon(j)$ which is the common deepest predecessor of topics in $\Delta(i)$ (Figure 4.4(b)). $T_\Upsilon(1)$. Notice that the case where $T_\Upsilon(j) \in \Delta(i)$ can also occur.
  2. **NSA** (Non-Semantic Association, topics belonging to $\Delta(i)$ do not share a predecessor, except from the root). ALOT uses a majority voting scheme and selects the deepest maximally agreed topics predecessor, i.e. the topic $T_\Upsilon(j^{max})$ associated with the node $j^{max}$ such that $depth(j^{max})$ and $|successors(j^{max}) \cap L(i)|$ are both maximized. The following subcases can occur:
     (a) **S-dmatp** (Single deepest maximally agreed topic predecessor); a single topic $T_\Upsilon(j^{max})$ is obtained and its label is asso-




(a) Semantic Association: Path




(b) Semantic Association: Subtree

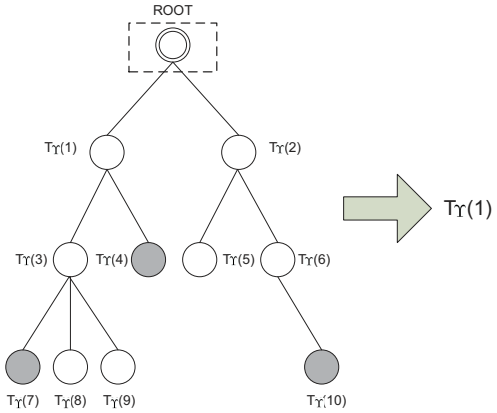
**Figure 4.4:** Semantic Association ALOT Cases.

ciated with $T_e(i)$. In Figure 4.5(a), the selected topic is $T_\Upsilon(1)$ since it has two successors in $\Delta(i)$ compared to only one on the other branch of the tree.

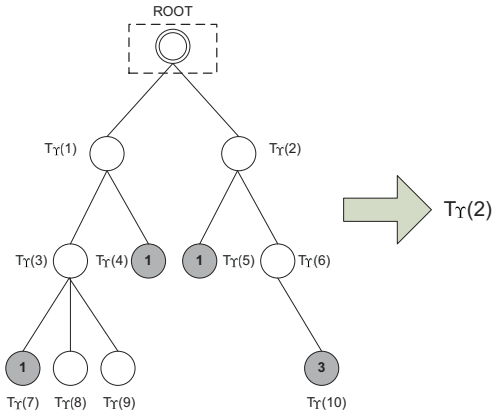
(b) **M-dmatp** (Multiple deepest maximally agreed topic predecessor); more than one topic is returned by the majority voting scheme. ALOT computes how many times each $T_\Upsilon(j_l^*)$ is a descendant of all the $T_\Upsilon(j^{max})$, stores this information into *info* and finds the $T_\Upsilon(j^{max})$ with the maximum number of occurrences. In Figure 4.5(b), the majority voting returns $T_\Upsilon(1)$ and $T_\Upsilon(2)$ and between them, $T_\Upsilon(2)$ is selected since

it has four successors in $\Delta(i)$ compared to only two of $T_1(x)$.

(c) **R-dmatp** (Rooted deepest maximally not agreed topics predecessor); the $root\ node$ is returned by the majority voting scheme and the maximum number of occurrences $T_\Upsilon(j^{max})$ is the same for at least two descendants in $\Delta(i)$. Then, the $root\ node$ is returned by ALOT.




(a) Non Semantic Association: S-dmatp




(b) Non Semantic Association: M-dmatp

**Figure 4.5:** Non Semantic Association ALOT Cases

### 4.1.3 Multi-label Document Tagging

The topic extraction step offers a set of topics that project the document collection in a lower dimensional space, which represents the core arguments, while the ALOT algorithm associates each topic with reliable labeling and with an allocation inside the hierarchy. The tagging of new documents accordingly with the extracted topics is an important feature in business environments. A reliable tagging allows user to retrieve important document seamlessly and efficiently. The proposed classifier is a specific instance of the Multi-Net Naïve Bayes classifier (MNNB) [62] that despite its simplifying assumptions (i.e. the attributes are independent given the class variable) gives good performances. The MNNB classifier is the obvious choice for the document classification step due to the Bayesian nature of LDA, and although other algorithms like Support Vector Machines are known to have very good performances in binary and multi-class problems, in a multi-label set-up they are known to have long training phase and suffer of less good performances. Moreover, the linking of SVM with LDA models is more complex, not straightforward and implies the discarding of the Bayesian approach. The Naïve Bayes classifier computes the probability that a document $d_j$ represented as a vector $\underline{w}$ of words, belongs to a class $z_i$ through the Bayes' theorem as follows:

$$P(z_i|\underline{w}) = \frac{P(\underline{w}|z_i) \cdot P(z_i)}{p(\underline{w})} = \frac{e^{t_{ij}} \cdot P(z_i)}{e^{t_{ij}} \cdot P(z_i) + P(\overline{z_i})} \tag{4.1.3}$$

where:

$$t_{ij} = \log \frac{P\left(\underline{w}|z_i\right)}{P\left(\underline{w}|\overline{z_i}\right)} \tag{4.1.4}$$

Exploiting this framework and arranging the derivation of $t_{ij}$ accordingly to different document representation it is possible to compute $P(z_i|d_j)$.

The $t_{ij}$ factor for *binary* representation is defined as follows:

$$t_{ij}^{binary} = \sum_{w \in \underset{\rightarrow}{w}} 1_w \log \frac{P(w|z_i)}{P(w|\overline{z_i})} \tag{4.1.5}$$

where $w$ is a word of the vocabulary, $1_w$ is an indicator function equal to 1 if $w \in d_j$, and 0 otherwise.
The *term frequency* representation is as follows:

$$t_{ij}^{TF} = \sum_{w \in \underset{\rightarrow}{w}} tf(w) \log \frac{P(w|z_i)}{P(w|\overline{z_i})} \tag{4.1.6}$$

where $\mathrm{tf(w)}$ is the term frequency of word $w$ for the document $d_j$.

The *term frequency inverse document frequency* representation is defined as follows:

$$t_{ij}^{TFIDF} = \sum_{w \in \underset{\rightarrow}{w}} \mathrm{tf-idf(w)} \log \frac{P(w|z_i)}{P(w|\overline{z_i})} \tag{4.1.7}$$

where $\mathrm{tf-idf(w)}$ is equal to the term TF-IDF of the word $w$ in the document $d_j$.

**From LDA to Naïve Bayes**

Equation 4.1.3 and 4.1.4 shows the quantity needed to compute the desired posterior. Thus the classifier exploits the values computed by the LDA model for the prior probability over the topics $P(z_i)$, and the conditional probability $P(w|z_i)$ for each word in the vocabulary given the topic. The derivation of *non-class conditional probability* $P(w|\overline{z_i})$ can be derived by remebering that given a topic $z_i$ we can compute $P(w)$ as:

$$P(w) = P(w|z_i) \cdot P(z_i) + P(w|\overline{z_i}) \cdot P(\overline{z_i}). \tag{4.1.8}$$

then the *non-class conditional probability* is defined as follows:

$$P(w|\overline{z_i}) = \frac{P(w) - P(w_j|z_i) \cdot P(z_i)}{P(\overline{z_i})} \tag{4.1.9}$$

where $P(w)$ is defined to be:

$$P(w) = \sum_{i=1}^{T} P(w|z_i) \cdot P(z_i) \tag{4.1.10}$$

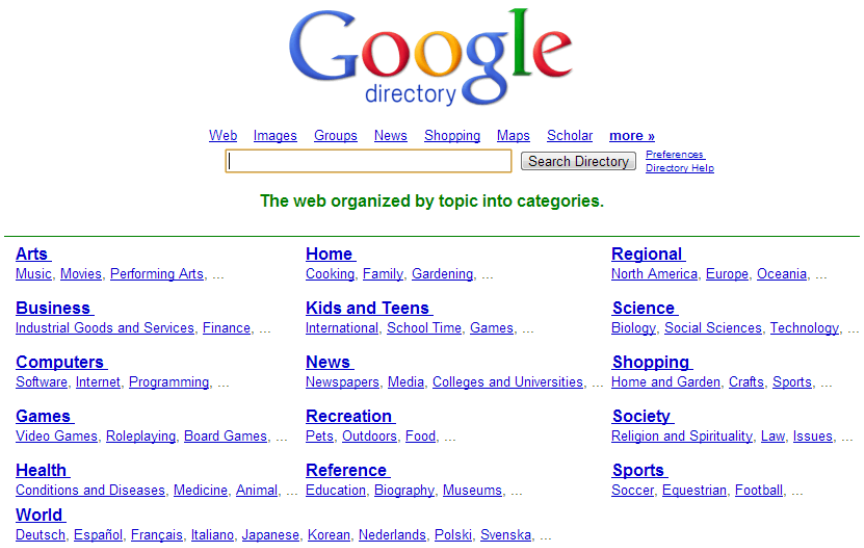
## 4.2 Numerical Experiments

In order to evaluate the quality of the pipeline it is customary to find a corpus with an adequate number of documents which presumably contain a variety of topics and a related taxonomy to exploit. Common literature corpora like Reuters-21875 satisfy the first prerequisite, but usually lack an associated taxonomy. Taxonomy, are typically used in the business world or in very specific sectors where the taxonomy is tailored for the specific needs. Possible examples are represented by Medical Subjects Headings, Criminal Law - Lawyer Sources and the Google Directory [30, 34, 80] that requires the related corpus to be harvested and mined.

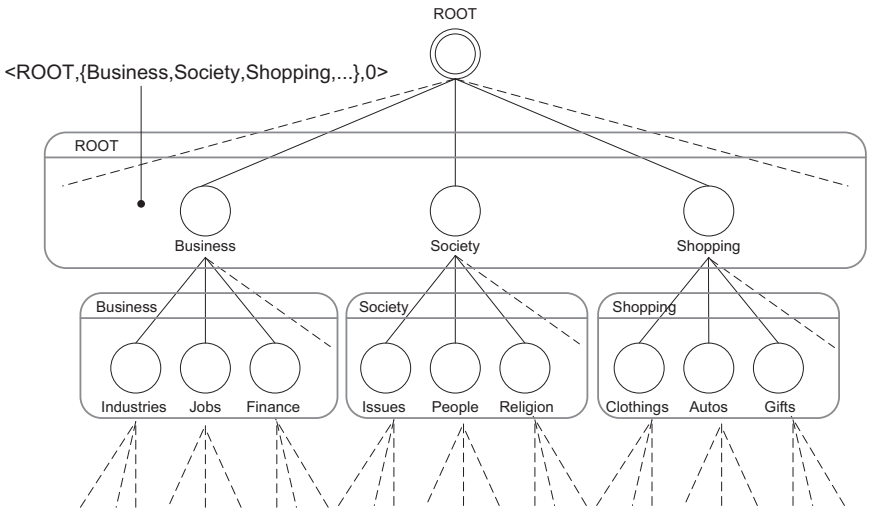
In the proposed experimental set-up, we choose as reference taxonomy the Google Directory (*gDir*), which is part of the Open Directory Project. This project manages the largest human-edited directory available on the web. Editors guarantee fairness and correctness of the directory.

### 4.2.1 Document Corpus and Taxonomy

The Google Directory (Figure 4.6(a)) is a hierarchical structure that organizes web-sites according to 16 macro categories, and for each category offers a classification subtree of variable depth. The experimental setup used all the 16 macro-categories, but discarded the implicit category ADULT that is a mere replication of the hierarchy including adult suggested pages. The tree has been cut to depth $5$. The *topic tree* (Figure 4.6(b)) has been built as follows: each node of the directory is a topic and its children are the words that specify the topic, this structure is repeated recursively. The topic tree contains $4,516$ nodes. The corpus has been generated by submitting a set consisting of $960$ queries to the Google search engine through the Google Ajax API. Each query is formed by a couple of words randomly selected from the union of word lists associated with the topic tree. Some examples of random queries are "`Music Environment`", "`News and Media Current Events`", "`Holidays Ukrainian`'. For matters of simplicity, the results are filtered, and only PDF files written in English are retrieved. The query process retrieved $46,480$ documents.




(a) The Google Directory




(b) Google Directory Topic Tree

**Figure 4.6:** Document Corpus and Taxonomy.

### 4.2.2 Preprocessing

The document corpus has been standardized to plain text (TXT) format. The preprocessing stage consisted of stop-words removal. Then, in order to regularize the document lengths, we applied a size-based filtering: only documents with a size between $2$ and $400$ KB have been retained, thus almost empty documents and too long or garbage documents are discarded. The filtered document corpus consisted of $33,801$ documents, while the global vocabulary has been reduced by removing the distribution tails: words mentioned in less than $10$ or in more than $2,551$ documents are removed: the resulting vocabulary consists of $111,795$ words.

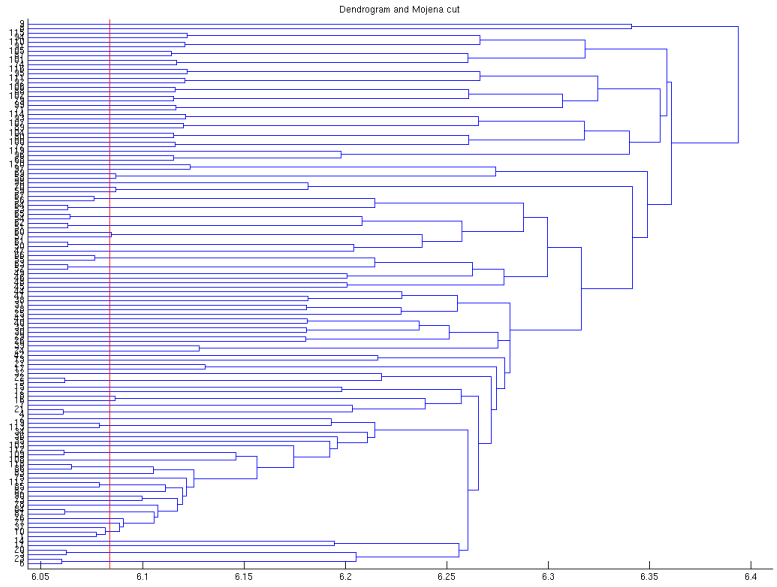
### 4.2.3 Topic Extraction and Labeling

The topic extraction step has been performed with the following hyperparameter values: $\alpha = 50/T$ and $\eta = 0.01$. The number of iterations has been set to $700$. Each run had a burn-in period of $200$ iterations. Each run has been repeated to assure the topic stability according to KL divergence.

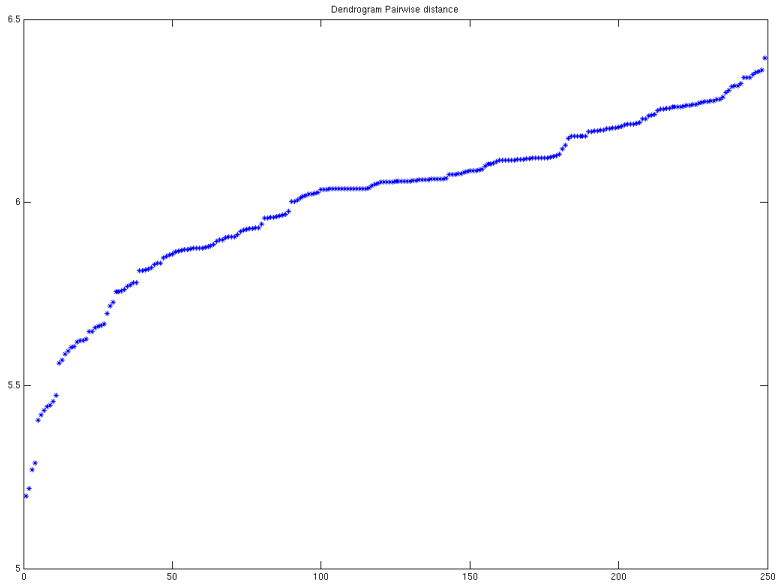
In order to assess the correct number of topics, we started by computing a large number of topics, computing infra-topic similarities according to sKL distance and computing a hierarchical clustering.

Then, we applied the Mojena rule: the value of the parameter has been adapted to the particularities of topic models and the related distances. We extracted two different topic models with number of topics $T = 250$ and $T = 500$. The dendrogram in Figure 4.7(a) and 4.7(b) shows topic similarities according to KL divergence: shallow joins indicate higher similarities between topics. According to the Mojena stopping rule, the optimal number of clusters is $100$. Then we run the LDA model setting the number of topics to $100$ while keeping constant the other parameter values. Some of the extracted topics are shown in Table 4.1.

Then, the ALOT algorithm is applied (Table 4.3) and the labels assigned (Table 4.2). The **Topic_66** is a topic concordance case and it is easy to label. The **Topic_0** and **Topic_15** are associated with a topic discordance case and a semantic association subcase: the former contains labels on the same path while the latter contains labels of the same sub-tree. The **Topic_24** in-




(a) Dendrogram




(b) Pairwise Distance

**Figure 4.7:** Topic Hierarchical Clustering and Mojena cut

| **Topic_66** | **.008** |
|---|---|
| encyclopedia | .023 |
| atlas | .023 |
| bibliography | .017 |
| directories | .016 |
| dictionary | .015 |
| catalog | .014 |
| periodicals | .012 |
| genealogy | .012 |
| abstracts | .012 |
| librarians | .012 |

| **Topic_0** | **.013** |
|---|---|
| cells | .025 |
| protein | .010 |
| genetic | .008 |
| gene | .007 |
| samples | .006 |
| acid | .006 |
| proteins | .005 |
| bone | .005 |
| genes | .005 |
| tissue | .005 |

| **Topic_15** | **.011** |
|---|---|
| firms | .037 |
| suppliers | .012 |
| enterprises | .012 |
| venture | .009 |
| entrepreneurs | .008 |
| productivity | .008 |
| procurement | .007 |
| logistics | .007 |
| supplier | .006 |
| incentives | .006 |

| **Topic_38** | **.013** |
|---|---|
| assets | .022 |
| loan | .018 |
| banks | .018 |
| loans | .017 |
| debt | .014 |
| investments | .013 |
| equity | .012 |
| securities | .011 |
| banking | .011 |
| expenses | .010 |

**Table 4.1:** Extracted Topics

stead, is a non-semantic association subcase, and the labeling is resolved by computing the maximally agreed topic predecessor (**M-dmatp**).

| | | | |
|---|---|---|---|
| **Topic_0** | SA | Path | **Biology** |
| **Topic_15** | SA | Subtree | **Business** |
| **Topic_38** | NSA | M-dmatp | **Financial Services** |
| **Topic_66** | TC | Concordance | **Reference** |

**Table 4.2:** Labeling rules and resulting labels

| **Topic_66** | |
|---|---|
| Tanimoto | root→**Reference** |
| Jaccard | root→**Reference** |
| Dice | root→**Reference** |
| Cosine | root→**Reference** |
| Overlap | root→**Reference** |
| Mutual | root→**Reference** |

| **Topic_0** | |
|---|---|
| Tanimoto | root→Science→**Biology** |
| Jaccard | root→Science→**Biology** |
| Dice | root→Science→**Biology** |
| Cosine | root→Science→**Biology**→Bioinformatics→Online Services |
| Overlap | root→Science→**Biology**→Bioinformatics→Online Services |
| Mutual | root→Science→**Biology**→Bioinformatics→Online Services |

| **Topic_15** | |
|---|---|
| Tanimoto | root→**Business**→Financial Services |
| Jaccard | root→**Business**→Financial Services |
| Dice | root→**Business**→Financial Services |
| Cosine | root→**Business**→Financial Services |
| Overlap | root→**Business**→Business Services→Consulting |
| Mutual | root→**Business**→Business Services→Consulting |

| **Topic_38** | |
|---|---|
| Tanimoto | root→Business→**Financial Services** |
| Jaccard | root→Business→**Financial Services** |
| Dice | root→Business→**Financial Services** |
| Cosine | root→Business→**Financial Services** |
| Overlap | root→Computers→Software→Industry Specific→Insurance |
| Mutual | root→Computers→Software→Industry Specific→Insurance |

**Table 4.3:** Candidate labels according to similarities measures. For each label, the full path is displayed.

### 4.2.4 Document Tagging

The last module of the pipeline is devoted to document tagging. Once the topics have been labeled the system generates the inducer. The system requires the user to specify the model and the number of words used for each topic: this information will generate the global vocabulary associated with the inducer. After the inducer has been built, the user is requested to choose the document representation to use and a probability threshold associated with class labels. These parameters are particularly important whenever the system is used as a component in a bigger system, e.g. document management systems, where for each document a meaningful labeling is requested: a unique labeling with high probability. In [74] has been shown that for the Italian language a probability threshold set to $0.5$ provides stable and reliable results.

To evaluate the performance of the proposed system we downloaded 189 new documents using the same methodology used for the corpus generation, then we manually labeled the documents according to 4 different classes: *Biology*, *Business*, *Financial Services* and *Reference*. Then, the documents were inputted to the inducer with document *term frequency* representation and a threshold value equal to $0.5$. The results are summarized in Table 4.4.

| | **Precision** | **Recall** | **Accuracy** |
|---|---|---|---|
| **Biology** | 1.00 | 0.45 | 0.71 |
| **Business** | 1.00 | 0.35 | 0.80 |
| **Financial Services** | 0.96 | 0.69 | 0.83 |
| **Reference** | 0.88 | 0.13 | 0.61 |

**Table 4.4:** Classification performances.

The precision of the topics *Biology, Business* and *Financial Services* is nearly perfect due to the high specificity of the word lists. The test set is composed of many technical papers which are easily correctly labeled. On the other hand the achieved recall value is consistently lower than precision. A possible explanation to this behavior is as follows; the manual labeling procedure is both complex and ambiguous; it could label documents

by using a broader meaning for each topic. Therefore, it is expected that automatic document classification would not achieve excellent performance with respect to both precision and recall. However, it is important to keep in mind the difficulty of the considered labeling task, together with the fact that human labeling of documents can result in ambiguous and contradictory label assignment. The *Reference* topic has sensibly lower performances due to its particular nature. Indeed, it is not easily captured by the human labeling who tends to assign this topic to many generic documents, and moreover, the documents matching the topic semantics in the test set are too few.

## 4.3 Conclusion and Future Works

Capturing document meaning will be the gold standard in the next few years. Nowadays search engines are very efficient in document retrieving but they still lack the ability to capture the meaning of the retrieved documents according to the user query. The Semantic Web is becoming the corner stone of the future of the web search. However, it is still unable to deal with the enormous amount of existing data and still relies on human intervention for the creation and maintenance of knowledge repositories. The specialized literature has spent a great deal of effort to develop new automatic ways to capture and aggregate bits of information and to create knowledge bases that can be used to satisfy users information needs [25, 121].

Topic extraction models offer an efficient and effective answer to capturing the meaning of document collections. They are particularly useful whenever we have to deal with mid-sized document repositories and when we need an overview on what the documents are about. Future research in topic models will be more and more focused on the integration of topics with network data, like social networks with the aim to help users to identify sub-networks that meet their interest and vision.

In this chapter we proposed an information processing pipeline that exploits a probabilistic topic models, an automatic labeling procedure and a

document tagging service that helps users to efficiently manage and organize their documents. We presented results of applying the pipeline to a real world corpus automatically generated by retrieving documents through random queries. The extracted topics show a high quality and semantic significance. Topic labeling has been performed by exploiting the Google directory, a real and hand crafted taxonomy that tries to classify web pages according to predefined categories. The performances of the document tagger have been evaluated against a set of manually labeled documents and show interesting results. The probabilistic models behind the pipeline exploits Latent Dirichlet Allocation for topic extraction and the classifier implements a particular implementation of a multi-net Naïve Bayes model that automatically maps the topics to the inducer. The pipeline offers many application scenarios: it can be used for document management, implemented in a vertical information retrieval application, for press coverage management service and many others. Recent models can also be applied to improve or characterize the particular application: hierarchical topic models, relational models or author-topic models can offer interesting developments and improved applicability.

# Chapter 5

# Topic Model performance estimation

In this chapter we are concerned with the analysis and validation of the semantic coherence of the results obtained through Latent Dirichlet Allocation and with the problem of their comparison with the results obtained through alternative models. The proposed approach consists of transforming each topic model into a "hard" overlapping partition over documents through the discretization of the "soft" document-topic associations. The Fowlkes-Mallows (FM) index [40,114], a cluster validation metric, has been generalized in order to make it suitable to validate overlapping and incomplete clusterings. Such generalization was performed on the basis of its underlying probabilistic interpretation and allows us to link the Fowlkes-Mallows index to the semantic coherence of the model rather than to the mere similarity between cluster partitions.

Thus, the validation is performed by exploiting this novel probabilistic metrics, based on the interpretations of the widely known *precision* and *recall* performance measures. The proposed validation approach has the following advantages with respect to existing metrics:

- it offers an explicit probabilistic interpretation;
- it allows the validation of overlapping partitions;

- it allows the validation of incomplete partitions.

Moreover, the harmonic mean of precision and recall can be computed to obtain a combined single-metric measurement of the quality of the partition i.e. *F-measure*. The paper also shows how the proposed metrics allow to perform a "drill-down" analysis into the individual topics (clusters) to make straightforward the determination of:

1. which are the best and the worst clusters in the partition;
2. which topics are better recalled by any given cluster.

The rest of this chapter is organized as follows. In section 5.1 the problem of cluster evaluation is presented according to the Fawlkes-Mallows index and its probabilistic interpretation. In section 5.2 The FM index is adapted to incomplete and overlapped partition and the proposed metric to evaluate a topic model quality are introduced. The results of the numerical experiments performed on the Reuters-21578 data set are described in Section 5.3. Finally, conclusions and research directions are reported in Section 5.4.

## 5.1 Clustering Evaluation

Every *hard-clustering* problem applied to a multi labeled document corpus involves the following elements:

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

Most of the existing validation metrics [122] can be expressed in terms of a $|U| \times |C|$ contingency matrix (Table 5.1) 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 uni-labeled, the following properties hold:

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

**Table 5.1:** Contingency matrix

1. $\bigcup_1^K u_i = \mathcal{D}$;
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.

A comprehensive review of the traditional metrics used to validate non overlapping partitions can be found in [122] and [32].

### 5.1.1 The Fowlkes-Mallows index

Among the existing cluster validation metrics, a particular interesting one is the Fowlkes-Mallows (FM) index [40], [114] . Using the contingency matrix notation from Table 5.1, the FM index is defined as follows:

$$FM = \frac{\sum_{ij} \binom{n_{ij}}{2}}{\sqrt{\sum_i \binom{n_{i*}}{2} \sum_j \binom{n_{*j}}{2}}} \tag{5.1.1}$$

In order to analyze the FM index, the events associated with the experiment of randomly sampling two documents $d_1$ and $d_2$ without replacement from $D$, are defined as follows:

- $S_{*c}$: $d_1$ and $d_2$ belong to the same class;
- $S_{u*}$: $d_1$ and $d_2$ belong to the same cluster;
- $S_{uc}$: $d_1$ 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:

$$P(S_{*c_j}) = \frac{\binom{n_{*j}}{2}}{\binom{n_{**}}{2}} \tag{5.1.2}$$

In a similar manner, we write $S_{u_i*}$ to denote that $d_1$ and $d_2$ belong to cluster $u_i$, where the corresponding probability value is given by:

$$P(S_{u_i*}) = \frac{\binom{n_{i*}}{2}}{\binom{n_{**}}{2}} \tag{5.1.3}$$

The probability of two documents belonging to the same class can be computed from expression (5.1.2) to be:

$$P(S_{*c}) = \sum_j P(S_{*c_j}) = \frac{1}{\binom{n_{**}}{2}} \sum_j \binom{n_{*j}}{2} \tag{5.1.4}$$

while the probability of two documents belonging to the same cluster can be computed from expression (5.1.3) to be:

$$P(S_{u*}) = \sum_i P(S_{u_i*}) = \frac{1}{\binom{n_{**}}{2}} \sum_i \binom{n_{i*}}{2} \tag{5.1.5}$$

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

$$P(S_{uc}) = \sum_{ij} P(S_{u_i c_j}) = \frac{1}{\binom{n_{**}}{2}} \sum_{ij} \binom{n_{ij}}{2} \tag{5.1.6}$$

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

$$P(S_{*c}|S_{u*}) = \frac{P(S_{uc})}{P(S_{u*})} = \frac{\sum_{ij} \binom{n_{ij}}{2}}{\sum_i \binom{n_{i*}}{2}} \tag{5.1.7}$$

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

$$P(S_{u*}|S_{*c}) = \frac{P(S_{uc})}{P(S_{*c})} = \frac{\sum_{ij} \binom{n_{ij}}{2}}{\sum_{j} \binom{n_{*j}}{2}} \tag{5.1.8}$$

It is worthwhile to note that the FM index (5.1.1) 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 (5.1.7) and (5.1.8) allow us to write the following:

$$FM = \sqrt{P(S_{*c}|S_{u*})P(S_{u*}|S_{*c})}. \tag{5.1.9}$$

**Hypergeometric Distribution** The previous formulations can also be expressed in terms of the hypergeometric distribution (See A.3). Equation (5.1.3) and (5.1.5) can be rewritten in terms of hypergeometric distribution as follows: the probability of sampling two documents from the same cluster could be rewritten as $P(S_{u*}) = \sum_i h(n_{**}, n_{i*}, 2, 2)$ and the probability of sampling two documents from the same class becomes $P(S_{*c}) = \sum_j h(n_{**}, n_{*j}, 2, 2)$. In a similar fashion, Equation 5.1.6 can be interpreted as $P(S_{uc}) = \sum_{ij} h(n_{**}, n_{ij}, 2, 2)$.

Thus, the conditional probabilities (Eq. 5.1.7 and 5.1.8) expressed above can be rewritten as:

$$P(S_{*c}|S_{u*}) = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sum_i h(n_{**}, n_{i*}, 2, 2)} \tag{5.1.10}$$

and:

$$P(S_{u*}|S_{*c}) = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sum_j h(n_{**}, n_{*j}, 2, 2)} \tag{5.1.11}$$

Finally, by geometric averaging (5.1.10) and (5.1.11) the new expression for

(5.1.1) is:

$$FM = \frac{\sum_{ij} h(n_{**}, n_{ij}, 2, 2)}{\sqrt{\sum_i h(n_{**}, n_{i*}, 2, 2) \sum_j h(n_{**}, n_{*j}, 2, 2)}} \tag{5.1.12}$$

It is worthwile to notice that, when $N$ is greater than 50 and $m/N \leq 0.10$, the hypergeometric distribution can be conveniently approximated by the binomial distribution [19]. Thus the proposed formalization serves two goals. On one hand, it is helpful for computational purposes as it allows the usage of lower cost approximations; on the other hand, it is useful to better understand the properties of the considered metric.

## 5.2 Proposed metrics

In this section, we introduce a version of the FM index adjusted for overlapping and incomplete clusters. In this set-up the properties assumed by the FM index do not hold due to the particular nature of topic models:

- the tresholding 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-labeled, and thus every document can be assigned to no, one or more classes.

### 5.2.1 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 (5.1.1) because the assumption of sampling without replacement does not hold. The main difficulty with overlapping when computing the FM index, is due to the use of the contingency matrix notation, which hides the probability being computed that easily results in making the wrong assumption that

$n_{i*} = |u_i|$, $n_{*j} = |c_j|$ and $n_{**} = |D|$. The implications of such wrong assumptions are shown through the following example.

**Example 2.** *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* $\{d_4, d_5\} \in c_2$ *and let* $u_2 = \{d_6, d_7, d_8, d_9, d_{10}\}$ *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* | $n_{2*} = 5$ |
| | $n_{*1} = 4$ | $n_{*2} = 6$ | $n_{**} = 10$ |

*According to (5.1.2) and (5.1.4) we can compute* $P(S_{*c})$ *as follows;* $P(S_{*c}) = \sum_j P(S_{*c_j}) = \sum_j h(n_{**}, n_{*j}, 2, 2) = \frac{\binom{4}{2}}{\binom{10}{2}} + \frac{\binom{6}{2}}{\binom{10}{2}} = \frac{21}{45}$ *to obtain a correct probability value.*

*Now suppose to extend the example above in a class overlapping scenario, due to multi-labeled documents. 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 (5.1.4) yields the incorrect result of* $31/105$, *which is smaller than the correct one* $21/45$. *This is due to the fact that the sampling without replacement assumption no longer holds. Indeed, there are not* $\binom{15}{2} = 105$ *ways to select 2 documents, as that would allow the possibility to select the same document more than one time. The right number of ways to select 2 elements is still 45 and it is given by* $\binom{|\mathcal{D}|}{2} = \binom{10}{2}$. *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 (5.1.4).*

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:

$$P(S_{*c}) = \sum_j h(|D|, |c_j|, 2, 2) - J(C) \tag{5.2.1}$$

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

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

or accordingly to the hypergeometric notation by the following expression:

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

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 (5.2.1) is a re-expression of (5.1.4) under the general addition rule of probability for non independent events which states that: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

**Example 3.** *In the previous example, if any of the pairs* $\{(d_4, d_8), (d_4, d_9), (d_4, d_{10}), (d_8, d_9), (d_8, d_{10}), (d_9, d_{10})\}$ *are 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:*

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

### 5.2.2 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 marginal totals correspond to the size of the classes, i.e. $n_{*j} = |c_j|$, and that the row marginal totals equal the size of the clusters. 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 incorrectly reduces 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:

$$P(S_{u*}) = \sum_i h(|D|, |u_i|, 2, 2) - J(U) \tag{5.2.2}$$

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 cluster 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 note that formula (5.2.2) is also valid 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.

### 5.2.3 Generalized Fowlkes-Mallows Index

As discussed in section 5.1, if the FM index is expressed in terms of the contingency matrix, it can not be used to validate overlapping or in-

complete clusters. The reason is that while its addition terms come from the hypergeometric distribution, 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 in order to use the cluster size $|u_i|$ and the class size $|c_j|$, the probabilities $P(S_{*c})$ in (5.2.1) and $P(S_{u*})$ in (5.2.2) 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 $P(S_{uc})$ in such a way that non-independent events $S_{u_i c_j}$ are correctly taken into account. This generalization requires to compute the probability of the intersection of elementary events. 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:

$$J(U,C) = \sum_{ij} \sum_{i',j'} h(|D|, |\{S_{u_i c_j} \cap S_{u_{i'} c_{j'}}\}|, 2, 2) \tag{5.2.3}$$

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

$$P(S_{uc}) = \sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U,C) \tag{5.2.4}$$

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.:

$$P(S_{*c}|S_{u*}) = \frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U, C)}{\sum_i h(|D|, |u_i|, 2, 2) - J(U)}; \tag{5.2.5}$$

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

$$P(S_{u*}|S_{*c}) = \frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U, C)}{\sum_j h(|D|, |c_j|, 2, 2) - J(C)} \tag{5.2.6}$$

In conclusion, the generalized version of the FM index, which will be referred to as GFM, is given by[1] :

$$\frac{\sum_{ij} h(|D|, n_{ij}, 2, 2) - J(U, C)}{\sqrt{[\sum_i h(|D|, |u_i|, 2, 2) - J(U)]\,[\sum_j h(|D|, |c_j|, 2, 2) - J(C)]}} \tag{5.2.7}$$

### 5.2.4 Partial Class Match Precision

This metric is inspired by 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:

$$P_{PM} = P(\tilde{S}_{*c}) = \sum_i P(\tilde{S}_{*c}|u_i)P(u_i) \tag{5.2.8}$$

[1]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 more theoretical interest will be presented in future works.

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 in the case of class 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. Thus, each individual $P(\tilde{S}_{*c}|u_i)$ would be given by:

$$P(\tilde{S}_{*c}|u_i) = \sum_j P(\tilde{S}_{*c_j}|u_i) - J(u_i) \tag{5.2.9}$$

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:

$$J(u_i) = \sum_j \sum_{j'>j} P(\{S_{u_i c_j} \cap S_{u_i c_{j'}}\}) \tag{5.2.10}$$

The previous equation represents the probability of selecting two elements from cluster $u_i$ that simultaneously belong to two different classes.

Thus, in order to obtain $J(u_i)$ we need to compute the individual probabilities of selecting two documents that simultaneously belong to every distinct pair of classes $(c_j, c_{j'})$ and then add them up to obtain the probability of selecting two documents that simultaneously belong to any pair of classes.

The expression for the individual probabilities can also be represented using the formula of the hypergeometric distribution, where the parameter accounting for the number of successful outcomes is the number of elements in $u_i$ that belong to both $c_j$ and $c_{j'}$, that is, the "overlap" between $c_j$ and $c_{j'}$.

$$J(u_i) = \sum_j \sum_{j'>j} h(|u_i|, |\{S_{u_i c_j} \cap S_{u_i c_{j'}}\}|, 2, 2) \tag{5.2.11}$$

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 a success. We consider that this property of the metric is a valuable feature to focus on measuring semantic coherence rather than mere partition similarity.

For instance, in contrast to similarity oriented metrics, more than one clustering solution can achieve the maximum evaluation in terms of the PCMP metric. In fact, we can think of two clustering solutions that will obtain a PCMP value of 1, where any pair of elements sampled from within a given cluster will belong to the same class.

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 exactly the same class labels.

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

### 5.2.5 Clustering Recall

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 rely on the values of the contingency matrix to compute recall values, because they can account for duplicates. They also do not consider elements not assigned to any clusters.

**Class recall**

If we are interested in measuring which classes are better captured by the clustering it is straigthforward 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.

$$R(c_j) = P([d \in \cup_i^k u_i]|c_j) = \frac{|\bigcap_i^k \{u_i \cap c_j\}|}{|c_j|} \tag{5.2.12}$$

In other words, equation (5.2.12) means dividing the number of documents labelled with class $c_j$ that were recalled by any cluster $u_i$ by the total number of documents labelled with class $c_j$.

**Gross clustering recall**

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

$$R_U = P(d \in \cup_i^k u_i) = \sum_j P(d \in \cup_i^k u_i|c_j)P(c_j) \tag{5.2.13}$$

where the probability of selecting each class would be given by $|c_j|/N_c$. So (5.2.13), it can be conveniently expressed as:

$$R_U = \frac{1}{N_c} \sum_j R(c_j)|c_j| \tag{5.2.14}$$

### 5.2.6 Single-metric performance

In retrieval and classification it is widely known that it is trivial to achieve high recall at the expense of precision and viceversa. Thus, 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 this way, however, for the particular case of topic-model validation we are interested in balancing the best measurement for semantic coherence with the best measure for completeness, so our proposed metric is defined by the harmonic average of equation (5.2.9) and equation (5.2.14) to obtain:

$$F_o = \frac{2P_{PM}R_U}{P_{PM} + R_U} \tag{5.2.15}$$

Notice that the selection of (5.2.9) and (5.2.14) comes at the expense of not penalizing some clustering dissimilarities. Thus, if the ultimate performance criteria 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 (5.2.9) and (5.2.14) in order to work with unconditional probabilities.

## 5.3 Numerical Experiments

In this section the correctness of the theoretical formulations is checked. Moreover some insights on their different characteristics is provided.

In order to generate the presented performance measures the Reuters-21578 corpus, ModApte split, is adopted and only the documents that are labeled with any topics are considered. The pre-processing step has not filterd the stopwords but numbers were replaced by a unique symbol. After this pre-processing, documents with less than 10 unique words were removed. Both traning and test set were included in the corpus, making a total of 10,468 unique documents and 117 ground-truth classes.

### 5.3.1 Topic Extraction

For demonstration purposes, the evaluated algorithm was LDA with parameters $\beta = 0.01$, $\alpha = 50/K$ running 1,000 iterations of Gibbs sampling. As noted earlier, the proposed measurement techniques require a discretization of the document-topic assignments. Thus, in order to better observe the effects of the discretization on the final mesurement we generated models using document-topic probability thresholds $t$ of $0.05, 0.1, 0.2, 0.25$ and number of topics $K$ of $10, 30, 50, 70, 90$ and $117$. An example of the extracted topics with $K = 90$ is shown in Table 5.2. Each topic is associated with the prior probability $P(z_i)$ and each word is associated with its conditional probability $P(w_i|z_i)$

| **Topic_0** | **.010** |
|---|---|
| coffee | .062 |
| brazil | .040 |
| said | .037 |
| export | .035 |
| quotas | .026 |
| quota | .022 |
| producers | .018 |
| ico | .017 |
| brazilian | .015 |
| international | .015 |

| **Topic_2** | **.008** |
|---|---|
| price | .062 |
| prices | .047 |
| oil | .034 |
| effective | .028 |
| cts | .022 |
| crude | .022 |
| increase | .021 |
| raised | .021 |
| barrel | .020 |
| raises | .019 |

| **Topic_49** | **.013** |
|---|---|
| rate | .095 |
| rates | .077 |
| interest | .055 |
| pct | .048 |
| cut | .035 |
| bank | .027 |
| market | .023 |
| money | .022 |
| prime | .020 |
| point | .016 |

| **Topic_56** | **.011** |
|---|---|
| wheat | .036 |
| agriculture | .034 |
| us | .033 |
| usda | .032 |
| corn | .030 |
| grain | .029 |
| program | .028 |
| farm | .027 |
| said | .024 |
| farmers | .020 |

**Table 5.2:** Example of Extractred topic with K=90.

### 5.3.2 Empirical approximation to the metrics

First, in order to check the correctness of the GFM and $F_o$ formulations, we performed some Monte Carlo simulations. In order to estimate the GFM metric the following procedure, described in Algorithm 2 was performed:

1. Randomly sample a pair of documents.
2. Check if they belong to the same class.
3. Check if they belong to the same cluster.
4. Check if they belong to the same class and cluster.
5. Compute empirical values for $P(S_{uc})$, $P(S_{*c}|S_{u*})$, $P(S_{u*}|S_{*c})$ and $GFM$ (Algorithm 2, lines 16-22).

Then, in order to demonstrate the correctness of the PCMP and $F_o$ formulations, the following simulation, described in algorithm 3, was performed:

1. Randomly select a cluster, based on its prior probability.
2. Randomly select 2 documents from the cluster, check whether if they belong to the same class.
3. Randomly select a class, based on its prior probability, check whether if they are included in the clustering.
4. Compute empirical values for $P(\tilde{S}_{*c}|u_i)$, $R_U$ and $F_o$ (Algorithm 3, lines 20-22).

**Algorithm 2** Approximation to GFM

**Require:** $D = \{d_1, ..., d_N\}$ is the set of input documents and $maxTrials$ is the maximum number of trials.

**Ensure:** $\Gamma$, the empirical approximation to the Generalized Fowlkes-Mallows index.

$ClaSet(d)$ and $CluSet(d)$ return respectively the set of classes and clusters associated with document $d$. $SampleDocsPair(D)$ randomly samples a pair from the set of documents $D$.

$sameClassFreq \leftarrow 0$
$sameClustFreq \leftarrow 0$
$sameClassAndClustFreq \leftarrow 0$
**for** $trials = 1$ to $maxTrials$ **do**
  $sameClass \leftarrow False$
  $sameClust \leftarrow False$
  $d_x, d_y \leftarrow SampleDocsPair(D)$
  **if** $\{ClaSet(d_x) \cap ClaSet(d_y)\} \neq \emptyset$ **then**
    $sameClassFreq \leftarrow sameClassFreq + 1$
    $sameClass \leftarrow True$
  **end if**
  **if** $\{CluSet(d_x) \cap CluSet(d_y)\} \neq \emptyset$ **then**
    $sameClustFreq \leftarrow sameClustFreq + 1$
    $sameClust \leftarrow True$
  **end if**
  **if** $(sameClass \wedge sameClust)$ **then**
    $sameClassAndClustFreq \leftarrow sameClassAndClustFreq + 1$
  **end if**
  $P_{uc} \leftarrow sameClassAndClustFreq/trials$
  $P_{*c} \leftarrow sameClassFreq/trials$
  $P_{u*} \leftarrow sameClustFreq/trials$
  $\Gamma \leftarrow \sqrt{\frac{P_{uc}}{P_{u*}} \cdot \frac{P_{uc}}{P_{*c}}}$
**end for**
**return** $\Gamma$

**Algorithm 3** Approximation to $F_o$

**Require:** $U = \{u_1, ..., u_k\}$ is the set of clusters, $C = \{c_1, ..., c_s\}$ is the set of classes and $maxTrials$ is the maximum number of trials.

**Ensure:** $\Phi_0$, the empirical approximation to the $F_o$ metric.

$ClaSet(d)$ returns the set of classes associated with document $d$. $SampleClass(C)$ randomly samples an element from the set of classes $C$. $SampleClust(U)$ randomly samples an element from the set of cluster $U$. $SampleDocClass(c)$ randomly samples a document associated with the class $c$. $SampleDocsClust(u)$ randomly samples a pair of documents associated with the cluster $u$.

$sameClassGivenClustFreq \leftarrow 0$
$recDocsFreq \leftarrow 0$
$RecalledDocs \leftarrow \emptyset$
**for all** $u_j \in U$ **do**
  $RecalledDocs \leftarrow \{RecalledDocs \cup u_j\}$
**end for**
**for** $trials = 1$ to $maxTrials$ **do**
  $sameClass \leftarrow False$
  $sameClust \leftarrow False$
  $u_x \leftarrow SampleClust(U)$
  $d_x, d_y \leftarrow SampleDocsClust(u_x)$
  **if** $\{ClaSet(d_x) \cap ClaSet(d_y)\} \neq \emptyset$ **then**
    $sameClassGivenClustFreq \leftarrow sameClassGivenClustFreq + 1$
  **end if**
  $c_x \leftarrow SampleClass(C)$
  $d_z \leftarrow SampleDocClass(c_x)$
  **if** $(d_z \in RecalledDocs)$ **then**
    $recDocsFreq \leftarrow recDocsFreq + 1$
  **end if**
  $P_{PM} \leftarrow sameClassGivenClustFreq/trials$
  $R_U \leftarrow recDocsFreq/trials$
  $\Phi_0 \leftarrow \frac{2 \cdot P_{PM} \cdot R_U}{P_{PM} + R_U}$
**end for**
**return** $\Phi_0$

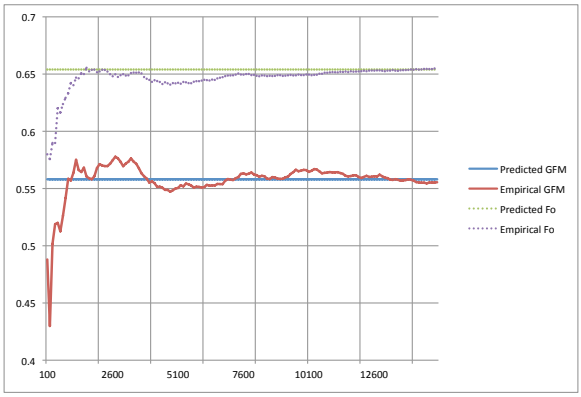



**Figure 5.1:** Monte Carlo approximation of GFM and $F_o$.

Results of an individual simulation for K=90, t=0.2 are shown in Figure 5.1 where the convergence pattern of the empirical measurements to their correct values is depicted.

### 5.3.3 Relation of GFM and Overlapping $F_o$

This subsection is devoted to measure how sensitive the $F_o$, PCMP and GFM metrics are to the document-topic discretization threshold and number of topics parameters. Moreover, a measure of how the presented metrics correlate to each other is presented.

Thus, we consider that it is important to report two statistical measurements. First, in Table 5.3 we present the results of a cross-correlation tab between GFM and the components of $F_o$ for the overall data set. In Table 5.3 it is possible to observe a high correlation between GFM and $F_o$, although not high enough to make the metrics redundant. Recall is positively correlated with $F_o$ and GFM and inversely correlated with Precision; this property is widely acknowledged in the retrieval field.

In order to determine the effects of the parameterization on the measurements we also performed a two-factor analysis of variance using the number of topics K and the selected probability threshold as factors. The results, summarized in Table 5.4 show that both factors, the threshold and the number of topics, have a statistically significant effect on the $F_o$ metric with confidence of above 94%, while this effect can only be moderately no-

| | $F_o$ | GFM | PCMP | Recall |
|---|---|---|---|---|
| $F_o$ | 1 | | | |
| GFM | 0.57 | 1 | | |
| PCMP | 0.91 | 0.33 | 1 | |
| Recall | -0.04 | 0.38 | -0.41 | 1 |

**Table 5.3:** Cross-correlation between metrics

ticed in the GFM metric for the threshold factor with a confidence of about 88%. A potentially important consequence of the $F_o$ metric's higher sensitivity to parametrization is that it makes itself more suitable to perform model selection analysis.

| Factor | $F_o$ | GFM |
|---|---|---|
| Rows (Treshold) | 0.0004* | 0.1267 |
| Columns (K) | 0.0523* | 0.5105 |

**Table 5.4:** Two-Factor ANOVA P-values

## 5.4 Conclusions and Future Work

In this capter we have shown that it is possible to measure the semantic coherence of topic models by considering them to be special instances of soft-clustering algorithms and then using multi-labeled corpora as external validation input. In order to accomplish this goal, we have generalized existing metrics designed to evaluate non-overlapping partitions like the Fowlkes-Mallows Index. We have also proposed metrics with more straightforward probabilistic interpretations and of easier implementation. In both cases we have shown the correctness of the formulations by empirically approximating the predicted values using a Monte Carlo simulation.

In future works we are interested in discussing how the different properties of a topic modeling algorithm like completeness, similarity between partitions or semantic coherence are stressed by the different metrics. We are also interested in evaluate the proposed metrics with other soft clustering methods like the one proposed in [97] and to study possible interplays with probabilistic topic models. Moreover, although this metric is already

based on human input, it would be useful to more clearly visualize the predictive power of such probabilistic metrics on the performance of machine learning tasks like classification or retrieval.

# Chapter 6

# Hybrid search

The interest in semantic web techniques has steadily been growing in recent years. By comparing the standard web with the semantic web one might realize that there are two common but largely distinct ways to represent, store and retrieve information. On the one hand, there are large collections of unstructured text documents. Most web documents are of this form, and also many popular services, such as Wikipedia, are fundamentally document or text based. Information retrieval in this unstructured domain is commonly done with keyword-based search and the results of a query are typically ranked lists of documents.

On the other hand there are structured data sources, from which information can be extracted with formal queries. As structured information sources we consider semantic networks such as DBpedia [4], YAGO [104] or Linked Life Data (LLD) [91]. These sources directly encode entity-relationship (ER) graphs, which consist of entities like persons, countries, etc. and of relations or facts concerning these entities, e.g. statements like `Albert Einstein bornIn Germany`. Moreover, we also consider traditional relational databases, which can be mapped into ER graphs via tools like D2R Server [11] or Openlink Virtuoso [36].

The motivation of this chapter relies on the observation that many information repositories are in fact a combination of the two data-storage regimes described above: they both contain structured and unstructured information. For example consider the textual repository Wikipedia where

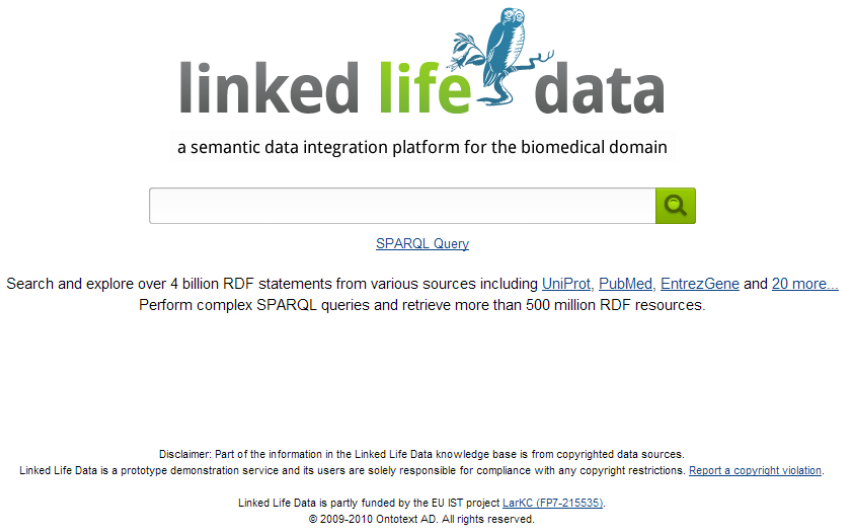



**Figure 6.1:** Linked Life Data

many important facts are available in structured form, e.g. via DBpedia [5] or YAGO. However the structured information is only available for the infoboxes in each wikipage and not for the content of the main article and important content remains hidden in the textual description of each Wikipedia node. In this chapter, both the structured part of Wikipedia and the textual information are exploited. We consider the joint information sources as a textual enriched ER graph. As a second example, consider the computing environment of a large company. There are often huge collections of unstructured textual documents available like emails, project reports, or product handbooks. At the same time, there are typically many well-curated databases available listing and linking entities like employees, hierarchies of departments, customers and products. Available documents can often be linked to one or more entities in the structured representation. In total, one can thus consider the whole information repository again as a textually enriched ER graph.

## 6.1 Information Retrieval on Textually-Enriched ER graph

Keyword-based search [77] is very powerful, since it is flexible, can be implemented efficiently, and is highly intuitive for most users. Yet, keyword-based search also has its well-known problems. High recall is hampered by the fact that there might be many expressions with the same semantic meaning (polysemy). Specificity is negatively influenced by context-dependent semantics of many words (polymorphism). Due to low specificity, a meaningful ranking of the search results is indispensable in keyword-based search. The extraction of knowledge from the retrieved documents is typically up to the user himself. Thus, when searching directly for specific entities, list of entities or facts, traditional search engines are only of limited use.

For structured information repositories information retrieval is typically performed with structured queries in languages like SPARQL [112] for semantic web domains or with SQL for traditional relational databases. These languages allow for very precise query formulation, for efficient filtering and aggregation, such that the search results produce a well-defined set. For structured queries, a ranking of the query results is of less importance.

Querying structured semantic repositories brings two different problems. First, much knowledge is still and will continue to be in text form. While entity extraction and relation extraction have recently made great progress [20, 102, 105], it seems highly likely that, also in the near future, important information will remain in textual, unstructured form.

A second problem is the high complexity of structured queries (e.g. an example queries of [91] is shown in Figure 6.2). So even if all relevant knowledge could be transformed into structured form, many users would still have great difficulties to retrieve this information. For standardized queries an intelligent user interface may automate the formulation of a certain type of query. However, for any non-standard search task the user has to write specific structured search queries which requires deep formal thinking and good knowledge of the structure of the data store.

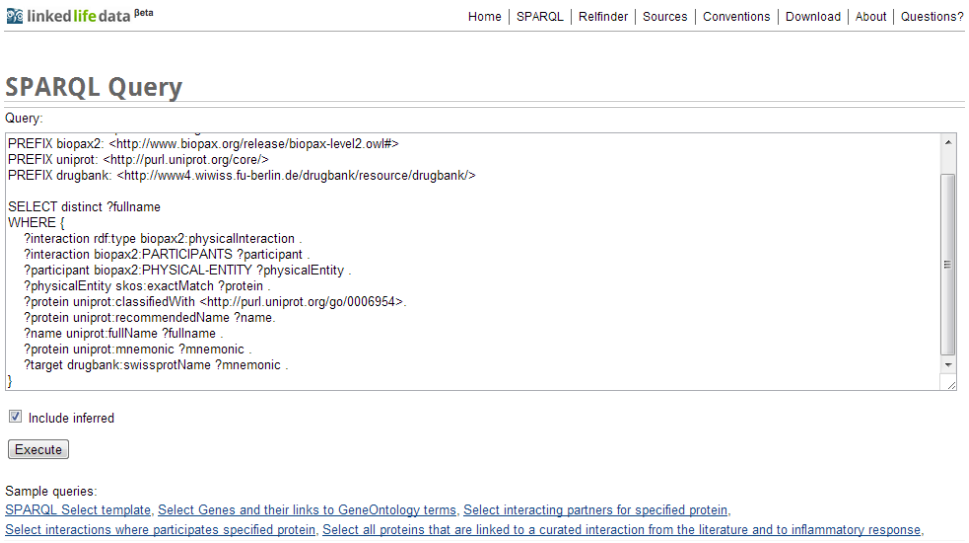



**Figure 6.2:** Example of SPARQL query on Linked Life Data

## 6.2 Proposed Hybrid Search Engine

Given a textually enriched ER graph we show how to formulate hybrid queries consisting of user provided keywords and simple structured queries, which might be encoded in a user interface. Dependent on the query, the results will be ranked lists of entities or lists of facts, that hold between the entities. Technically, we contribute a novel, sound and efficient method to propagate text-based relevance scores on ER graphs to use these for ranking the results of a SPARQL query. While in one setting of our approach, the keywords can be used to rank the results of a classical faceted search, our method is much more flexible and powerful. It realizes approximate keyword matches in the ER graph and also allows for more complex structured queries. The advantages of the proposed approach will be demonstrated in the experimental section.

In information retrieval it is often useful to first expand a query in order to alleviate some of the problems with polysemy and subsequently to narrow down the search results according to the semantics of the query. We implement this basic idea as follows: First, we use a keyword based query on the text-documents in the ER graph and then propagate the keyword

relevance score via a propagation algorithm to the whole ER graph. We then perform a SPARQL query as an effective semantic filter and rank the SPARQL results according to the relevance computed from the keywords.

We mainly follow the terminology of [104]. We consider an *entity* to be an instance of a certain *entity class* such as people, companies, diseases, genes and proteins. Two entities can stand in a *relation*. Facts come in form of triples and are composed of two participating entities and one relation (e. g. `[Angela Merkel, born_in,Hamburg]`). Under *category* we mean the type-level representation of the ER graph resulting from the categories of WordNet or Wikipedia (see [104]).

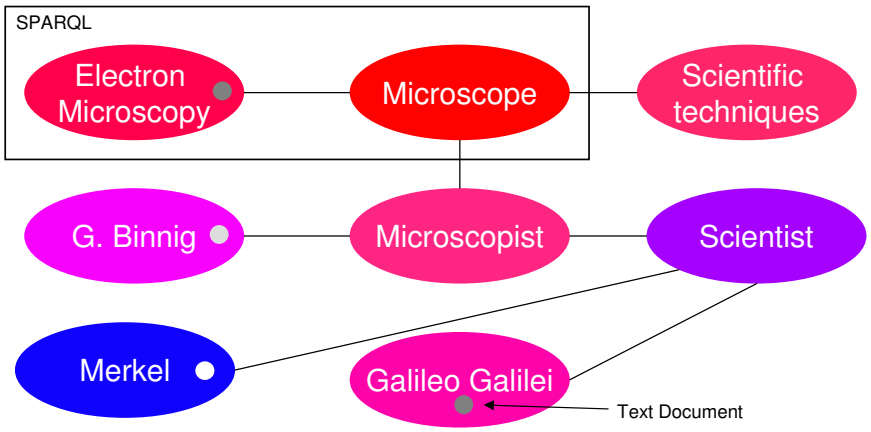



**Figure 6.3:** Stylized subgraph of the YAGO ER graph. Some of the nodes are linked to text documents depicted via circles within the nodes.

**Example 4.** *In Figure 6.3 is represented an qualitative eaxample of the hybrid search. Given a keyword* `microscope`*, a full text query is performed and some documents obtain a relevance score (darker circles mean high relevance). This translates directly into a score for the nodes of the ER graph connected to the document. These scores are generalized to all nodes with and without text with help of the proposed relevance propagation algorithm (the resulting relevances are color-coded, red means high score, blue low score). A SPARQL query finally select a subgraph of the ER graph that contains nodes of type* `category`*. These are ranked according to their aggregated relevance scores.*

### 6.2.1 Keyword Query and Relevance Propagation

Given the ER graph with links to textual documents, we index a node with the words in all the texts associated with that node using Lucene [53]. If a node is connected to more than one document, we join the documents and index the result.

At query time, we can then retrieve all the documents that contain the given keywords or match a given regular expression efficiently. Since the documents are linked to nodes in the ER graph, a query implicitly specifies a subgraph of the ER graph.

With $G = (V, E)$ being the ER graph with $n$ vertices $v_i \in V$ and $m$ edges $(i, j) \in E$, the Lucene query formally yields a Lucene relevance score $l_i$ for the text associated with a node and thus for the node $v_i$ itself. For nodes that are not returned by Lucene we define $l_i = 0$.

The keyword-based node relevances $l_i$ are then generalized via exploiting the known, meaningful structure of the ER graph. Following a page rank like principle [18] or similarly activation spreading ideas [29], we compute novel relevance scores $r_i$ for each node, by iterating

$$r_i = l_i + \lambda \sum_{(j,i)\in E} \frac{r_j}{d_j}.$$

Here, $d_j$ is the out-degree of node $j$ and $0 \leq \lambda \leq 1$ is a weighting factor. Thus, the novel relevance $r_i$ of each node is the Lucene relevance $l_i$ plus contributions of the novel relevances of nodes that are connected via incoming edges. The weighting factor $\lambda$ discounts relevance propagation over long distances. $\lambda$ close to 1 means that propagation distance is not restricted, whereas small $\lambda$ means that only close neighbors in the ER graph get significant activation. The optimal choice of $\lambda$ is task dependent. An example is discussed in the experimental section.

We compute the solution of the resulting sparse linear equation efficiently with an iterative sparse equation solver, namely GMRES [100].Alternatively, this equation could be solved with locally optimal updates which are performed until convergence; this would correspond the Gauss-Seidel method for solving the sparse linear system.

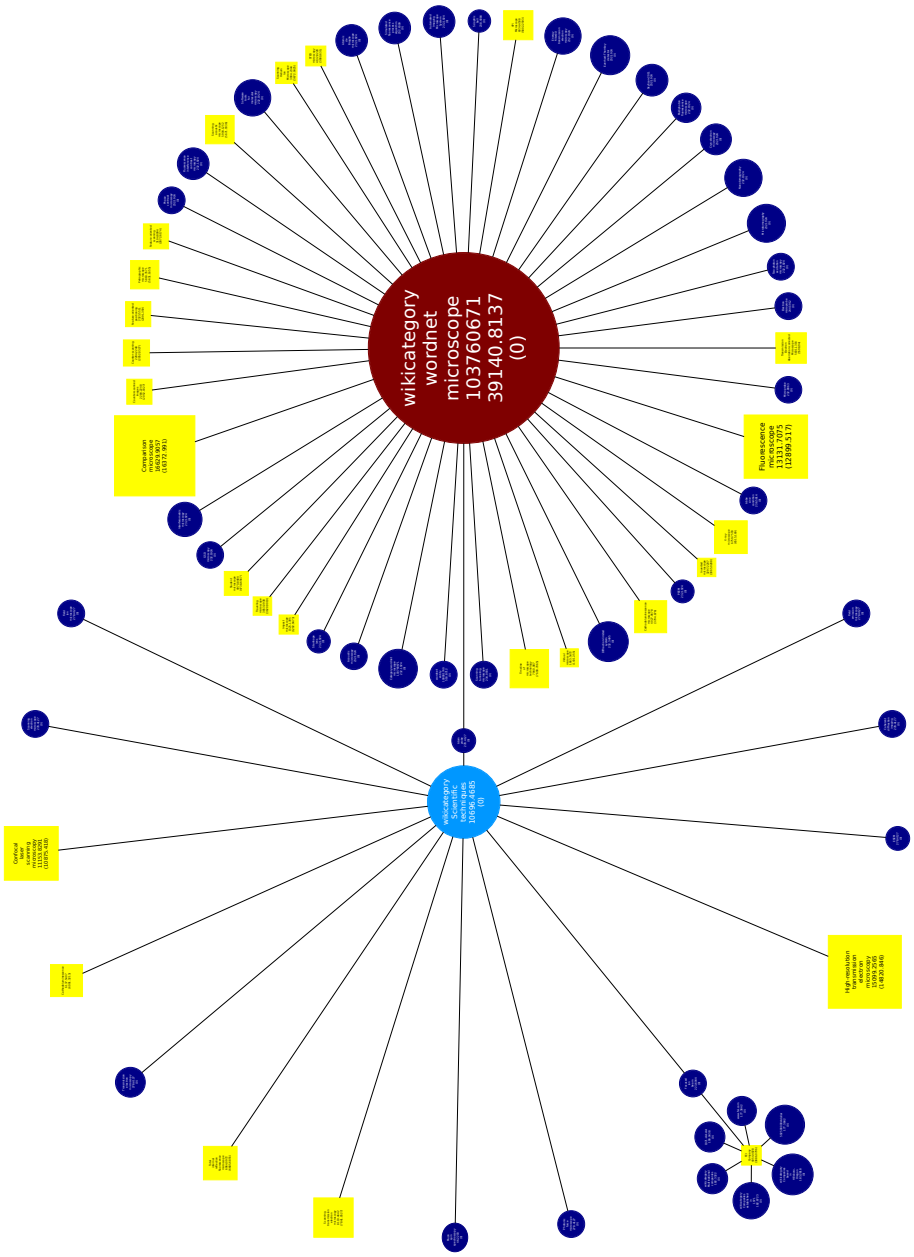



**Figure 6.4:** Spreading Activation Example with query term `microscope`. Larger and red nodes indicate higher relevance, blue indicate lower relevance, yellow squared nodes are textually enriched nodes.

### 6.2.2 Structured Search and Final Result Ranking

In a next step, we filter out relevant entities and facts using a SPARQL-select query. We store the ER graph in a RDF store [92] where we execute the SPARQL query. The result is table where the columns encode the different variable bindings and the rows are instances found in the database, generally not sorted. Each row thus consists of one or more references to entities or literals, and potentially expresses one or more facts about these objects.

Having performed the keyword-based scoring of all the nodes in the ER graph, we now rank the rows returned by the SPARQL query such that more relevant entities or facts are placed on top of the result list. At the moment, we simply sum-up the relevances of all entities in each row. This roughly expresses an OR semantics for ranking. In the future, we will also investigate other combination rules.

## 6.3 Experiments

In order to evaluate the proposed approach we selected the YAGO knowledge base[1] [104] as a structured information source and Wikipedia as a textual resource. YAGO consists of approx. 2 million entities and more than 20 million facts describing these entities [104]. Facts are automatically extracted from Wikipedia and combined with concepts from Wordnet [39]. YAGO's accuracy is estimated to be about 95%. In addition, many YAGO entities are linked to Wikipedia pages via the relation `describes` and we thus obtain an interlinked, textually enriched ER graph.

The full text of all Wikipedia pages is indexed with Lucene. We adapted the standard Lucene score by adding a normalization factor that takes into account the length of the document. This normalization factor is necessary due to the nature of encyclopedic-style document collections, where important or famous entities tend to have longer than average textual descriptions.

In the following we present three show cases. We focus on examples,

[1] version from February 1st 2010

where it is either difficult or impossible to retrieve particular information with a standard SPARQL query or a keyword search alone. Qualitatively, the examples can be grouped as follows:

- *Context-Aware Fact Search:* Search for entities and facts by specifying the interesting aspects with keywords (see **Example 5 and 6**).
- *Context-Aware Category Search:* Search with keywords for abstract categories which are not linked to a special textual description (see **Example 7**).

### 6.3.1 Context-Aware Entity Search

In this setting we would like to retrieve entities or facts about entities where the specific context is specified via keywords.

**Example 5.** *Give me companies with number of employees and annual revenue which have sth. to do with ultrasound.*
*In this example, we query our hybrid search engine with the keyword "ultrasound" and a SPARQL-select with the following WHERE-clause,*

```
   ?company rdf:type wordnet_company
?company yago:hasEmployeees  ?employees
   ?company yago:hasRevenue ?revenue
```

*The results of our approach are presented in Table 6.1. The 10 top-ranked results are all companies that produce ultrasound devices. An exception here is Turtle Beach Syst., which produces a PC sound card named* `Ultrasound`.

Note that there is no category for companies that produce ultrasound devices in YAGO. Thus, this question could not have been answered with a structured query on YAGO alone, which could have only retrieved general companies. At the same time, a keyword-only query on Wikipedia with the keywords "company ultrasound" produces only two companies while the other returned items are related to technology pages. This example thus demonstrates the need for hybrid search techniques.

| r | Company | Employees | Revenue | Score |
|---|---|---|---|---|
| 0 | General Electric | 327000 | $ 172.738 M. | 9083.89 |
| 1 | GE Healthcare | | | 8419.13 |
| 2 | Philips | 125500 | 26.976 M. | 8404.8 |
| 3 | Siemens AG | 430000 | $ 110,820 M. | 8092.45 |
| 4 | Neusoft Group | 12000 | $ 355 M. | 4640.18 |
| 5 | SRI International | | | 4299.93 |
| 6 | Agfa-Gevaert | 13565 | âĆň 3.300 M. | 3759.03 |
| 7 | Foster-Miller | | | 3055.97 |
| 8 | Ellex Medical Las. | | | 3011.97 |
| 9 | Turtle Beach Syst. | | | 2958.37 |

**Table 6.1:** Ranked results of our approach for example 5 (keyword "ultrasound").

Moreover, note that our approach directly returns not only relevant company names, but also their number of employees and the revenue. This approach thus goes beyond keyword-aware faceted search.

**Example 6.** *Give me physicists and their advisors that worked in the area of quantum mechanics and have sth. to do with Los Alamos.*
*The results of our hybrid approach are shown in Table 6.2. Again the result is very reasonable. The first result, Robert Oppenheimer, was the director of the famous Los Alamos Scientific Laboratory, and both Oppenheimer and his advisor Born worked in the area of quantum mechanics. But also the other persons obtained with our approach were famous quantum physicists with connections to Los Alamos laboratories.*

### 6.3.2 Context-Aware Category Search

Relevance propagation allows us to generalize keyword relevance of a subset of nodes to all nodes in the ER graph, even if some of them are not linked to texts themselves. Such nodes are for example the category nodes in YAGO, which group a number of entities with respect to a specific topic or property, but which do not have a description other than the title of the category. With the help of relevance propagation, our system is thus nevertheless able to search for categories by specifying context related keywords.

| r | Physicist | Advisor | Score $r_i$ |
|---|---|---|---|
| 0 | Robert Oppenheimer | Max Born | 86579.94 |
| 1 | David Bohm | Robert Oppenheimer | 77108.46 |
| 2 | Willis Lamb | Robert Oppenheimer | 62835.95 |
| 3 | Philip Morrison | Robert Oppenheimer | 61497.02 |
| 4 | Richard Feynman | John Archibald Wheeler | 54672.82 |
| 5 | George Zweig | Richard Feynman | 52808.75 |
| 6 | Chen Ning Yang | Edward Teller | 47098.77 |
| 7 | Edward Teller | Werner Heisenberg | 46347.56 |
| 8 | Lincoln Wolfenstein | Edward Teller | 45903.60 |
| 9 | John von Neumann | Leopold FejÃl'r | 34172.28 |
| 10 | Emilio G. Segre' | Enrico Fermi | 31561.64 |
| 10 | Enrico Fermi | Luigi Puccianti | 31561.64 |

**Table 6.2:** Ranked results of our approach for example 6 (keywords "Los Alamos" and "Quantum")

**Example 7.** *Give me categories related to microscopes.*
*Here, we query our search engine with the keyword "microscope" and a SPARQL-select with WHERE-clause,*

```
?category rdf:type ?concept
```

*To give a notion about the influence of the weighting factor $\lambda$, we assess the distance of the shortest path D between the nodes that obtained a Lucene score and the nodes from the top ten result set. Table 6.3 shows the results and corresponding distances for $\lambda = 1$ (respectively Table 6.4 for $\lambda = 0.1$). It can be seen that for $\lambda = 1$, the top ten results are enriched with nodes that have a longer shortest path distance than the top ten results for $\lambda = 0.1$.*

*In both cases the results are highly plausible. They can be grouped roughly into two types: Categories that get investigated with the help of microscope techniques, e. g. plants or pathogens, and categories that are technically related to the concept microscope, e. g. X-rays.*

| r | Category | score | D |
|---|---|---|---|
| 0 | wikicategory wordnet microscope | 1234643.5 | 3 |
| 1 | wikicategory Scientific techniques | 1112785.29 | 3 |
| 2 | wordnet disease | 962138.78 | 1 |
| 3 | wordnet person | 632517.98 | 1 |
| 4 | wikicategory wordnet optics | 521473.51 | 3 |
| 5 | wikicategory Plant pathogens and diseases | 486216.12 | 3 |
| 6 | wikicategory wordnet measuring instruments | 322737.59 | 3 |
| 7 | wikicategory X-rays | 281627.11 | 3 |
| 8 | wordnet anatomy | 280101.66 | 1 |
| 9 | wikicategory wordnet igneous rock | 230442.75 | 3 |

**Table 6.3:** Results of our system for example 7 (keyword "microscope", $\lambda = 1$)

| r | Category | score | D |
|---|---|---|---|
| 0 | wikicategory wordnet microscope | 22079.62 | 3 |
| 1 | wikicategory Scientific techniques | 11873.37 | 3 |
| 2 | wordnet disease | 7419.3 | 1 |
| 3 | wikicategory wordnet optics | 6621.56 | 3 |
| 4 | wordnet person | 4254.36 | 1 |
| 5 | wordnet scientist | 4056.44 | 1 |
| 6 | wikicategory wordnet lens | 3426.64 | 3 |
| 7 | wikicategory Types of cancer | 2964.22 | 1 |
| 8 | wikicategory Technology timelines | 2849.33 | 1 |
| 9 | wikicategory wordnet genetics | 2477.66 | 1 |

**Table 6.4:** Results of our system for example 7 (keyword "microscope", $\lambda = 0.1$)

## 6.4 Related Work

Unlike our proposed approach, most semantic search engines aim at retrieving relevant documents supported by semantic annotations, see [76] for a recent survey. Our goal, however, is to directly retrieve entities and facts from the ER graph, such that the user does not have to search through the documents himself.

This paradigm is similar to structured search engines such as NAGA [61], where a flexible subgraph pattern is given to retrieve pieces of an ER graph. While they also rank the results of a structured query, they do not

take into account keywords.

Searching and ranking entities given a keyword query is done for the scientific domain by [89]. For instance, the Libra system [81] returns lists of conferences, persons and research papers. However, this system does not return facts and no generalization by relevance propagation is utilized.

Another attempt to bridge the gap between textual input and structured search is the translation of natural language or keyword queries into a structured formalism. However, this involves advanced understanding of language, a highly ambitious effort. To alleviate this problem [110] introduce a method that maps keyword queries to entities in a knowledge base.

We also refer to work in the traditional database community that provides mechanisms to make Relational Database Management Systems (RDBMS) searchable with keywords, see e. g. [1, 10]. In contrast, our aim is to use the effectiveness and richness of textual features to *rerank* the formal query and thus to narrow down the user intention.

Most similar to our approach is the work proposed by [98]. In their work the authors describe a hybrid approach for searching ER-graph like data repositories. As we do in our work, they also use the idea relevance propagation or activation spreading. Our work can be seen as an extension of this work by including a reranking component based on the user's intention expressed via keywords. More precisely, activation spreading [29] does not include IR-style ranking scores when retrieving relevant results.

## 6.5 Conclusions

We have presented a framework for hybrid search on textually enriched ER graphs. It integrates flexible and intuitive keyword search with the specificity of structured query languages. This is advantageous in several respects: First, keywords are very flexible and allow the incorporation of unstructured information that is not made explicit in the semantic structure of the ER graph. At the same time the graph structure is exploited both for a suitable generalization of keyword relevance via relevance propagation, as well as for a precise filtering in terms a given SPARQL query. Our

method can not only retrieve lists of entities based on a hybrid query, but gives direct access to lists of facts that otherwise would have to be collected manually starting from each relevant entity.

There are several directions to extend and improve our work. While preliminary results have shown the effectiveness of our proposed approach, a quantitative evaluation in terms of retrieval performance will be a matter of ongoing research. Second, we have so far not used formal reasoning in the ER graph. Such additional, logic-based generalization would have to be compared with the more probabilistic generalization step that we have included via the relevance propagation formalism. We imagine that these two approaches could well complement each other.

In total, we feel that, while information retrieval in the structured and in the unstructured domain separately are well-established, searching and retrieving information on mixed data sources such as textually enriched entity relationship graphs will be a growing field of interest. There are many more interesting problems that can be looked at such as, for example, clustering or hierarchical question answering. With the advent of linked data stores like Linked Life Data or YAGO/Wikipedia there are many large and interesting data sources available for experiments with such tasks. While we have not demonstrated our method on corporate intranet data, we believe that this area could also be a major application field for the proposed method.

# Chapter 7

# Conclusions

In this dissertation we have shown how Text Mining can be fruitfully exploited in real world applications with the aim to distill knowledge from unstructured textual sources.

We have shown how structured data, commonly produced in a business environments (i.e. taxonomies, databases and semantic repositories) could be enhanced with unstructured document collection to provide powerful tools for knowledge management and information integration.

Thus, the aim of this dissertation has been the development of text mining models that can be easily integrated in business environments. The models provide efficient tools for gaining competitive advantage and easing the access to relevant informations.

More precisely, we have applied probabilistic topic models for document management: the proposed approach exploits a document repository and a related taxonomy by efficiently extracting a representative set of topics labeled accordingly to the supplied taxonomy. The result is a a simple, fast and efficient multi-label Bayesian classifier specifically tailored to document tagging. The system can be easily integrated in a existing document management system and offers a reliable way to manage and retrieve informations.
Such model has also been applied for the Italian legal domain as a vertical faceted search engine, and offers to lawyers the possibility of querying

existing document repositories, receive updates on recent relevant news and manage their own documents. A prototype of the system is publicly available online at `http://dedalus.cnds.disco.unimib.it:8080/sideinformer/`.

Moreover we proposed a novel approach for topic models performance estimation by exploiting a soft-clustering interpretation. In particular we derived a probabilistic interpretation of a well known performance index, that has been exploited for the construction of a novel index that is able to evaluate soft-clustering algorithms characterized by incomplete and overlapping partitions.
The proposed generalization allowed the formulation of novel metrics correlated with the classical information retrieval measures of precision and recall. Furthermore a montecarlo procedure has been proposed in order to offer an efficient and reliable estimation of the model performances.

Finally we have shown how text mining techniques can be used to enrich existing semantic repositories with textual resources. The proposed approach allows to query the semantic repositories via simple keyword-based query. The keyword-based query is extended to the semantic graph by means of a novel spreading activation algorithm enabling the users to discover entities, facts and concept otherwise hidden.

## Future directions

Future work direction should investigate further extension of the proposed models. In particular the use of richer topic models integrating additional information like authors or relations could be useful for document management.The topic labeling algorithm could be enhanced by the application of different rules over semantic graphs; the tagging process could be improved by new classification models.

The performance estimation metrics could be improved by the definition of an upper bound of the error in the evaluation of high overlapped and incomplete partitions. Different topic models could be investigated and compared according to the proposed metrics.

Finally the hybrid search model can be extended by improving the indexing of textual resources with topic models. We are also interested in investigating the applicability of the proposed paradigm for the query answering problem.

# Appendix

# Appendix A

# Probability Distributions

## A.1 Dirichlet Distribution

The Dirichlet is the conjugate prior distribution for the parameters of the multinomial distribution. The Dirichlet is a multivariate generalization of the beta distribution. Let $\mathbf{p}$ denote a $k$ dimensional random vector. Under the dirichlet model with parameter vector $\alpha$ the probability density at $\mathbf{p}$ is

$$\begin{aligned} p(\mathbf{p}) \sim \mathcal{D}(\alpha_1, \cdots, \alpha_k) = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k p_k^{\alpha_k - 1} \\ \text{where } p_k > 0 \\ \sum_k p_k = 1 \end{aligned} \tag{A.1.1}$$

## A.2 Multinomial Distribution

Let $X_1, X_2, \cdots, X_n$ be a set of random variables. The probability mass function is given as follows:

$$
\begin{aligned}
&P(X_1 = x_1, \cdots, X_n = x_n) = \frac{N}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} \theta_i^{x_i} \\
&s.t.\ x_i > 0,\ \theta_i > 0 \\
&\sum_{i=1}^{n} x_i = N \\
&\sum_{i=1}^{n} \theta_i = 1
\end{aligned}
\tag{A.2.1}
$$

## A.3 Hypergeometric Distribution

Hypergeometric Distribution is defined as the probability of selecting exactly $x$ successes in a sample of size $m$, obtained without replacement from a population of $N$ objects from which $k$ contain the characteristic of interest. Formally:

$$
h(N, k, m, x) = \frac{\binom{k}{x}\binom{N-k}{m-x}}{\binom{N}{m}} \tag{A.3.1}
$$

# Bibliography


[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, page 5. Published by the IEEE Computer Society, 2002.

[2] David J. Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII 1983 - Lecture Notes in Mathematics*, 1117:1–198, 1985.

[3] Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics*, 2(6):1152–1174, 1974.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *Lecture Notes in Computer Science*, 4825:722–735, 2007.

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pages 722–735, November 2008.

[6] Ricardo Baeza-Yates and Prabhakar Raghavan. Next Generation Web Search. *Search Computing: Lecture Notes in Computer Science*, 5950:11–23, 2010.

[7] Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards Semantic Web Mining. *Lecture Notes In Computer Science*, 2342:264–278, 2002.

[8] T. Berners-Lee, T. Hendler, and J. Lassila. The Semantic Web. *Scientific American*, 284:34–43, 2001.

[9] Michael W. Berry and Malu Castellanos. *Survey of text mining II: clustering, classification, and retrieval, Volume 2*. Springer, 2007.

[10] Gaurav Bhalotia, Charuta Nakhe, Arvind Hulgeri, Soumen Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *Proceedings of the 18th International Conference on Data Engineering*, 2002.

[11] C. Bizer and R. Cyganiak. D2R Server - publishing relational databases on the semantic web. In *Proceedings of the 5th International Semantic Web Conference (ISWC)*, 2006.

[12] David M. Blei. *Probabilistic models of text and images*. PhD thesis, 2004.

[13] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):1–30, 2010.

[14] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems*, 16, 2004.

[15] David M. Blei and J.D. Lafferty. Visualizing Topics with Multi-Word Expressions. *arXiv:0907.1013v1 [stat.ML]*, 2009.

[16] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] D. Brickley and R.V. Guha. Resource description framework (rdf) schema specification 1.0. Technical report, World Wide Web Consortium Candidate Recommendation, April 2002.

[18] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[19] HD Brunk, JE Holstein, and Frederick Williams. A comparison of binomial approximations to the hypergeometric distribution. *American Statistician*, 22(1):24, February 1968.

[20] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, 9(1):207, 2008.

[21] Markus Bundschus, Volker Tresp, and HP Kriegel. Topic models for semantically annotated document collections. In *NIPS workshop: Applications for Topic Models: Text and Beyond*, pages 1–4, 2009.

[22] Markus Bundschus, Shipeng Yu, Volker Tresp, Achim Rettinger, and M. Hierarchical bayesian models for collaborative tagging systems. *2009 Ninth IEEE International Conference on Data Mining*, pages 728–733, December 2009.

[23] Bob Carpenter. Integrating Out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling, 2010.

[24] George Casella and EI George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[25] Soumen Chakrabarti, S. Sarawagi, and S. Sudarshan. Enhancing Search with Structure. *IEEE Data Engineering Bullettin*, 33(1):1–13, 2010.

[26] Jonathan Chang, J. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, pages 1–9. Citeseer, 2009.

[27] S Chib. Marginal Likelihood from the Gibbs Output. *Journal of the American Statistical Association*, 90(432):1313– 1321, 1995.

[28] David Cohn and Thomas Hofmann. The missing link-a probabilistic model of document content and hypertext connectivity. *Advances in neural information processing systems*, pages 430–436, 2001.

[29] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

[30] Criminal Law. http://www.criminal-law-lawyer-source.com/terms.html, 2010.

[31] A.P. Dempster, N.M. Laird, D.B. Rubin, and Others. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[32] Lucile Denoeud and A. Guénoche. Comparison of distance indices between partitions. *Data Science and Classification*, (1981):21–28, 2006.

[33] Yi-qun Ding, Shan-ping Li, Zhen Zhang, and Bin Shen. Hierarchical topic modeling with nested hierarchical Dirichlet process *. *Journal of Zhejiang University SCIENCE A*, 10(6):858–867, 2009.

[34] Dmoz. http://www.google.com/dirhp, 2010.

[35] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, 2001.

[36] O. Erling and I. Mikhailov. RDF Support in the Virtuoso DBMS. In *Proceedings of the 1st Conference on Social Semantic Web*, volume 113, pages 1617–5468. Springer, 2009.

[37] EU-FP7. LarKC: The Large Knowledge Collider, 2010.

[38] Ronen Feldman and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.

[39] C. Fellbaum. *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[40] EB Fowlkes and CL Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553– 569, 1983.

[41] John Gantz and David Reinsel. The digital universe decade - Are you ready?, 2010.

[42] Gartner. http://www.gartner.com/technology/media-products/reprints/microsoft/vol10/article3/article3.html, 2009.

[43] Alan E Gelfand and Adrian F M Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85(410):398– 409, 1990.

[44] Andrew Gelman. *Bayesian data analysis*. Chapman & Hall/CRC, 2004.

[45] Thomas L. Griffiths, M. Steyvers, David M. Blei, and Joshua B Tenenbaum. Integrating topics and syntax. *Advances in neural Information Processing Systems (NIPS)*, 17:537–544, 2005.

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

[47] Thomas L. Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in Semantic Representation. *Psychological Review*, 114(2):211–244, 2007.

[48] Sergio Guadarrama and Marta Garrido. Concept-Analyzer: A tool for analyzing fuzzy concepts. In *Proceedings of IPMU*, pages 1084–1089, 2008.

[49] N. Guarino. Formal Ontology and Information Systems. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, pages 3– 15. IOS Press, June 1998.

[50] N. Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars, editor, *2nd International Conference on building and sharing very large-scale knowledge bases (KBKS)*, pages 25–32. IOS press, April 1995.

[51] Alon Halevy, Peter Norvig, and Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2):8–12, March 2009.

[52] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[53] Erik Hatcher, Michael McCandless, Otis Gospodnetic, and Mike Mccandless. *Lucene in Action*. Manning Publications, 2010.

[54] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.

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

[56] Andreas Hotho, Steffen Staab, and Gerd Stumme. Ontologies improve text document clustering. In *Third IEEE International Conference on Data Mining (ICDM'03)*, pages 541–545. IEEE Comput. Soc, 2003.

[57] Thorsten Joachims. *Learning to classify text using support vector machines*. Kluwer Academic Publishers, Norwell (MA), 2002.

[58] SC Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[59] Michael I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155, 2004.

[60] Michael I. Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

[61] G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. Naga: Searching and ranking knowledge. In *24th IEEE International Conference on Data Engineering (ICDE 2008)*, pages 1285–1288, 2008.

[62] Sang-bum Kim, Kyoung-soo Han, Hae-chang Rim, and Sung Hyon Myaeng. Some Effective Techniques for Naive Bayes Text Classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466, 2006.

[63] Ross Kindermann and J Laurie Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[64] Thomas K. Landauer, Peter Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2):259–284, 1998.

[65] O. Lassila and R. Swick, editors. *Resource Description Framework (RDF) Model and Syntax Specification*, Boston, 1999. World Wide Web Consortium.

[66] Wei Li. *Pachinko Allocation : Dag-Structured Mixture Models Of Topic Correlations*. PhD thesis, 2007.

[67] Wei Li, David M. Blei, and Andrew Mccallum. Nonparametric Bayes pachinko allocation. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 633—-640, 2007.

[68] Wei Li and A McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577—-584, 2006.

[69] Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on information theory*, 37(I):145–151, 1991.

[70] J Macqueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium*, 233(233):281–297, 1967.

[71] Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic Labeling Of Topics. In *Proceedings of 2009 IEEE International Conferences on Intelligent Systems Design and Applications*, pages 1227—-1232, 2009.

[72] Davide Magatti, Florian Steinke, Markus Bundschus, and Volker Tresp. Combined Structured and Keyword-Based Search in Textually Enriched Entity-Relationship Graphs. In *AKBC2010 First Workshop on Automated Knowledge Building*, 2010.

[73] Davide Magatti and Fabio Stella. *Probabilistic Topic Discovery and Automatic Document Tagging*, pages –. IGI Global, 2011.

[74] Davide Magatti, Fabio Stella, and Marco Faini. A Software System for Topic Extraction and Document Classification. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, 2009.*, volume 1, pages 283–286. IEEE, 2009.

[75] Alessia Mammone, Marco Turchi, and Nello Cristianini. *Analysis of Text Patterns Using Kernel Methods*. Taylor and Francis, 2009.

[76] C. Mangold. A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34, 2007.

[77] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[78] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[79] Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. Automatic Labeling of Multinomial Topic Models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499, 2007.

[80] Mesh. http://www.nlm.nih.gov/mesh/trees2008.html, 2008.

[81] Microsoft. http://academic.research.microsoft.com/, 2010.

[82] Glenn W. Milligan and Martha C. Cooper. An examination procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[83] David Mimno and Andrew Mccallum. Organizing the OCA: Learning faceted subjects from a library of digital books. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 376–385, 2007.

[84] Riichiro Mizoguchi. Tutorial on ontological engineering: part 3. *New Generation Computing*, 22(2):198, 2004.

[85] R. Mojena. Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal*, 20(4):359–363, 1977.

[86] Radford M. Neal. Annealed importance sampling, 2001.

[87] Richard E Neapolitan. *Learning Bayesian Networks*, volume 43 of *Artificial Intelligence*. Prentice Hall, 2003.

[88] David Newman, C Chemudugunta, and P. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, 2006.

[89] Z. Nie, Y. Zhang, J.R. Wen, and W.Y. Ma. Object-level ranking: Bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, page 574. ACM, 2005.

[90] Kamal Paul Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, 2001.

[91] Ontotext. Linked life data. `http://www.linkedlifedata.com/`, year = 2009.

[92] Ontotext. OWLIM, 2009.

[93] J Pitman. *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, 2006.

[94] A Popescul, LH Ungar, DM Pennock, and S. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. *Proceedings of the 17th Conference on Uncertainity in Articfical Intelligence UAI2001*, pages 437—-444, 2001.

[95] Eduardo .H. Ramirez, Ramon Brena, Davide Magatti, and Fabio Stella. Probabilistic Metrics for Soft-Clustering and Topic Model Validation. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 406–412, 2010.

[96] Eduardo H. Ramirez, Ramon Brena, Davide Magatti, and Fabio Stella. Topic Model Validation. *Neurocomputing. Special Issue on "Advances in Web Intelligence" - Invited paper - Submitted*, 2011.

[97] E.H. Ramirez and R.F. Brena. 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*, pages 331–334. IEEE Computer Society, 2009.

[98] C. Rocha, D. Schwabe, and M.P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM New York, NY, USA, 2004.

[99] M. Rosen-Zvi, C. Chemudugunta, and Thomas L. Griffiths. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):1–38, 2010.

[100] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[101] G Salton. Developments in automatic text retrieval. *Science (New York, N.Y.)*, 253(5023):974–80, August 1991.

[102] Sunita Sarawagi. Information extraction. *Found. Trends databases*, 1(3):261–377, 2008.

[103] T. Steyvers, M. and Griffiths. *Probabilistic topic models*, chapter Probabilis. Lawrence e edition, 2007.

[104] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA, 2007. ACM Press.

[105] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. Sofie: a self-organizing framework for information extraction. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 631–640, New York, NY, USA, 2009. ACM.

[106] C.M. Sperberg-McQueen T. Bray, J. Paoli and E. Maler, editors. *Extensible Markup Language (XML) 1.0 (Second Edition)*. World Wide Web Consortium., October 2000.

[107] PP Talukdar, Marie Jacob, MS Mehmood, and K. Learning to create data-integrating queries. In *Proceedings of the VLDB Endowment*, volume 1, pages 785–796, 2008.

[108] Yee Whye Teh, Michael I. Jordan, Matthew J Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[109] The Economist. Data, data everywhere..., August 2010.

[110] T. Tran, P. Cimiano, S. Rudolph, and R. Studer. Ontology-based interpretation of keywords for semantic search. *Lecture Notes in Computer Science*, 4825:523, 2007.

[111] Volker Tresp, Markus Bundschus, Achim Rettinger, and Yi Huang. Towards Machine Learning on the Semantic Web. *Lecture Notes In Artificial Intelligence*, pages 282–314, 2008.

[112] W3C. SPARQL Query Language for RDF, 2007. `http://www.w3.org/TR/rdf-sparql-query/`.

[113] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1):1–305, 2007.

[114] David L. Wallace. A method for comparing two hierarchical clusterings: comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

[115] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, number 1, pages 977–984. ACM, 2006.

[116] Hanna M Wallach, Iain Murray, and Ruslan Salakhutdinov. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*, number d, pages 1–8, 2009.

[117] HM Wallach. *Structured topic models for language*. PhD thesis, 2008.

[118] Bo Wang and DM Titterington. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. In *Proceedings of the 20th conference in Uncertainity in Artificial Intelligence*, pages 577–584, 2004.

[119] Wei Wang, P Mamaani Barnaghi, and A. Bargiela. Probabilistic topic models for learning terminological ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):1028–1040, 2010.

[120] X. Wang, Andrew Mccallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE International Conference on Data Mining (ICDM2007).*, pages 697–702. IEEE, 2008.

[121] Gerhard Weikum and Martin Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems of data*, pages 65–76. ACM, 2010.

[122] Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for K-means clustering. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 877, 2009.

[123] Elias Zavitsanos, Georgios Paliouras, George A Vouros, and Sergios Petridis. Learning subsumption hierarchies of ontology concepts from texts. *Web Intelligence and Agent Systems: An International Journal*, 8:37–51, 2010.