SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of

ECONOMICS, MANAGEMENT, AND STATISTICS

Ph.D. program: **Economics and Statistics** Cycle: **XXXV°**
Curriculum: **Statistics**

A NOVEL METHODOLOGY TO MAKE TOPIC MODELS PREDICT REAL TOPICS AND TO COMPARE THEM IN BIG DATA CORPUS

Surname: **GERLI**
Name: **SILVIO**
Registration number: **854308**

Supervisor: Prof. **BORROTTI MATTEO**

Academic Year: 2022-2023

Acknowledgment

My first thought goes to my wife, Fiorentina, and my daughter, Michela, who supported and encouraged me, allowing me the time to study and work on this exciting research.

I am also grateful to my parents, who instilled in me the love for learning.

I want to thank Professor Giuletta Minozzi for providing me with many useful pieces of advice, my colleague Dr. Teresa Cigna, who accompanied me through my lucubrations and frequent changes of direction, and my supervisor, Professor Matteo Borrotti, who has always been very kind and focused on finding solutions.

Abstract

Nowadays, the global volume of written text is growing at an ever-increasing pace. Since 2011, the number of posts per minute on Facebook has increased from 650'000 to 3 million. These unstructured data represent a vast source of information that can be exploited with automatic engines. This is primarily achieved through Natural Language Processing (NLP), a field of Artificial Intelligence (AI) dedicated to the analysis and comprehension of human language as it is spoken and written.

One common task in NLP is topics identification, which involves recognizing the topic(s) of a text. Among the automatic solutions (in contrast to engines developed and maintained by linguistic experts), there are two main approaches: Statistical Learning Models (SLM) trained on supervised datasets, capable of identifying real topics and Topic Models (TM), capable of identifying latent topics in unsupervised corpora of documents.

In general, in topic identification research, it is always challenging to find a high-quality training dataset with a known mixture of topics for each text and so that topics come from a taxonomy that covers all possible subjects. A dataset of this kind, preferably extensive and easy updatable, could be of enormous value to train supervised models or to validate the results of various types of models.

Furthermore, TMs have proven to be highly effective in numerous tests since the introduction of the Latent Dirichlet Allocation (LDA) model. While many variants and advancements have been developed in recent years, they all face two issues. Firstly, it is difficult to comprehend what are the "meaning" the identified latent topics. To address this, several methods for labeling these latent topics have been proposed. Secondly, comparing different TMs is tricky because there is no direct relationship between the topics of one model and those of another. Consequently today we have only been able to rely on "self-referential" indicators or manual verification.

In this PhD research many novel methodologies are proposed on these three challenges: two methodologies for creating a large corpus of documents with well-defined mix of topics, four methods for labeling latent topics using this corpus with a supervised approach, six metrics used for performances evaluation of topic models in this context.

These three significant advancements allows to get the main contribution of this research: to establish a rigorous methodological framework compare different TMs on a common and objective ”arena”, providing the opportunity for quantitative performance comparisons, particularly in terms of their ability to accurately identify the actual mix of real topics in documents.

Several experiments have been conducted to validate the effectiveness of this approach. Firstly, extensive comparisons of the ability to identify topics in unknown documents have been carried out between the methodology proposed in this study and random models on one side and supervised statistical learning models on the other side. This was done to ensure that the proposed solution yields reliable outcomes, and the results indeed confirm the correctness of the proposed methodology.

Secondly, multiple comparisons of four TMs, the already cited LDA, Correlated Topic Model (CTM), Hierarchical Dirichlet Process (HDP) and Pachinko Allocation Model (PAM) have been conducted measuring how well they identify the real topics using the proposed methodology. This was assessed using both classical indicators of classification (accuracy, precision, and recall) and all of new metrics proposed in this work.

Last but not least, as a byproduct, a new SLM based on TM has been developed, capable of competing with established ones. This could serve as a viable alternative, given its low computational demands and its production of additional information that can be valuable for refining taxonomies. Consequently in the last part of the research a tuning of the hyperparameters of the best TM emerged from the comparison tests has been performed. At the end, with that optimal settings, a test over a huge dataset of 6 Millions of documents has been conducted.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context

Artificial Intelligence (AI) stands at the forefront of contemporary statistical applications. One of the most important components of any AI model are Natural Language Processing (NLP) methods[1], as language serves as the natural interface for interacting with human beings. Within NLP methods the identification of topics in texts is a very important task. This is crucial as it aids NLP engines in comprehending text but it has also a plethora of direct applications. These include automatic document classification, identification of crucial content (e.g., for privacy, classified information or intellectual properties), automatic categorization of customer tickets, filtering of queries to chatbot, user profiling based on their reading interests, discerning types of complaints on the web, and many others. It is in the area of topic identification that this research aims to make its contribution.

With the advent of machine learning techniques, many tasks that were once manual are now executed by algorithms trained on large datasets. This holds true in NLP, especially in Topic Identification, where linguists can be replaced by automated algorithms. There exist numerous automatic solutions, but they can generally be categorized into two main type of model:

- **Statistical Learning Model (SLM)** trained on supervised datasets, used to identify real topics,

[1] https://online.york.ac.uk/the-role-of-natural-language-processing-in-ai/, https://onlinedegrees.sandiego.edu/natural-language-processing-overview/

- **Topic Model (TM)**, used to identify latent topics in unsupervised corpora of documents.

For the first type of models (SLMs), it's imperative to have a supervised dataset, essentially a corpus of documents, each with a known mixture of topics. Acquiring such a dataset proves to be challenging and, as language evolves rapidly, it needs frequent updates to prevent it from becoming outdated. In this work two solutions to address this crucial need are proposed.

The second type of models (TMs) do not encounter this problem. Among them, there exists a substantial family of highly potent models falling under the umbrella of generative statistical models. They originated with the widely acclaimed Latent Dirichlet Allocation LDA [6]. However, a proliferation of new models has emerged since its introduction, evolving its original approach and promising to enhance the engine's capabilities. While these models have demonstrated remarkable effectiveness in identifying latent topics in documents, they are not designed to provide insights into the actual topics of documents. They are usually employed in an unsupervised environment to cluster corpora of documents. This characteristic not only typically restricts the interpretability of results (a challenge faced by many researches attempting to assign "labels" to latent topics), but it also makes it unclear how to evaluate the "performance" of this family of engines. Usually generic indicators like "coherence" [23] and perplexity [15] are employed. However, they primarily assess the robustness of the models themselves rather than comparing how effectively they truly identify topics (as humans perceive them) in comparison to others. This distinction is crucial because over the years, many evolutions of LDA have emerged, each claiming superiority in one particular aspect. For instance, Correlated Topic Model (CTM) [5] is constructed with consideration for the correlation between topics, suggesting it may excel in identifying the correct mix of topics in a document. Yet, without a means to test this, these improvements remain uncertain.

Thus, what it is required is an "arena" in which TMs can be compared with rigorous measurements. This is precisely this research scope.

1.2 Motivation and Objectives

The core idea revolves around utilizing powerful TMs, initially pioneered by the well known LDA [6], in a novel manner. These models have proven highly effective in uncovering latent

topics within documents but, as stressed before, they do not directly provide information about the actual topics of documents and evaluating the performance of this family of engines remains a complex task. To accomplish this a new methodology is proposed that needs three key prerequisites:

1. Corpora of documents with deterministic real topics, ideally with known measured mixes[2]. Additionally, this dataset should possess the following characteristics:
    - Updatability on demand
    - Big data dimensions to facilitate noise reduction (as this type of data tends to contain a significant amount of noise)
    - Availability in multiple languages
2. Methods to assign each latent topic identified by the model to a real topic. While there are numerous attempts to assign real topics to latent topics (as explored in Chapter 2.1.2), none uses a dataset of documents with known mixes of topics. Consequently, without the use of documents with deterministic topics, none of these methods afford the opportunity to verify and quantify the accuracy of predictions.
3. Metrics for gauging the model's proficiency in predicting the mix of real topics. While classical indicators like precision, accuracy, and recall are suitable when both the prediction and the value to predict are singular, additional metrics are necessary to measure the ability to predict the mix in this context.

In this research, all three fronts of this endeavor are addressed, always remembering that the final outcome is to establish, thanks to these three ingredients, a methodology for assessing the true capability of TMs in identifying real topics, providing a tool to ascertain which model is superior (and how) in this task.

Of course there is even the intention to use the new methodology to compare several TMs. However, before doing this, it has been necessary to pass through a validation step. The last objective is to use the best TMs as a SLMs on a very huge dataset of 6 Millions of documents, after tuning its hyperparameters.

[2]Indeed is more realistic to consider a mix of real topics for a document than to think it is about a single topic.

1.3 Contributions

The primary contributions of this work lies in realizing a methodological framework able to perform an objective comparison of TMs.

To achieve this scope, all three goals presented in chapter 1.2 are addressed and each of them carries other original contributions. More precisely:

1. Development of two methods to create a novel big data corpus of documents with a deterministic mix of topics. In details:

    - First method dynamically creates big data corpora in every language using a customized web crawler to download texts from monothematic websites. So it acquires sets of deterministic monotopic documents (as big as it needed). Moreover it can get sets of deterministic bi-topic documents from subarea of that sites about complementary topics (the two topic would be the topic of the site and the topic of the sub area).
    - Second method can dynamically create big data corpora of deterministic multi-topic documents in every language starting from Wikipedia documents and their categories by an innovative graph algorithm.

2. Development of four methods of automatic topic identification in a supervised environment, namely distribution-based, top words-based, regression-based, and correlation-based methods.

3. Introduction of a new set of performance metrics to evaluate topic labeling approaches in cases of mixed topics (that are typical compositional data)

At the end of all the result will be not just a methodology to compare TMs, but even a new SLMs built using a TMs, that can be an interesting alternative to the existing ones.

1.4 Experiments

In the two papers merged in this thesis several experiments has been performed using four TMs: Latent Dirichlet Allocation (LDA) [6], Correlated Topic Model (CTM) [5], Hierarchical Dirichlet Process (HDP) [7] and Pachinko Allocation Model (PAM) [19].

In the first paper the tests have been focused on two TMs (LDA and CTM). First of all two classical TMs indicators, Coherence [23] and Perplexity [15], have been measured in a big data dataset got by the crawler. Then using two of the four introduced labelling methods (called baptisms) several tests has been performed to evaluate performances using classification indicators (accuracy, precision and recall).

In the second paper the tests cover all the four selected TMs, using even the new metrics, over the new big data corpus of documents built from Wikipedia. First of all many experiments have been conducted to validate the effectiveness of this approach comparing it versus random models and versus SLMs. Then the new comparison methods is used with the four TMs selected. At the end it has been interesting to test the winner on a very huge dataset of 6 Millions of documents after tuning its hyperparameters.

1.5 Original Papers

This thesis comes from the two papers listed below:

1. Gerli, S., Ascari, R., Migliorati, S., Cigna, T., and Borrotti, M., Beyond human labelling: an automatic topic identification framework for big web data, *Submitted to Electronic Journal of Applied Statistical Analysis, currently under review*.

2. Gerli, S., Cigna, T., and Borrotti, M., A novel methodology for developing and testing automatic topic models in big data environment, *in preparation*.

1.6 Thesis Overview

To allow an easy reading of the research, the contents of the two papers listed in chapter 1.5 are merged in a single document.

The thesis is organized as follows. Section 2 gives a brief introduction to the primary theoretical background. Section 3 presents in details the methodologies proposed in both the two papers. Section 4 and 5 describe the experiments and results in the first paper and in the second paper respectively. Section 6 includes final considerations and outlines future works.

Furthermore, appendix A provides a set of definitions used in this thesis. After that there are other three appendixes: one (B) about hyperparameters used and other two (C, D) about details of project 1.

Chapter 2

Background and Theory

2.1 State-of-the-art

Without the possibility to compare different topic models on a common, objective and measurable field actually the only possibility is to use self-referential metrics. In chapter 2.3 two of the most common will be presented: Coherence ([23],[27],[29]) and Perplexity ([15]). They will even be used them in chapter 4.

As explained in chapter 1.2, to provide a methodology to overcome this limit, it is necessary to tackle three challenges. Therefore in the following three paragraphs the state of the art of each of these three fronts is described.

2.1.1 Corpus of document with deterministic topics

Numerous efforts have been undertaken to automatically compile document corpora, associating a set of topics with each document. Most of these approaches, such as those outlined in Lau et al. (2011) [17], Hulpus et al. (2013) [14] and Misra et al. (2021) [24], involve web crawling or utilizing APIs to retrieve documents on specific subjects. For instance, they may involve downloading all articles from a single-themed website, a particular section of a generalist website, or from Usenet newsgroups. However, certain limitations have been identified, including the fact that not all topics are comprehensively covered, and typically, each document is associated with only one topic, rather than a combination of topics.

2.1.2 Automatic labelling of latent topics

Topic Models (TM) produce a collection of latent topics, where each topic is described by a distribution of words. The association of a semantic meaning to these latent topic is not always straightforward. Traditionally, this task is left to human interpretation. However, in the last 15 years, an increasing number of works proposed different approaches for automatic labelling of latent topics. We can divide them in two main families:

- the ones that try to ”create” a label for each topic from its textual content
- the ones that try to find which real semantic topic is more ”similar” to the latent topic

(as shown afterwards, a third way is proposed in this work using a supervised dataset).

Let's see several solutions for both approaches before proceeding.

Labelling from texts

In 2007 Mei et al. [22] proposed an unsupervised probabilistic framework to automatically assign a label to a topic model extracting frequent n-grams and phrases in the corpus and assigning to each latent topic the one most semantically similar to it, most discriminating and with the greatest coverage. In 2010 Lau et al. [18] developed a method for labelling topics based on the top-n terms. The method exploits different ranking mechanisms based on pointwise mutual information and conditional probabilities. In 2011 Lau et al.[17] generated label candidates for a topic using its top words to query Wikipedia and using significant n-grams of titles of the resulting articles to generate candidate labels. Then, they built a Support Vector Regression (SVR) model for ranking the label candidates. In 2021 He et al. [12] introduced a novel two-phase neural embedding framework to generate candidate labels on the basis of similarity between sentences embeddings and the average of the top words embeddings of each topic. Then, a redundancy-aware graph-based ranking process has been implied to rank candidates labels.

Similarity with real topics

Methods relying on external sources for automatic labelling of topics include the work of 2009 by Magatti et al. [20] which, starting from a hierarchy obtained from the Google Directory

service, and expanded through the use of the OpenOffice English Thesaurus, assigned to each latent topic the label of the most similar topic inside the hierarchy. In 2013 Hulpus et al. [14] developed an automatic topic labelling approach by using a structured data source (DBpedia[1]), and deploying graph centrality measures to find the most important *concepts* corresponding to each latent topic and using their labels to generate candidate labels that can characterize the content of the corresponding latent topic. More recently, in 2017, Allahyaru et al. [3] proposed a knowledge-based topic model, namely KB-LDA, which integrates the previous structured data, DBpedia, as a knowledge base for the statistical topic models and makes use of concepts, as before, to find the candidate labels.

2.1.3 Metrics

Metrics, able to measure if and how much a predicted mix of topics is right given the real mix of topics, are needed.

Typical classification indicators (accuracy, recall, precision) could be used but they don't fit well this need, as they are thought to measure correctness of prediction in situation where the target belongs just to one class. So the prediction could be only right or wrong.

As we are working on a Simplex we could use the often used Hilbert distance defined in [26]

$$d_H(p,r) = log \frac{max_{i\in(i,\dots,T)} \frac{p_i}{r_i}}{min_{i\in(i,\dots,T)} \frac{r_i}{p_i}} \tag{2.1}$$

where T is the length of $\mathbf{r}$ and $\mathbf{p}$ i.e. the number of real topics, but it is not suitable because it enlarges distances till infinity near boundaries, that doesn't fit in our case.

Another possibility could be to use simple Euclidean metric, however it may not be most suitable, because for instance if in a simple case of three topics (Animals, Sports and Food), if in a document the real topics are: Animals and Sports and the predicted is Food, their distance depends on the weights of the two real topics (that sum to 1), but indeed, it must be always the maximum values (i.e. 1), since the method didn't guess any real topic whatever balance between Animals and Sport.

[1] http://dbpedia.org

2.2 Topic models

2.2.1 LDA, HDP, CTM and PAM

In this section, the Latent Dirichlet Allocation (LDA) ([6]), the Correlated Topic Model (CTM) ([5]), the Hierarchical Dirichlet Process (HDP) ([7]) and Pachinko[2] Allocation Model (PAM) ([19]) are introduced.

LDA is chosen because it is the first acclaimed Topic Model, ancestor of all the other, and still on the edge. As LDA needs to be fed with the number of topics k, that is always tricky to define, the obvious alternative is HDP, that estimates the number of topics too. Moreover, as for the aim of this research it is important to show the ability to recognize mix of topic, the last two ones are chosen because they claim to take into consideration the correlation between topics even in a different way. Moreover all these model are implemented in the same Python libraries (Tomotopy) making it easy to uniform results.

Let's start from LDA ([6]) from which all other TM are derived. Let D be the number of documents belonging to a corpus, the d-th document having N_d words ($d = 1, \dots, D$). The set $\mathscr{V}$ of unique words appearing in the corpus has cardinality V, and it is referred to as "vocabulary".

TM techniques are based on the "bag-of-words" assumption, which implies that word order in a document is irrelevant. The only relevant information in a document is the number of times (i.e., the frequency) each word appears in the document itself. The "bag-of-words" coincides with an exchangeability assumption for the words within a document.

A further assumption of these approaches is the representation of a document as a probability distribution over a set of K (latent) topics, where a topic is represented by a distribution over words (i.e., with support the vocabulary $\mathscr{V}$). Thus, a document can be depicted as a point $\boldsymbol{\theta}$ in the K-part *topic simplex*

$$\mathscr{S}^K = \left\{ \boldsymbol{\theta} = (\theta_1, \dots, \theta_K)^\intercal, \theta_k > 0, \sum_{k=1}^{K} \theta_k = 1 \right\} \tag{2.2}$$

[2]The name comes from a mechanical arcade game originating in Japan

whereas a topic corresponds to a vector ϕ belonging to the V-part *word simplex*

$$\mathscr{S}^V = \left\{ \phi = (\phi_1, \ldots, \phi_V)^\intercal, \phi_v > 0, \sum_{v=1}^{V} \phi_v = 1 \right\} \tag{2.3}$$

Once the K topic-specific word distributions $\phi_1, \ldots, \phi_K$ have been generated from a proper distribution, the generative process for the d-th document can be summarized by the following steps:

1. sample θ_d from a distribution $\mathscr{F}$ defined on $\mathscr{S}^K$;
2. for the n-th word of the document ($n = 1, \ldots, N_d$):
   (a) sample a topic $z_{d,n}$ from $Z_{d,n} \sim Categorical(\theta_d)$;
   (b) sample a word $w_{d,n}$ from $W_{d,n}|Z_{d,n} = z_{d,n} \sim Categorical(\phi_{z_{d,n}})$.

LDA assumes a Dirichlet distribution for the word distribution for k-th topic

$$\phi_k \sim Dir(\beta), \qquad k = 1, \ldots, K, \tag{2.4}$$

where

$$\beta = (\beta_1, \ldots, \beta_V)^\intercal \tag{2.5}$$

and

$$\beta_v > 0 \qquad for v = 1, \ldots, V \tag{2.6}$$

and again a Dirichlet distribution for the topic distribution on the d-th document

$$\theta_d \sim Dir(\alpha), \qquad d = 1, \ldots, D, \tag{2.7}$$

where

$$\alpha = (\alpha_1, \ldots, \alpha_K)^\intercal \tag{2.8}$$

and

$$\alpha_k > 0 \qquad \forall k = 1, \ldots, K \tag{2.9}$$

Figure 2.1 summarize the LDA by means of a directed acyclic graph (DAG).

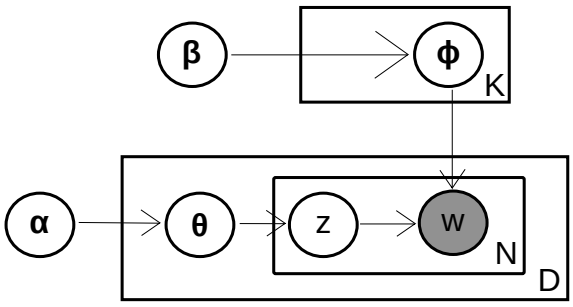

Figure 2.1: DAG representing the LDA model. Filled nodes represent observed variables.

Differently, CTM ([5]) assumes that $\boldsymbol{\theta}_d$ follows a logistic-normal distribution [2, 5], that is the log-ratio transformation

$$\boldsymbol{\eta}_d = \left(\log\left(\theta_{d1}/\theta_{dK}\right), \ldots, \log\left(\theta_{d(K-1)}/\theta_{dK}\right)\right)^{\mathsf{T}} \tag{2.10}$$

is assumed to follow a $(K-1)$-dimensional normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. Therefore, CTM enriches the dependence structure of LDA by including any kind of correlation (i.e., not only negative but also positive) between log-ratio transformed elements of $\boldsymbol{\theta}$. Though, this comes at the cost of complicating the interpretation of the dependence structure on the original space. There is not a clear relationship between the correlations between log-ratio transformed elements and the original ones. Moreover, and differently from the Dirichlet distribution, the logistic-normal distribution does not possess conjugacy with respect to the categorical distribution, which has a negative impact on the computational aspects of model inference.

Figure 2.2 summarize CTM model by means of a DAG.

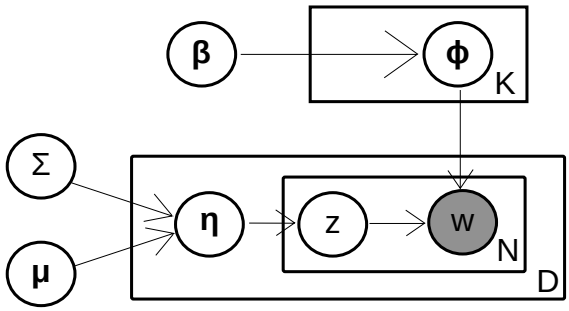

Figure 2.2: DAG representing the CTM model. Filled nodes represent observed variables.

HPD ([7]) is a nonparametric extensions of LDA, which allows the number of topics to be learnt from data. It uses again a Dirichlet process to capture the uncertainty in the number K of topics. So a common base distribution is selected which represents the countably-infinite set of possible topics for the corpus, and then the finite distribution of topics for each document is sampled from this base distribution.

HDP model assumes three matrices to establish relationships between words, topics and documents, constructed as below:

- α_k represents top level Dirichlet variables sampled for a given topic k
- θ_d represents the topic distribution for a given document d
- ϕ_k represents the word distribution for a given topic k

Thus the generative process for HDP is, given parameters β, γ, η:

$$\alpha_k \sim Dir(\gamma/K)$$
$$\theta_d \sim Dir(\eta\,\alpha_k)$$
$$\phi_k \sim Dir(\beta)$$
$$z_{id} \sim \theta_d \ , \ w_{jd} \sim \phi_{z_{id}}$$

In this model, K denotes the number of topics, and for the purposes of HDP, is taken to tend to infinity. Like in the LDA model, each variable is modelled by a symmetric Dirichlet distribution, while each topic z_{id} of a document d ($i = 1, ...k_d$) is sampled from θ_d and each word w_{ij} is sampled from the corresponding topic $\theta_{z_{id}}$. This process is 'hierarchical' in the sense that it adds another layer to the model, the Dirichlet Process, which determines the number of topics.

The last topic model, PAM [19], tries to intercept captures arbitrary, nested, and possibly sparse correlations between topics using a directed acyclic graph (DAG). The leaves of the DAG represent individual words in the vocabulary, while each interior node represents a correlation among its children, which may be words or other interior nodes (topics). In other words the concept of topics are extended to be distributions both over words and over other topics; PAM therefore captures not only correlations among words (as in LDA), but also correlations among topics.

In the tests the four-level version is used, well represented by the figure 2.3, consisting, from the upper part, of a root, a set of super-topics, a set of sub-topics and a words vocabulary. Both the root and the super-topics are associated with Dirichlet distributions, from which we sample multinomials over their children for each document.

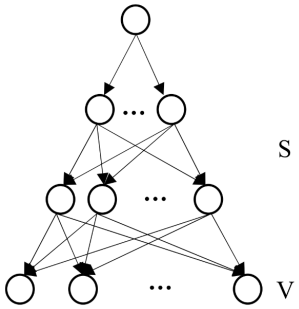

Figure 2.3: Four-level hierarchy PAM

So for this model we have to define two number of topics: the number s_1 of the first level topics and the number s_2 of second level topics, higher than the first (called respectively "super-topics" and "subtopics", not to be confused with what it is called in this way as in A).

2.2.2 Inference with topic models

TM can be fitted by either a variational inference approach, as originally proposed by [6], or by a fully collapsed Gibbs sampling [10]. The latter approach can be improved by partitioning the data across separate processors and performing inference in parallel, as suggested by [25].

Furthermore, inference for unseen documents is based on techniques from discriminative text classification as proposed by [31]. The main idea of their approach is to move the collapsed Gibbs for additional iterations on an "updated" corpus including also the unseen documents. At the end of these additional iterations, an estimate for the vector of topic proportions of the new documents is obtained, namely $\hat{\boldsymbol{\theta}}_{new}$. Additional details on the implementation of these techniques can be found in the cited works.

2.3 Typical latent topic performance indicators

To evaluate the performance of competing models, two well-established classes of measures, namely coherence and perplexity have been used.

2.3.1 Coherence

Coherence measures (e.g., see [23], [27], and [29]) give an evaluation of how much the model is "certain" in what it says. They have been introduced to assess the intrinsic coherence of the k-th latent topic ($k = 1, \ldots, K$) identified by a latent topic model. These measures are strictly connected with the largest estimated elements of ϕ_k. More precisely, for each pair of words in $\mathscr{I}_k^M$, a "confirmation measure" is computed, which is a function depending on the probability $P\left(w_m^{(k)}\right)$ that a document contains the word $w_m^{(k)} \in \mathscr{I}_k^M$ at least once, and the probability $P\left(w_m^{(k)}, w_l^{(k)}\right)$ that a document contains at least once word $w_m^{(k)} \in \mathscr{I}_k^M$ and at least once word $w_l^{(k)} \in \mathscr{I}_k^M$ ($m, l = 1, \ldots, M$; $m \neq l$). Then, a coherence measure for topic k is simply obtained by computing the mean of the confirmation measures over all the pairs of words in $\mathscr{I}_k^M$. Higher values of coherence measures are associated with the most interpretable topics.

In particular, three types of coherence measures based on different confirmation measures have been used. The first considers the pointwise mutual information (PMI) as a confirmation measure:

$$C_{\text{UCI}}^{(k)} = \frac{2}{M \cdot (M-1)} \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \text{PMI}\left(w_m^{(k)}, w_l^{(k)}\right), \tag{2.11}$$

where

$$\text{PMI}\left(w_m^{(k)}, w_l^{(k)}\right) = \log\left(\frac{P\left(w_m^{(k)}, w_l^{(k)}\right) + \varepsilon}{P\left(w_m^{(k)}\right) \cdot P\left(w_l^{(k)}\right)}\right)$$

and ε is a small positive term added to ensure stability of the logarithm function.

A slightly modification of PMI is its normalized version (NPMI), which allows to define a second coherence measure:

$$C_{\text{NPMI}}^{(k)} = \frac{2}{M \cdot (M-1)} \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \text{NPMI}\left(w_m^{(k)}, w_l^{(k)}\right), \tag{2.12}$$

where

$$\text{NPMI}\left(w_m^{(k)}, w_l^{(k)}\right) = \frac{\text{PMI}\left(w_m^{(k)}, w_l^{(k)}\right)}{-\log\left(P\left(w_m^{(k)}, w_l^{(k)}\right) + \varepsilon\right)}.$$

Lastly, the topic coherence measure introduced by [23] is defined as:

$$C_{\text{UMass}}^{(k)} = \frac{2}{M \cdot (M-1)} \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{P\left(w_m^{(k)}, w_l^{(k)}\right) + \varepsilon}{P\left(w_l^{(k)}\right)}. \tag{2.13}$$

Coherence measures can also be aggregated by averaging the same measures over topics (e.g., $C_{\text{UMass}} = \sum_{k=1}^{K} C_{\text{UMass}}^{(k)}/K$).

2.3.2 Perplexity

The perplexity index is a further measure of model performance. It gives a measure of how much the model is "uncertain" in front of new documents. In particular, given a new corpus composed of D' unseen documents $\mathscr{C}^T$ (playing the role of a test set), perplexity is computed as

$$perplexity\left(\mathscr{C}^T\right) = \exp\left\{-\frac{\sum_{d=1}^{D'} \log p(\mathbf{w}_d)}{\sum_{d=1}^{D'} N_d}\right\}, \tag{2.14}$$

where $\mathbf{w}_d = (w_{d,1}, \ldots, w_{d,N_d})^\intercal$ is the vector of words composing the d-th document, and $p(\mathbf{w}_d)$ denotes the probability assigned by the model to words in document d (i.e., its likelihood). [6] showed that this measure can be represented as the inverse of the geometric mean of per-word likelihood, thus the larger the likelihood, the smaller the value of the perplexity. This entails that, in comparing fitted models, the lower the perplexity the better the model.

2.4 Compositional data

In all this research there are mix of topics, i.e. vectors of n values that are not independent as they are all positive and bound to sum to 1. It is true both for the vector of latent topic and for the vectors of real topics (actual of predicted). Vectors of positive real numbers in which the sum S is fixed are called compositional data (see [1]. Called N the dimension of the vector, there

are (N-1) degrees of freedom, as know (N-1) of them the remaining one is uniquely determined by the difference between S and the other (N-1). In fact they take values on a portion of (N-1)-dimensional hyperplane that owns all the point P_k $(k = 1, ...N)$:

$$P_k = (v_1, v_2, ..., v_N) \qquad where \begin{cases} v_i = S \; if \; i = k \\ v_i = 0 \; if \; i \neq k \end{cases} \quad k = 1, ..., N \tag{2.15}$$

This portion of the (N-1)-dimensional hyperplane is called Simplex. In a 3-D space the Simplex is the triangle with vertexes (S,0,0), (0,S,0) and (0,0,S), as it is possible to see in figure 2.4 in the case of S=1.

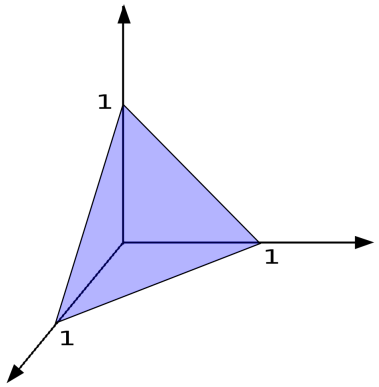

Figure 2.4: The 3-D simplex with S=1

2.4.1 Regression with compositional data

Another element to be taken into consideration is that they are bounded in the range [0,S] that in cases of this research, with quotas and probabilities, is [0,1] (S=1). So dealing with this data in a regression this bounds have to been managed.

One possibility is to use Dirichlet regression (see [13]).

Another possibility is to use linear regression modifying the response variable with *Logit* function:

$$y_i^{new} = logit(y_i) = \frac{1}{1+e^{-y_i}}, \qquad i = 1,\dots,H \tag{2.16}$$

In this way the original range [0,1] is dilated to the range to $[-\infty,+\infty]$. To manage the problem of infinity values in case of original 0 or 1 values they can be substituted by 0.001 and 9.999 respectively and then conduct regression analysis on this transformed dataset. Consequently, if necessary, the prediction are re-transformed using the *softmax* function.

$$\hat{y}_i^{new} = \frac{e^{\hat{y}_i}}{\sum_{j=1}^{H} e^{\hat{y}_j}}, \qquad i = 1,\dots,H \tag{2.17}$$

The second approach has been preferred for many reasons. First of all Dirichlet regression is not well implemented yet anywhere. There is a library in R [3] (as analysed in [21]) that is not so efficient and able to manage big data datasets (furthermore coding is developed in Python). Moreover linear regression is computationally far less expensive. And, last but not least, a test to compare two approaches on a small dataset on R gave similar performances (surprisingly the linear regression with logit transformation is even slightly better).

[3] https://cran.r-project.org/web/packages/DirichletReg/

Chapter 3

Proposed Methodologies

This chapter describes the solutions proposed in the two papers for each of the three challenges presented in chapter 1.2. Furthermore, in the last section, details on how data has been prepared for the experiments are given.

3.1 Corpus of document with deterministic topics

Currently, there is an urgent need to establish methods for constructing labeled data corpora, i.e. datasets of documents in which the real topic of each document is known. Such a dataset would be of enormous value in topic analysis, Not only for our purpose to label latent topics, for instance it would be very valuable to train and test all statistical learning methods. Another important fact is that in real world a text is not about single topic, instead a text is quite always about several subjects in a specific proportion. Consequently our aim is to create a big data corpus with documents where, in each document, the topics and their proportions are known. So, even if a corpus with ”single topic documents” (i.e. where just the main topic is known) would be acceptable for our aims, it would be far better to have a corpus with ”mix of topics documents” (i.e. where different topics and their proportions are known).

3.1.1 Crawling thematic websites

To achieve this result, the first introduced method [1] uses a web crawler [2] executed on mono-thematic portals (i.e. websites that speak about a particular topic). The use of mono-thematic portal guarantee us that articles of that site are most likely about the topic of website. Thus, it can create a dataset where two types of recorded information are provided for each raw:

- the article text
- the real topic of the text (=the topic of the mono-thematic website it is downloaded from)

To obtain articles with a mix of known real topics, subsection of mono-thematic portals are considered. Indeed, portals are organized in subsections, typically sub-homepages, on which articles are related both to the main topic of the portal and to the topic of the subsection (e.g., a subsection about “Health” in a mono-thematic portal about “Animals”). Thus, all texts downloaded from a subsection are labelled with a mix of this two real topics (considered agnostically 50% each).

3.1.2 Download from Wikipedia using its categories

Big data corpus of documents with defined topics: download from Wikipedia

A very interesting alternative (followed in the second paper) to create huge corpus of documents is to use one of the biggest and best-structured free source of documents in the world: Wikipedia.

The dataset can be created from the Wikimedia dumps `https://dumps.wikimedia.org/enwiki/latest/` which has all articles of the English version of Wikipedia ($\sim$ 6,5 Million of articles). Each article is related to a vector or topics (called ”categories”) taken from a set of $\sim$ 2 Million of topics. This is an unmanageable number of topics, but they are not independent as they are interconnected through a complex network of approximately 8 million relationships.

[1] In the first paper.

[2] A web crawler is a software able to browse autonomously a website following all internal links and to download text from all articles.

Taxonomy construction on Wikipedia categories

Starting from Wikipedia's topics/categories, we aim to create a methodology to assemble on demand a custom, unique and very rich corpus of documents starting from the selection of a language and the definition of a topics taxonomy. As a matter of fact with the proposed methodology one can select a language (e.g. French) and choose a set of topics (ideally that covers all possible subjects of a document) and obtain a big data corpus of texts, each with the measured mix of real topics among the ones in the chosen taxonomy.

For this work we decided to select English language and to use a taxonomy based on first level topics list the Wikipedia choice (https://en.wikipedia.org/wiki/Category:Main_topic_classifications). Wikipedia identifies 39 primary or first-level topics, which are listed in Table 3.1.

Our ultimate objective is to automatically assign each of the 2 million categories to one of these 39 main topics, so, using our convention, $H = 39$.

Table 3.1: List of the 39 top-level topics according to Wikipedia

Academic disciplines	Entities	Internet	Philosophy
Business	Ethics	Knowledge	Politics
Communication	Food and drink	Language	Religion
Concepts	Geography	Law	Science
Culture	Government	Life	Society
Economy	Health	Mass media	Sports
Education	History	Mathematics	Technology
Energy	Human behavior	Military	Time
Engineering	Humanities	Nature	Universe
Entertainment	Information	People	

For their origin and maintenance the 2 million of topics are not organized in a taxonomy. Each editor can introduce a new topic placing it as subtopic of one or more existing ones. So each topic is associated with a set of child topics and can have multiple parent topics. These topic relationships can be structured in a directed graph with 2 million nodes representing the topics and 8 million directed edges indicating the 'subtopic of' relationships. And this is what we did, and then we imported this huge structure in Neo4J[3] cluster for further analysis.

We define the 'distance between topics' as the minimum[4] number of hops between two nodes. Selected two topics a and b, called p one of the the possible P_{a-b} path to get from a to b, and called $n(p)$ the number of nodes in the p path, we define the distance between a and b as:

[3] https://neo4j.com/

[4] Obviously there could be a lot of paths connecting two nodes in such a graph

$$d(a,b) = min_{p\in(0,1,...,P_{a-b})}(n(p)) \tag{3.1}$$

For each node, which represents a Wikipedia topic, we identify the 'first-level' topic it belongs to as the nearest 'main node'. Called f_h the h-th 'first-level' topic with $h \in (0,1,..,H)$, selected a topic a and defining r_a

$$r_a = \underset{h\in(0,1,...,H)}{\arg\min}\ d(a,f_h) \tag{3.2}$$

we say that a topic a belongs to the 'first-level' topic f_{r_a} (see Figure 3.1 for an example). In rare cases where a topic has equal distances to m 'first-level' topic nodes (*i.e.* r_a is not a single value but an arrey of values), all those 'first-level' topics are considered in equal proportions for that topic to sum to 1. In other words in this rare case each 'first-level' topic for the original topic a has a coefficient s not equal to 1 but to 1/m.

Subsequently, based on these associations, we replace each original topic of the T topics in each article with their associated 'first-level' topic(s). After this process each article is associated with a list of 'first-level' topics with their coefficient s. To calculate the percentages of the 'first-level' topic h related for each article, we sum all coefficient s_h of that 'first-level' topic and divide by the number of original topics. Considering an article let's define n_h with $h \in (0,1,...,H)$ the number of time that the h-th 'first-level' topic appears in the document and call s_n with $n \in (0,1,...,n_h)$ the coefficient of each 'first-level' topic. The quota q_h of that d-th first-level topic in that document is:

$$q_h = \frac{\sum_{n=1}^{n_h} s_n}{\sum_{i=1}^{H} n_i} = \frac{\sum_{n=1}^{n_h} s_n}{T}$$

If an original topic is related to multiple 'first-level' topics, the summed value of each 'first-level' topic is not an integer anymore but 1 divided by the total number of 'first-level' topics associated with that topic (see Table 3.2 for an example).

Therefore, the final dataset is composed of the title, text of the article, and an array of 'first level' topics with their percentages.

Note that the proposed method is independent of the selection of the first-level topics. A different list of first-level topics (among the 2 Million categories in Wikipedia) can be selected

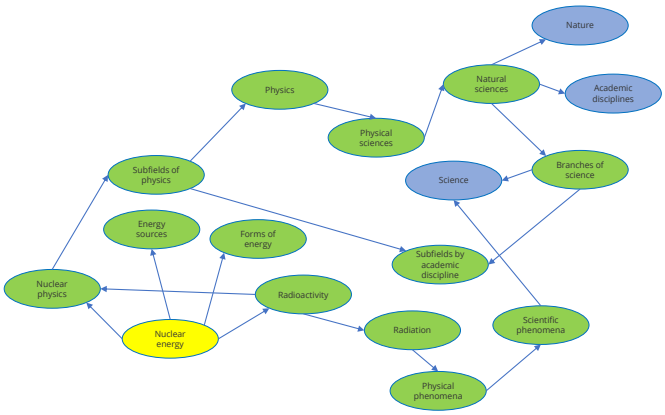

Figure 3.1: Nuclear energy is categorized as a subtopic of Science because the minimum distance from this first-level topic is 5 hops, which is shorter than the minimum distance from the other two first-level topics, Academic discipline and Nature, where the minimum distance is greater, specifically 6 hops.

and a new mix of topics per article can be defined by exploiting the direct graph.

Values of the dataset

The main features of a dataset created with this methodology are:

1. **Language independent**: In this work it is created in English. But using the same method, a similar dataset could be created in every language present articles of other languages Wikipedia (18 languages have more than 1M of documents, and 30 more then 500K [5]).

2. **Big data**: It is a very large dataset made of about 6 million articles (in English).

3. **Topics coverage**: Topics cover all human knowledge and they are organizable in a Taxonomy of different levels. With the same procedure the second level list of topics could be used (and so on).

4. **Measured mix**: Even the measure of the weight of each topic in an article comes automatically, computed from the Wikipedia categorization.

[5] https://meta.wikimedia.org/wiki/List_of_Wikipedias

Categories	Nearest main topic
Comparative Politics	Politics
Revolution	Human Behaviour, Society
Revolutions	Government, Society
Social concepts	Society
Social conflicts	Society

Main topic	Percentage
Society	60%
Politics	20%
Human Behaviour	10%
Government	10%

Table 3.2: Example of main topics mix of the Wikipedia article about Revolution (https://en.wikipedia.org/wiki/Revolution). In the left table the first column shows the categories assigned by Wikipedia to this article. In the second column, the correspondent nearest main topics are shown. It is notable that Revolution and Revolutions have 2 main topics at the same distance. In the second table, the final mix for this article is shown.

5. **Updatable**: Is is possible to keep it up-to-date following the changes in the languages. It's enough to download the up-to-date dump from Wikipedia and to perform the automatic procedure again.

6. **Taxonomy independent**. It is not limited to the first level topics of Wikipedia. A different list of "first level" topics can be selected (among the 2 Million of categories in Wikipedia) and, using the graph, the mix of these topics for each article can be identified . I have to say that I don't think that the Wikipedia choice of the first level topics is the best one that could be produced. But this subject is outside the scope of this work and it has been decided to trust the choice of Wikipedia.

3.2 Automatic labelling of latent topics

As seen in Appendix A "baptism" is the process that leads to the assignment of a real topic to a latent topic or to the definition of it as a pseudo-topic.

In the two papers four methods are introduced. The first two needs monotopic documents to be trained, the following two, instead, can work on documents with a know mix of topics.

3.2.1 Baptism methods using monotopic documents

In the first paper, two methods for baptizing topics are proposed, that need monotopic documents dataset to be trained: "distribution-based" method and a "top words-based" method.

Distribution-based method

Let $\mathbf{y}$ be the real topics vector, with elements $y_d \in \mathscr{R} = \{r_1, \ldots, r_H\}$ representing the real topic of document d, and let $\hat{\boldsymbol{\theta}}_d$ be the score distributions obtained from a latent topic engine (that is, the estimate of the topic-composition of document d, $d = 1, \ldots, D$). To assess whether latent topic k should be associated to real topic r, $r \in \mathscr{R}$, let's consider the scores vectors $\tilde{\boldsymbol{\theta}}_k = \left(\hat{\theta}_{1k}, \hat{\theta}_{2k}, \ldots, \hat{\theta}_{Dk}\right)^\intercal$, $k = 1, \ldots, K$, where $\hat{\theta}_{dk}$ is the k-th element of $\hat{\boldsymbol{\theta}}_d$ (i.e., the estimated proportion of topic k in document d).

Then, the distribution-based method sums the elements of $\tilde{\boldsymbol{\theta}}_k$ corresponding to those documents whose real topic is r, that is

$$\tilde{p}_k^r = \sum_{d=1}^{D} \hat{\theta}_{dk} \mathbb{I}\left(y_d = r\right), \qquad r \in \mathscr{R} \tag{3.3}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

Finally, a probability distribution is computed by normalization:

$$p_k^r = \frac{\tilde{p}_k^r}{\sum_{l \in \mathscr{R}} \tilde{p}_k^l}, \qquad r \in \mathscr{R}. \tag{3.4}$$

Let's define the K-dimensional vector $\mathbf{B}$ as the vector with elements b_k representing the real topic that is going to be assigned to latent topic k. In order to baptize the k-th latent topic, only the largest probability p_k^r and the second largest probability $p_k^{r'}$ are considered, and the decision is based on their difference. If this difference is equal to or greater than a given threshold t_d (that is an hyperparameter), then latent topic k is baptized as "real topic r" (i.e., the one corresponding to the highest probability) and $b_k = r$, otherwise it is considered as a pseudo-topic and thus discarded.

Algorithm 1 summarizes the main steps of the distribution-based method, while Table 3.3 illustrates it by means of a simple example. Here, a latent topic model has been fitted on a corpus composed of $D = 9$ documents, and $H = 4$ real topics, namely "Health", "Fashion", "Celebrities", and "Animals". The example considers the baptism of latent topic k when $\tilde{\boldsymbol{\theta}}_k = (0.1, 0.8, 0.7, 0.15, 0.1, 0.2, 0.01, 0.9, 0.04)^\intercal$. The latent topic k will be baptized as real topic "Fashion", since it is the real topic associated to the largest probability (i.e., 0.8), differing from

the second largest probability (i.e., 0.13) by more than a threshold t_d equal to 0.2.

Algorithm 1 Pseudo-code: distribution-based method (learning phase)

procedure DISTRIBUTION($\mathscr{C}, \mathscr{R}, \tilde{\theta}_k$)
  $\forall r \in \mathscr{R}$ compute $\tilde{p}_k^r = \sum_{d=1}^{D} \hat{\theta}_{dk} \mathbb{I}(y_d = r)$ ▷ $\mathbb{I}(\cdot)$ denotes the indicator function
  Normalization step:
  $\forall r \in \mathscr{R}$ compute $p_k^r = \dfrac{\tilde{p}_k^r}{\sum_{l \in \mathscr{R}} \tilde{p}_k^l}$
  $\hat{r}^{top1} \leftarrow$ real topic with the highest value p_k^r
  $\hat{r}^{top2} \leftarrow$ real topic with the second highest value p_k^r
  if $P(\hat{r}^{top1}) - P(\hat{r}^{top2}) > t_d$ **then** ▷ t_d is a fixed threshold
    $b_k \leftarrow$ real topic $\hat{r}^{top1}$
  else
    $b_k \leftarrow$ "pseudo-topic" ▷ The latent topic k is discarded
  end if
end procedure

	Real topic			
Documents	**Health**	**Fashion**	**Celebrities**	**Animals**
Doc1 (Health)	0.1			
Doc2 (Fashion)		0.8		
Doc3 (Fashion)		0.7		
Doc4 (Animals)				0.15
Doc5 (Health)	0.1			
Doc6 (Health)	0.2			
Doc7 (Celebrities)			0.01	
Doc8 (Fashion)		0.9		
Doc9 (Celebrities)			0.04	
$\tilde{p}_k^r$ (Equation (3.3))	0.4	2.4	0.05	0.15
p_k^r (Equation (3.4))	**0.13**	**0.8**	0.02	0.05

Table 3.3: Toy example - Baptism of latent topic k by means of the distribution-based method with threshold $t_d = 0.2$.

Top words-based method

The second method compares the most probable words recovered by a latent topic engine and the most frequent words appearing in a real topic.

Given a corpus $\mathscr{C}$ and a set of real topics $\mathscr{R}$ as in Algorithm 1, for each real topic $r \in \mathscr{R}$ let's define the set $\mathscr{T}_5^r$ of top-5 unique words, namely the set of the five most frequent words in documents for which $y_d = r$. The term "unique" means that words in $\mathscr{T}_5^r$ and $\mathscr{T}_5^{r'}$ are selected such that $\mathscr{T}_5^r \cap \mathscr{T}_5^{r'} = \varnothing$ for any $r \neq r'$. Also select the set $\mathscr{I}_k^{10}$ containing the ten most probable words of latent topic k is selected (i.e., the ten words associated to the largest values in $\hat{\phi}_k$, which is an estimate of ϕ_k, the word distribution for topic k). Then, the frequencies of words in $\mathscr{I}_k^{10}$ which appear in each of the sets $\mathscr{T}_5^r, r \in \mathscr{R}$ are computed, and the difference between the largest and the second largest frequencies is considered. If this difference is equal to or greater than a given threshold t_{tw} (that is an hyperparameter), then latent topic k is baptized with the real topic r corresponding to the highest frequency, otherwise it is considered as a pseudo-topic and thus discarded. The method is summarized in Algorithm 2, while Table 3.4 shows a simple example with $t_{tw} = 2$. In the example, the latent topic k will be baptized as real topic "Health", since it has the highest frequency (i.e., 4) and the difference between this value and the second largest frequency (i.e., 2) is greater than or equal to $t_{tw} = 2$. Then, the result will be stored in position k of the vector $\mathbf{B}$, by assigning b_k = "Health".

Algorithm 2 Pseudo-code: top words-based method (learning phase)

procedure TOPWORDS($\mathscr{C}, \mathscr{R}, \mathscr{I}_k^{10}$)
▷ $\mathscr{I}_k^{10}$, ten most probable words of latent topic k
  $\forall r \in \mathscr{R}$ find $\mathscr{T}_5^r$
  Initialize w as a vector of 0 with size equal to the number of real topics
  $r \in \mathscr{R}, w_r \leftarrow$ frequency of words in $\mathscr{I}_k^{10}$ that appear in each of the sets $\mathscr{T}_5^r$
  $\hat{r}^{top1} \leftarrow$ real topic with the highest value w_r
  $\hat{r}^{top2} \leftarrow$ real topic with the second highest value w_r
  if $P(\hat{r}^{top1}) - P(\hat{r}^{top2}) \geq t_{tw}$ **then** ▷ t_{tw} is a fixed threshold
    $b_k \leftarrow$ real topic $\hat{r}^{top1}$
  else
    $b_k \leftarrow$ pseudo-topic ▷ The latent topic k is discarded
  end if
end procedure

Real topics	**Frequency**	**Top 5 words**
Health	4	**'patients', 'treatment', 'visit'**, 'medicine', **'report'**
Animals	0	'dog', 'cow', 'cat', 'animals', 'veterinary'
Celebrities	2	**'photos'**, 'gossip', 'event', **'daughter'**, 'vip'
Fashion	1	'fashion', 'look', 'showroom', 'collection', **'style'**
Latent topic k		'patients', 'treatment', 'visit', 'daughter', 'car', 'report', 'photo', 'street', 'style', 'word'

ho

Table 3.4: Toy example - Baptism of latent topic k by means of the top words-based method with threshold $t_{tw} = 2$.

3.2.2 Baptism methods using documents with known mix of topics

Using a corpus of documents with a known mix of real topics, in the second paper the former baptism methods for generative topic models have been evolved in two new ones able to manage this more complex but richer dataset . The generative topic models have been trained on a subset of documents (the train set): a model that, for each document, gives the mix of latent topic with different proportions (that sum to 1). So for each document of the train set the mix of real topic on one side and the mix of latent topic on the other side have been obtained.

Two new methods have been introduced for this situation ("Regression Based" and "Correlation based").

Regression-Based

Given a training set of D documents labelled with H topics, let $\mathbf{X}$ be a matrix of size $D \times H$, where each row represents the vector $\mathbf{x}_d$ of quotas of the H real topics of the document d of the training set: the element x_{dh} represents the proportion of real topic h in document d. These proportions in $\mathbf{x}_d$ sum to 1.

Now, let Θ be a matrix of size $D \times K$ where each row represents the vector $\boldsymbol{\theta}_d$ of the quota of the K latent predicted by the model for the document d.

To determine whether latent topic k should be associated with real topic h, let's keep the k-th coloumn of Θ, that is a D-dimensional vectors $\boldsymbol{\theta}^k = \left(\theta_1^k, \theta_2^k, \ldots, \theta_D^k\right)^\intercal$, that is the vector of D-values predicted by the model for the k-th latent topic (one for each of the D documents).

At this stage, a linear regression model has been used, treating $\boldsymbol{\theta}^k$ as the dependent variable

and using $\mathbf{X}$ as the set of independent variables with corresponding values. Since this type of regressor does not have a specific range over the response variables whereas our data does (i.e. [0,1]) it has been decided expand [6] the range of each response variable from [0,1] to $(-\infty,+\infty)$ as explained in chapter 2.4.1 and so to work on $\mathbb{R}^k$. Let's call the transformed variable $\bar{\boldsymbol{\theta}}^k$. In matrix form, the following model is defined:

$$\bar{\boldsymbol{\theta}}^k = \mathbf{X}\boldsymbol{\beta}^k + \boldsymbol{\varepsilon}. \tag{3.5}$$

Where $\boldsymbol{\beta}^k$ is the H-dimensional vector $(\beta_1^k, \beta_2^k, ..., \beta_H^k)$ to be estimated by the linear regression for each k, $k = 1, \ldots, K$. Once obtained the estimated values $\hat{\boldsymbol{\beta}}^k$ each latent topic LT_k can be written as linear combination of the H real topic RT_h:

$$\begin{aligned} LT_1 &= \hat{\beta}_1^1 RT_1 + \hat{\beta}_2^1 RT_2 + ... + \hat{\beta}_H^1 RT_H \\ LT_2 &= \hat{\beta}_1^2 RT_1 + \hat{\beta}_2^2 RT_2 + ... + \hat{\beta}_H^2 RT_H \\ &... \\ LT_K &= \hat{\beta}_1^K RT_1 + \hat{\beta}_2^K RT_2 + ... + \hat{\beta}_H^K RT_H \end{aligned} \tag{3.6}$$

	Real topics			
Latent topics	**Health**	**Fashion**	**Celebrities**	**Animals**
1	2.9	0.5	0.1	1.2
2	0.1	0.8	0.2	0.9

Table 3.5: Toy example for regression based method - Baptism of latent topics 1 e 2 with threshold $t_r = 0.2$. Each cell value is the coefficient of the real topic of its column in the linear combination for the latent topic on its raw. The latent topic 1 is baptized as "health" because it has the greater coefficient (2.9) and the difference with the second (Animals=1.2) is grater then the threshold $(2.9 - 1.2 = 1.7 > 0.2)$. Instead the latent topic 2 is baptized as "pseudo-topic" because the difference between the greater coefficient (0.9 for Animals) and the second (0.8 for Fashion) is lower than the threshold $(0.9 - 0.8 = 0.1 < 0.2)$.

To perform the baptism of the k-th latent topic, the vector $\boldsymbol{\beta}^k$ from the k-th regression model is used. Specifically, the two highest values are taken into account: $\beta_{h_1}^k$ (the highest) and $\beta_{h_2}^k$ (the second highest). Our decision is based on the difference between these two values. If the

[6] As the focus is on the relations between real and latent topics, it is not necessary to re-transform variables to the original range subsequently.

difference is equal to or greater than a given threshold t_r (that is an hyperparameter), then latent topic k is baptized as 'real topic' h_1, corresponding to the highest probability. Otherwise, it is considered a pseudo-topic and discarded.

Correlation-Based

The second new proposed method is based on correlation.

The Pearson correlation is preferred to the appreciated Spearman one for two reasons. First of all our data has linear dependencies [7] and in this situation Pearson is surely a good choice (see for example [4]). Secondly a preliminary test is performed comparing the two approaches and it confirmed that Pearson correlation give far better performances.

For each of the KxH couples (LT_k, RT_h) of latent topic and real topic there are D couples of values $(\tilde{\theta}_{k,d}, \mathbf{x}_{k,d})$, one for each of the D documents. Using this D-couples the Pearson coefficients ρ_h^k between each couple (LT_k, RT_h) is computed and two topics are considered *strongly correlated* if the value is greater than a value t_c (that is an hyperparameter) [8].

$$if\ \rho_h^k > t_c \qquad \rightarrow \qquad LT_k\ is\ baptised\ as\ RT_h \tag{3.7}$$

	Real topics			
Latent topics	**Health**	**Fashion**	**Celebrities**	**Animals**
1	0.82	0.20	0.75	0.07
2	0.27	0.80	0.35	0.11

Table 3.6: Toy example for correlation based method - Baptism of latent topics 1 e 2 with threshold $t_c = 0.6$.. Each cell value is the Pearson correlation coefficient between the real topic of its column and the latent topic on its raw. The latent topic 1 is ignored because it is strongly correlated to two real topic (Health and Celebrities has correlation coefficient equal to 0.82 and 0.75 respectively, both higher than the threshold, so this latent topic is a supertopic of Health and Celebrities), the latent topic 2 is baptized as "fashion", because the only coefficient that overcomes the threshold is the one for "fashion" (0.80).

If a latent topic results to be strongly correlated only to one real topic the latent topic in exam is baptized with this real topic, if instead it is strongly correlated with no real topic or more than one real topic, it is ignored as it is a pseudotopic because it is a trasversal topic or a supertopic

[7] Even though we are dealing with proportions, it is reasonable to think that in the case in which a latent topic is a particular real topic, if its proportion is double even the proportion of the correspondent real topic is double

[8] The threshold is a positive value so negative values of the correlation coefficient are always below the threshold

respectively.

3.2.3 Combination of methods

All the baptism methods can be even combined and use together. Each method M^m assign to every latent topic LT_h ($h \in \{1,...,H\}$) a real topic RT_{k_m} ($k_m \in \{1,...,K\}$) and let's indicate as assigned to RT_0 a pseudotopic. Taken two mehtods M^1 and M^2 we can define two new methods.

$M^1 \bigvee M^2$: this new method puts the two original ones, M^1 and M^2, in "OR", so the new method assigns RT_k to LT_h if and only if they agree or one of them make this assignment and the other is unsure and take it as a pseudotopic. In case of disagreement it baptise the latent topic LT_h^m as pseudotopic. Formally given $\forall h = 1,..,H$

$$\begin{aligned} M^1 &: LT_h \rightarrow RT_{k_1} \\ M^2 &: LT_h \rightarrow RT_{k_2} \end{aligned} \tag{3.8}$$

The final baptism of LT_h^m is:

$$\begin{aligned} if\ k_1 = k_2 = k \Rightarrow\ & LT_h \rightarrow RT_k \\ if\ k_i = 0 \Rightarrow\ & LT_h \rightarrow RT_{k_j}\ where\ i \neq j\ and\ i,j \in \{1,2\} \\ if\ k_i \neq 0\ \forall i \in \{1,2\}\ and\ k_1 \neq k_2 \Rightarrow\ & LT_h \rightarrow RT_0 \end{aligned}$$

In other word this method try to find a real topic even for the latent topic in which a methods is not certain using the other methods and to clean assignment where they do not agree.

$M^1 \bigwedge M^2$: this new method puts the two original ones, M^1 and M^2, in "AND", so the new method assigns RT_k to LT_h if and only if they agree. In case of disagreement it baptise the latent topic LT_h^m as pseudotopic. Formally given equation 3.8 we have:

$$\begin{aligned} if\ k_1 = k_2 = k \Rightarrow\ & LT_h \rightarrow RT_k \\ if\ k_1 \neq k_2 \Rightarrow\ & LT_h \rightarrow RT_0 \end{aligned}$$

In other word this methods accepts the baptism if and only if the two methods agree.

Both these combinations will be tested in the chapter 4 with Distribution and Top Words methods, and then, in the chapter 5 all test are performed using even the combination of these two with the ”OR” logic, as it proved itself to give better results.

3.2.4 Threshold selection

As explained in chapters 3.2.1 and 3.2.2 all baptism methods depends on a threshold that is an hyperparameter. These thresholds impact on the decision to assign a latent topic to a real topic[9] or to consider it as pseudo-topic.

1. For distribution-based method it is the threshold of the difference between the score of the first latent topic (the candidate) and the second one, listed in order of score. It is a number in range $[0; 1]$.

2. For top words-based method it is the threshold of the difference between the words count of the first latent topic (the candidate) and the second one listed in order of words count. It is an integer number.

3. For regression-based method it is again the threshold of the difference between the coefficient of the first latent topic (the candidate) and the second one listed in order of coefficients. It is a number in range $[0; +\infty)$.

4. For correlation-based method it the threshold that the coefficient has to overcome. It is a number in range $[0; 1]$.

In Experiments of paper 1, chapter 4, where just Distribution Based and Top Words methods (and their combination) are used, the thresholds are studied showing performances with different values of that thresholds.

In Experiments of paper 2, chapter 5, an automatic threshold selection method has been developed that runs automatically with the model in use. At the beginning the train set is divided randomly in 5 subset with identical number of observations, so each subset is 20% of the train set. Then recursively for five times one of the subset is selected and called ”subtest” and the remaining 4 are collapsped in a new set called ”subtrain”. Then the baptism method is trained

[9] If the measured variable is above the threshold.

on the subtrain and the performance of different metrics are tested on the subtest. This is done for different values of the thresholds and the value with the best performance is selected. The performance is calculated with these metrics [10] in order (these metrics will be explained later in chapter 3.3.2):

1. SAD
2. CWQ
3. Accuracy
4. Precision
5. Recall

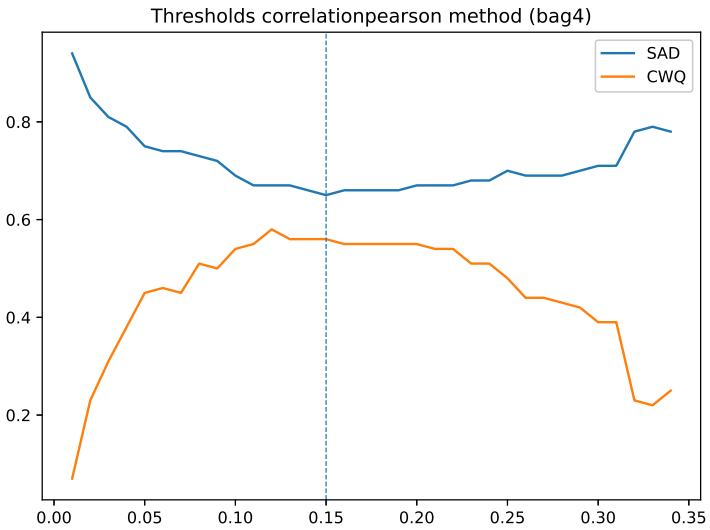

Figure 3.2: The performances with the SAD (for which lower is better) and CWQ (for which higher is better) of different values of the threshold d_c of the correlation methods are shown for the 4-th run of the cross validation. In this case the value 0.15 should be selected. The final value will be the mean of values of each run.

The threshold with best value for the first metric (SAD) is selected. If there is an ex-equo for one metric, the algorithm passes to evaluate the following one. In the (never happened) case where there is an ex-equo for all the metrics the engine selects the lower one. After running this

[10] Among new metrics that is introduced in chapter 3.3.2 the two that doesn't need the definition of hyperparamters to be used are selected

five times, the mean and the standard deviation of the thresholds is calculated and the model in use will adopt the mean as threshold to give results on the test set.

In this way the evaluation of the threshold is calculated by cross validation. However looking at the results quite always the thresholds have the same values in the different runs in the different folds and when there is a difference it is just in one fold and it is modest. This confirms the robustness of the approach.

3.2.5 Assignment of real topics to a document

Once a baptism method is defined, the procedure to assign real topics to unseen documents is the following. Let's consider a new document, its estimated vector $\hat{\theta}_{new}$, and the vector $\mathbf{B}$ obtained by one of the baptizing methods. For ease of explanation, let's consider a toy example considering a latent topic engine fitted considering $K = 8$ latent topics. A baptizing method produced the vector $\mathbf{B} =$ ("Health", "pseudo-topic", "Health", "Fashion", "Celebrities", "pseudo-topic", "Fashion", "Animals"), whereas the predicted vector of topic proportion resulted equal to $\hat{\theta}_{new} = (0.2, 0.05, 0.15, 0.02, 0.12, 0.18, 0.1, 0.18)^\intercal$.

b_k	Health	pseudo	Health	Fashion	Celebr.	pseudo	Fashion	Animals
$\hat{\theta}_{new,k}$	0.2	0.05	0.15	0.02	0.12	0.18	0.1	0.18

Table 3.7: Toy example - Classification of a new document having its predicted vector of topic proportions $\hat{\theta}_{new}$ and the vector of baptized topics $\mathbf{B}$.

Classification is performed by summing the elements in $\hat{\theta}_{new}$ having with the same label in $\mathbf{B}$. Thus, the topic-label "Health" leads to a total proportion equal to $\hat{\theta}_{new,1} + \hat{\theta}_{new,3} = 0.2 + 0.15 = 0.35$, whereas the other labels lead to $\hat{\theta}_{new,4} + \hat{\theta}_{new,7} = 0.12$ ("Fashion"), $\hat{\theta}_{new,5} = 0.12$ ("Celebrities"), $\hat{\theta}_{new,8} = 0.18$ ("Animals"), and $\hat{\theta}_{new,2} + \hat{\theta}_{new,6} = 0.23$ ("pseudo-topic"). Since pseudo-topics must be ignored, the corresponding value must be removed and value normalized so that they sum to 1. Thus obtaining: $Health = 0.45$, $Fashion = 0.16$, $Celebrities = 0.16$, $Animals = 0.23$. Given these totals, if a new document has to be classified to a unique real topic, the topic with the largest total proportion will be chosen for the final classification. In the above example, the new document is going to be classified as a "Health"-related document. This rule

can be extended to consider the case in which a “mix” (i.e., a document characterized by more than one real topic) has to be classified. Let us assume that the mix is composed by two real topics, then the two topics with the largest total proportion will be selected. In the toy example proposed in Table 3.7, the second largest total proportion is associated with the “pseudo-topic” label. When such a situation occurs, the topic with the largest total proportion among those remaining is selected as the second real topic. In our case, the new document is going to be classified with the mix (“Health”, “Animals”).

3.3 Metrics

3.3.1 Adaptation of classical classifiers’ indicators to predictions of mix

In the first paper classical indicators of classification (accuracy, precision, recall, F1) are adapted to work in a situation in which the prediction is not a class but a vector of quotes, one for each real topic. The prevision is considered correct if the topic with the higher predicted probability is exactly the real topic of the document.

3.3.2 Evaluation metrics on topics mix prediction

Evaluating mix predictions is a tricky task first of all it requires to define what is the meaning of ”*correctly identified*”. In a single topic scenario it’s trivial: if the prediction matches that topic, it’s correct; Whereas, now that a more rich corpus with a deterministic quantified mix of real topics is considered, some new metrics are needed to measure the ability of models to correctly identify topics. In a mix scenario, it could be defined in many different ways, such as:

1. the predicted topic with the greatest probability is one (or the highest) of the real ones
2. real topics are the same/or a subset of the predicted ones (despite the associated weight)
3. predicted topics are the same/or a subset of the real ones (despite the associated weight)
4. predicted topics are the same/or a subset of the real ones and the related probabilities are similar to the real ones

5. probabilities of each predicted topics are similar to the ones of real topics

In order to account all these types of evaluation, different metrics will be introduced. Before doing so, it is necessary to define two thresholds (which take value between 0 and 1): one for real topics (t_{real}) and one for predicted ones (t_{pred}). Once set these thresholds only topics with percentage greater or equal to these thresholds will be considered and for the remaining ones the score is re-normalized to 1. These thresholds are needed to define which real and predicted topics to consider to compute the metrics. The *predicted threshold* is necessary in particular in the count metrics (the first four below) because generative topic models assign values (even if very low) to all topics and thus it is necessary to exclude the tail of topic with "insignificant" score when a simple count is performed. The *real threshold* it is not strictly necessary, but since data are never perfect, it is useful even here to delete noise removing the topics with low percentages.

WTN (Wrong Topics Number) and MTN (Missed Topics Number)

Two metrics to measure how much the model "invents topics" and how much it "misses topics" will be now introduced. These metrics are suitable in particular in context where one is interested in the identification of which topic has a document regardless of the scores.

Let's define:

- R_i is the set of real topics for document i
- P_i is the set of predicted topics for document i
- W_i is the cardinality of $(P_i - R_i)$, i.e. the number of elements of P_i not in R_i for document i
- M_i is the cardinality of $(R_i - P_i)$ i.e. the number of elements of R_i not in P_i for document i
- D is the total number of documents presents in the corpus

The *Wrong Topics Number (WTN)* defines how many predicted topics are not between real ones, on average for each document, in essence how many topics the algorithm invents.

$$WTN = \frac{\sum_{i=1}^{D} W_i}{D} \tag{3.9}$$

Whereas the *Missed Topics Number (MTN)* defines how many real topics are not in predicted ones, on average for each document; in essence how many topics our algorithm misses.

$$MTN = \frac{\sum_{i=1}^{D} M_i}{D} \tag{3.10}$$

	Scores	
Topics	**Real**	**Predicted**
Animals	0.5	0.65
Sport	0.4	0.2
Fashion	0.1	0
Food	0	0.15
Sum	1	1

Table 3.8: Data used for next toy examples with 4 topics

We introduce even an unique synthetic indicator *Wrong-Missed Topics Number* (WMTN) as the mean the two values:

$$WMTN = \frac{WTN + MTN}{2} \tag{3.11}$$

WTQ (Wrong Topics Quota) and MTQ (Missed Topics Quota)

The previous metrics define how many topics are wrong or missed, but it does not take into account *how much is wrong*. It is different to miss a real topic that is 20% in the document and miss a real topic that is 80%, or to invent a topic with a score of 10% and with a score of 90%; so this second set of metrics has been developed to cover this aspect.

Let's define:

	Topic no thresholds	
Topics	**Real**	**Predicted**
Animals	YES	YES
Sport	YES	YES
Fashion	**YES**	NO
Food	NO	**YES**
Invented Number		**1**
Missed Number	**1**	

	Topic with thresholds	
Topics	**Real**	**Predicted**
Animals	YES	YES
Sport	**YES**	NO
Fashion	NO	NO
Food	NO	NO
Invented Number		**0**
Missed Number	**1**	

Table 3.9: Toy example to understand WTN and MTN, using data of table 3.8. The left table shows what happens if no thresholds are applied: WTN for this document in this case is 1, because Food is not in real topics, and MTN is 1 because Fashion have been missed. If instead *real threshold* = 0.15 and *predicted threshold* = 0.25 (right table), the real topics above this threshold are Animals and Sport, and the predicted one is only Animals, thus WTN becomes 0 and MTN remains 1.

- W_i is still the cardinality of $(P_i - R_i)$ for document i, i.e. the number of predicted topics that are not in the set of real ones
- M_i is still the cardinality of $(R_i - P_i)$ for document i, i.e. the number of real topics that are not in the set of predicted ones
- p_{ij} is the predicted quota of the j^{th} element of the set $|P_i - R_i|$, for document i
- r_{ij} is the real percentage of the j^{th} element of the set $|R_i - P_i|$, for document i
- D is the total number of documents presents in the corpus

Thus, to compute *Wrong Topics Quota*, after having identified the wrong topics as before, instead of counting the wrong predicted topics in each document and make the mean, the sum of the quotas of the wrong predicted topics in each document is computed and then the mean over all documents is performed, leading to a metric which indicates the quota of wrong topics on average.

$$WTQ = \frac{\sum_{i=1}^{D} \sum_{j=1}^{W_i} p_{ij}}{D}, \qquad p_{ij} \in (P_i - R_i) \tag{3.12}$$

Similarly *Missed Topics Quota* is computed summing together the quotas of the missed real topics, and averaging the values on all documents.

$$MTQ = \frac{\sum_{i=1}^{D} \sum_{j=1}^{M_i} r_{ij}}{D}, \qquad r_{ij} \in (R_i - P_i) \tag{3.13}$$

We introduce even an unique synthetic indicator *Wrong-Missed Topics Number* (WMTQ) as the mean the two values:

$$WMTQ = \frac{WTQ + MTQ}{2} \tag{3.14}$$

Also in this case two thresholds must be set. In this case another operation is needed: after having applied the thresholds, the values must be normalized, so that they sum to 1 (dividing each element by the sum of all elements).

	Scores no thresholds	
Topics	**Real**	**Predicted**
Animals	0.5	0.65
Sport	0.4	0.2
Fashion	**0.1**	0
Food	0	**0.15**
Invented Quota		**0.15**
Missed Quota	**0.1**	

	Scores with thresholds	
Topics	**Real**	**Predicted**
Animals	0.56	1
Sport	**0.44**	0
Fashion	0	0
Food	0	0
Invented Quota		**0**
Missed Quota	**0.44**	

Table 3.10: Toy example to understand WTQ and MTQ, using data of table 3.8. The left table shows what happens if no thresholds are applied: WTQ = 0.15 (value corresponding to Food), and MTQ = 0.1 (value corresponding to Fashion). If instead *real threshold* = 0.15 and *predicted threshold* = 0.25 (right table), after normalization the values shown in table are obtained and thus for this document and this pair of thresholds: WTQ = 0 because no topic has been invented, whereas MTQ = 0.44, because the only topic that the method has missed is Sports, and its *real* value is 0.44.

CWQ (Correct Weighted Quota)

The previous two metrics define how much a method is wrong in different ways, but in order to give an idea of how much it is right, another metric has been developed.

It takes into account, for each document, the intersection between predicted and real topic, in essence the subset of right predicted topics, and then sums together the percentage (as real topic) of each of these topics weighted by (1- the delta between the real vale and the predicted one). In this way similar values are more weighted than far values. After that the mean of these values for all document is computed and it is called CWQ. This metric wraps together the right topics and the related values, giving an estimate of how much the method predicts correct topics.

$$CWQ = \frac{\sum_{i=1}^{D} \sum_{j=1}^{RT_i} (r_{ij} * (1 - |r_{ij} - p_{ij}|))}{D}, \qquad p_{ij}, r_{ij} \in (R_i \cap P_i) \tag{3.15}$$

where:

- RT_i is the cardinality of $R_i \cap P_i$ for document i, i.e. the number of right predicted topics
- p_{ij} is the predicted quota of the j^{th} element of the set $R_i \cap P_i$, for document i
- r_{ij} is the real percentage of the j^{th} element of the set $R_i \cap P_i$, for document i
- D is the total number of documents presents in the corpus

Topics	Real	Predicted	Delta	Weigthed
Animals	**0.5**	0.65	0.15	0.425
Sport	**0.4**	0.2	0.2	0.32
Fashion	0.1	0	N/A	N/A
Food	0	0.15	N/A	N/A
			CWQ	0.745

Table 3.11: Toy example to better understand CWQ, using the data of 3.8. The intersection between the real topics set and the predicted one is Animals and Sports. In this case this metric does not need thresholds, so CWQ can be directly computed. The delta has to be computed and then each real values has to be weighted for 1-delta, thus for this document CWQ = $0.5 * (1 - |0.65 - 0.5|) + 0.4 * (1 - |0.4 - 0.2|) = 0.745$, meaning it is 74.5% right.

SAD (Scaled angular distance)

As we said in chapter 2.4.1 the kind of data we are dealing with are compositional data, which are represented as points on a simplex. So the idea is to use a metric on a simplex in order to add another valuable evaluation criterion.

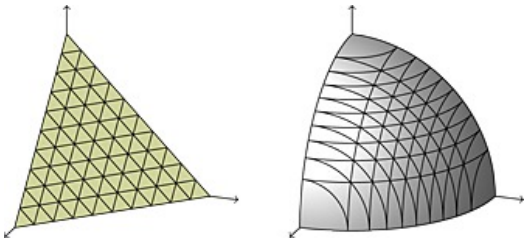

Figure 3.3: The geodesic distance of the projection of points on the slice of the sphere with radius $2/\pi$. Here, it is shown the case of three real topics.

Let's consider the following two points: **p** and **r**. **p** is the vector of percentages related to the predicted topics and **r** is the vector of percentages related to the known topics of a document. We introduce a new metrics as follow:

$$d_{SA}(p,r) = \frac{2}{\pi} * arccos\left(\sum_{h=1}^{H} \sqrt{\frac{p_h^2}{\sum_{j=1}^{H} p_j^2}} \sqrt{\frac{r_h^2}{\sum_{j=1}^{H} r_j^2}}\right) \tag{3.16}$$

where H is the length of **r** and **p**, in our case 39, the number of real topics.

d_{SA} ranges from 0 to 1, with 0 meaning it is 0% wrong, 0.5 meaning it is 50% wrong, and 1 meaning it is 100% wrong. See Table 3.12 for an example. This distance can be seen as a re-scaled angular distance.

The final value of a metrics (called for this reason SAD = Scaled Angular Distance) will be the mean of these distances between real and predicted topics within each document over all

documents.

$$SAD = \frac{\sum_{i=1}^{D} d_{SA}(p_i, r_i)}{D} \tag{3.17}$$

Topics	Real values	Predicted values
r_1	0.5	1
r_2	0	0
r_3	0.5	0
Total	1	1

Table 3.12: Toy example of SAD metric. In this simple case, real and predicted topics half disagreeing. If a method correctly identifies half of the topics, then: $d_{SA}(r,p) = \frac{2}{\pi} * arccos(\sqrt{0.5^2/(0.5^2+0.5^2)} * 1 + 0 + \sqrt{0.5^2/(0.5^2+0.5^2)} * 0) = \frac{2}{\pi} * arccos(\sqrt{\frac{1}{2}}) = \frac{2}{\pi} * \frac{\pi}{4} = 0.5$.

3.3.3 Topic score thresholds

As the prediction gives not null scores quite always to all the latent topics, the metrics which analyse the "missed" and "invented" topics (MTN, WTN, MTQ and WTQ) cannot work without filters: we would have no "missed" topics and a lot of "invented" ones. So it is mandatory to define a threshold t_{pred} ($\in [0,1]$) for predicted topics and consider, in the prediction, only predicted topics with a score equal or above this threshold renormalizing to 1 these scores and putting all the scores of other topics to zero. In other words we consider valid predicted topics just the ones that the model consider "important". Consequently in this case it has been decided to use a similar approach even for real topics, defining a threshold t_{real} ($\in [0,1]$) for real topics, so in the same way we consider just topics with a percentage equal or above this threshold.

For t_{real} all thresholds from 0 to 0.2 with step 0.01 have been analysed on the overall dataset of 6 million documents for higher robustness and it has been counted how many documents have the maximum score below the threshold. With 0.18 we have about 36'000 documents, with 0.17 we have only about 3700 documents (meaning about the 0.05% of all documents), as it can be seen in Figure 3.4. In the subset of 200'000 documents used for experiments, the number of documents below 0.18 is about 500, whereas there are none below 0.17. So we select t_{real} according to these results:

$$t_{real} = 0.17 \tag{3.18}$$

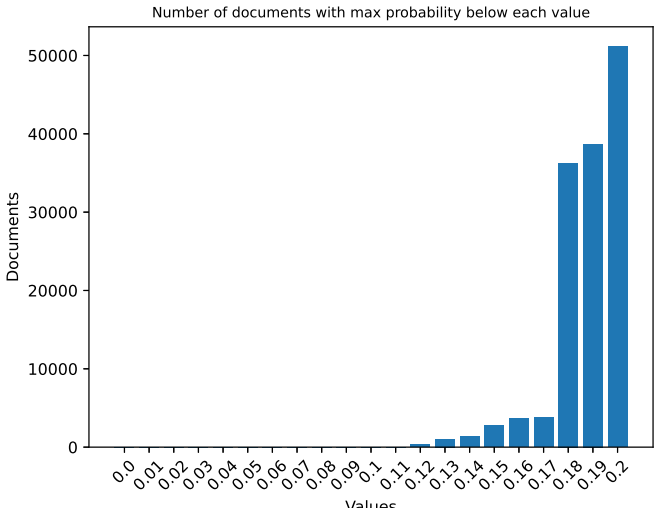

Figure 3.4: Distribution of documents with maximum score below each threshold

To find the best value of t_{pred} we define the optimus value for each couple (MTN-WTN and MTQ-WTQ) so that it balances the two values. In fact low values of the threshold favor the first metrics (MTN and MTQ), on the other side high values of the threshold favor the last ones (WTN and WTQ). So we are looking for the value that makes the two metrics as low as possible together. In practice for each method and for each pair of metrics, a threshold has been chosen with this criterion: the value which minimize the maximum between the two metrics. Thus, in the end two thresholds for each method have been identified (one for MTN-WTN and one for MTQ-WTQ). In other words, called $WTN(t_{pred})$ and $MTN(t_{pred})$ the values of the two metrics with the threshold equal to t_{pred}, the selected value of this threshold $\hat{t}_{pred}$ is:

$$\hat{t}_{pred} = \underset{t_{pred} \in [0,1]}{\arg\min} \left[max(WTN(t_{pred}), MTN(t_{pred})) \right] \tag{3.19}$$

3.4 Dataset preparations

3.4.1 Noise removal

To perform our methodology a dataset composed by a list of documents (i.e. texts) is needed, each with a list of topics, each with a proportion (to sum 1 in each document). Some preprocess procedures are needed to proceed.

Cleaning texts

First of all texts have been processed so that html tags, wikimedia markup language parts and stopwords have been removed.

Mono-topic and mixed-topic dataset

The dataset is divided in:

- mono-topic subset (i.e. the list of text where there is just one topic associated, in cases in which one topic had percentage greater or equal to 90% this document is considered as monotopic with this topic as the only topic. This subset contains $\sim$ 1 Million of articles.)
- mixed-topic subset (all other texts).

The reason why a mono-topic subset has been created is that it is useful for three purposes:

1. to measure classical indicators (accuracy, precision and recall)
2. to use "Top words" and "Distribution" baptism methods
3. to perform the following "Filter of Words" 3.4.1 simplification of text

Filtering of words

The amount of articles is huge and this would lead to high computational load. The key idea is to keep only relevant words assuming that they are sufficient to capture topics of texts. So, using the subset of mono-topic texts of the train set, a frequency analysis of the words for each topic has been performed, creating for each topic the list of words ordered by their numerousness in that topic. Defining a threshold N, only the N top words of each topic have been selected and all texts have been purged from words that don't belong to the first N words of any topic. Called t_h the H topics (h=,1...,H) and called $w_{h,i}$ the i-th word of the topic t_h after ordering the word on the basis of their presence in that topic, the dictionary $DICT$ is composed just the subset of words made by:

$$DICT = (w_{1,1}, ..., w_{1,N}, w_{2,1}, ..., w_{2,N}, ..., w_{H,1}, ..., w_{H,N})$$

that is composed by (HxN) words or less (if there are words that belongs to more than one topic). In all document just words that belongs to DICT are kept.

In this research it is selected N=30.

3.4.2 TF-IDF

While Topic Model can be feeded by simple texts, to use regressors, a transformation of text data in numeric vector has been necessary, so TF-IDF has been applied because it reflects the importance of each word for each document inside a corpus of documents giving much weight to words that are frequent in the document but with a low presence in all other documents (in order to avoid giving importance to words frequent in all texts, and thus not discriminatory). It is defined as

$$tfidf_{ij} = tf_{ij} * idf_i \tag{3.20}$$

$$tf_{ij} = \frac{n_{ij}}{|d_j|} \tag{3.21}$$

$$idf_i = log(\frac{D}{df_i}) \tag{3.22}$$

where:

- n_{ij} is the frequency of term i in document j
- $|d_j|$ is the total number of terms in document j
- D is the total number of documents presents in the corpus
- df_i is the document frequency of term i (i.e. the number of documents in which this term is present)

There are some variations of the idf formula. The one used by the library that has been utilized, is the following:

$$idf_i = log(\frac{1+D}{1+df_i}) + 1. \tag{3.23}$$

The effect of adding 1 to the end of the equation above is that terms with zero idf, i.e., terms that occur in all documents in a training set, will not be entirely ignored. The addition of 1 to the numerator and denominator of the idf is done to prevents zero divisions.

Chapter 4

Experiments of Paper 1

This chapter presents experiments described in the first paper and it focuses on two popular TM methods, namely latent Dirichlet allocation (LDA, [6]) and correlated topic model (CTM, [5]). Both methods assume that documents are probability distributions over the topics, whereas topics are probability distributions over the set of words composing the corpus. The main difference between the two methods lies in the prior distribution that is assumed for the probability vector of the topic distributions, which results in greater computational tractability as well as ease of interpretation for LDA vs. a more flexible correlation structure between topics for CTM.

The two TM methods are comparatively studied with a threefold objective. First, they are deployed in a big data scenario with the aim of comparing their performance by means of several widespread indicators. Second, supposing that each document of a corpus is characterized by a single real topic, two new methods have been proposed in order to automatically classify documents with respect to their real topic. In particular, LDA and CTM have been chosen as (latent) topic model engines, and then the two newly proposed methods for topic labelling have been applied, that is to *baptize* one of the latent topics with a single real topic name. Third, the proposed methods have been tested as multi-class classification tools under the more realistic assumption that a document is characterized by more than one real topic.

4.1 Data collection and dataset creation

In this work a crawler on monotematic websites is used, identifying sub-homepages to get article with a mix of two topics as explained in chapter 3.1.1 It has been considered a situation with $H = 4$ real topics (i.e., specialized websites) and two mixed real topics (specialized websites with subsections), as shown in Table 4.1. See Appendix C for a complete list of the websites.

(Mixed) Real Topics	N. of docs
Health	6.868
Animals	3.914
Celebrities	14.095
Fashion	14.064
Health - Animals	279
Celebrities - Fashion	227

Table 4.1: Dataset structure. Real topics and mixed real topics (column 1). Number of downloaded documents for each (mixed) real topic (column 2).

The set of downloaded documents (i.e., the final corpus $\mathscr{C}$) has dimension ≈ 39.500. All texts are in Italian. A common pre-processing phase has been performed on the corpus. Firstly, all texts with languages other than Italian, repeated texts, and texts with less than 200 characters have been neglected. Secondly, a set of trivial strings have been removed on each text. Thirdly, Italian stopwords, punctuation, and double spaces have been removed. In addition, words that are shared by two or more real topics are not considered in the analysis. Figure 4.1 shows the texts' length before and after the pre-processing phase for each (mixed) real topic.

4.2 Experimental setting and results

In this Section, the results of three experiments on the dataset described in the previous section are presented. In particular, the corpus $\mathscr{C}$ is used for accomplishing the three main objectives of the present work. In the first instance (Section 4.2.1), the whole corpus $\mathscr{C}$ (all six rows of Table 4.1) is used to test LDA and CTM in a big data scenario. Secondly, the texts labeled with a unique real topic (first four rows of Table 4.1) are used for testing the distribution-based and top words-based methods as classifiers with LDA and CTM being used as latent topic engines (Section 4.2.2). Lastly, the texts associated with mixed real topics (last two rows of Table 4.1)

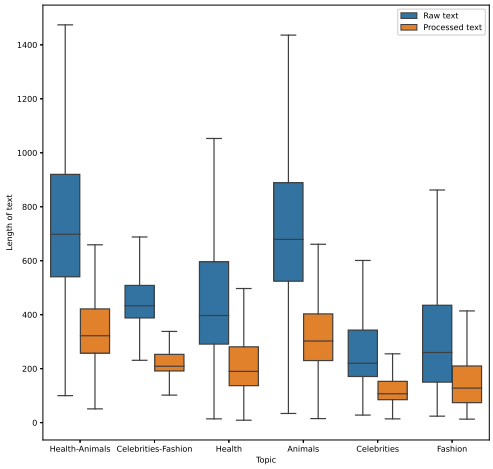

Figure 4.1: Box plot representation of the length of texts before (blue) and after (orange) the pre-processing activity.

are used to conduct a preliminary study with the aim of understanding the ability of the two novel classification methods as multi-classifiers, still with LDA and CTM being used as latent topic engines (Section 4.2.3).

In all experiments, the corpus $\mathscr{C}$ is divided into two disjoint parts, namely a training set (composed of 80% of documents), and a test set (the remaining 20% of documents). The random splitting is performed by stratifying over real topics.

All experiments are performed in Python, by using the `tomotopy`[1] library to estimate both LDA and CTM models. All elaborations have been carried out with an Intel Core i7 with RAM 16 Gb.

4.2.1 LDA and CTM in a big data scenario

The performance of LDA and CTM in our big data scenario are evaluated resorting to the perplexity and coherence metrics presented in Section 2.3. A number K of latent topics ranging from 2 to 16 has been considered, thus including also the "true" number of topics considered in generating the corpus, i.e., the value 4.

To select the best value of K from a predictive perspective, the perplexity measure computed

[1] https://bab2min.github.io/tomotopy/

on the unseen corpus, that is our test set, has been taken into account. Figure 4.2 shows the perplexity of the two models for several values of K. The perplexity decreases as the number of topics increases, thus suggesting to prefer the value $K = 16$ both for LDA and CTM models. Though, from Figure 4.2 it clearly emerges that LDA performs far better than CTM for any value of K.

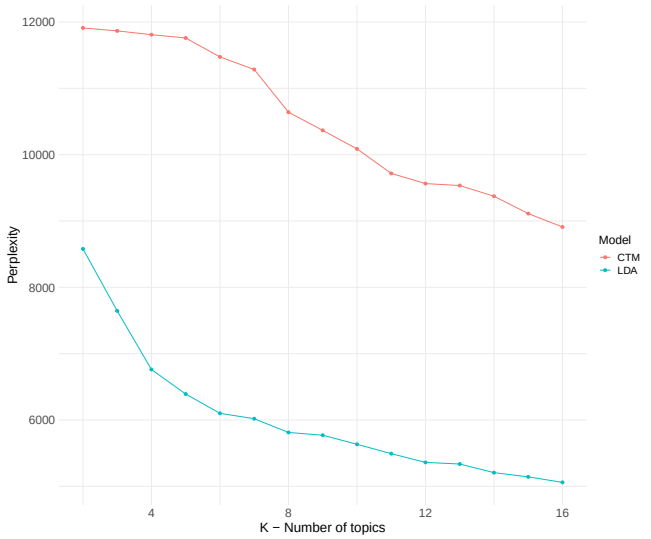

Figure 4.2: Big data scenario. Perplexity of LDA and CTM as K increases.

The quality of the recovered topics can be inspected by looking at the most probable words for each model (e.g., inspecting the word clouds as in Figure 4.3), and accordingly assigning a label to each latent topic in the light of these words. Table D.1 in Appendix D reports the 20 most probable words for the LDA model. It emerges that LDA identifies distinct topics that human judgment can hardly categorize in the four considered real topics "Health", "Animals", "Celebrities", and "Fashion". In particular, LDA recognizes some new topics (e.g., topic 2 is related to "Music", topics 3 and 6 deal with different aspects of "Beauty" routines) and splits some real topics into subtopics (e.g., topics 4, 15, and 16 split the Animal category into "Dogs", "Non-pets", and "Cats", respectively).

Contrarily, by inspecting the most probable words for the CTM model with $K = 16$ latent topics (Table D.2 in Appendix D), it is evident that there is no such clear semantic homogeneity in the recovered topics.

Interestingly, a value of $K = 16$ much larger than 4 (i.e., our ground truth) seems to perform better with both the LDA and the CTM models. Indeed, this result is coherent with the conclu-

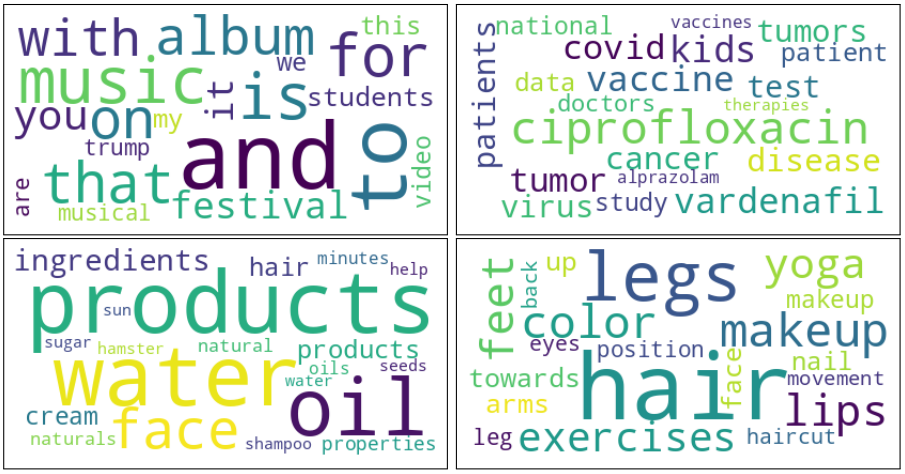

Figure 4.3: Word clouds representing four LDA latent topics. Detected latent topics refer to "Music" (topic 2), "Health" (topic 7), and different aspects of "Beauty" (topics 3 and 6).

sion of some authors affirming that the perplexity measure often selects a number of topics that is too large [16, 28]. For this reason, also the coherence results for four values of K has been inspected. For each $K \in \{4, 8, 12, 16\}$, the coherence measures (applying averages and standard deviations over topics of Equations (2.11), (2.12), and (2.13)) of the two models on the training corpus have been computed, by considering the $M = 10$ most probable words within each topic. Table 4.2 summarizes the coherence results by model and number of topics K. By inspecting Table 4.2, one can note that topics generated by the LDA model with $K = 4$ are considered the most coherent by all the coherence measures, since they are characterized by the largest mean and smallest standard deviation. Contrarily, the number of topics maximizing the coherence measures in the CTM varies between 12 and 16.

In conclusion, the LDA model seems to reliably identify topics both with $K = 4$ and $K = 16$, with a different granularity level (e.g., with more latent topics, it can discover "new" relevant topics in the corpus), whereas the CTM points to a larger number of topics that are not semantically homogeneous.

4.2.2 Automatic identification of unique real topics

The distribution-based method (hereafter, D) and the top words-based method (T) described in Sections 3.2.1 and 3.2.1 are tested as automatic classification techniques for the identification of real topics. In this context, LDA and CTM are used as latent topic engines for both methods.

Model	K	C_{UCI}	C_{NMPI}	C_{UMass}
LDA	4	**0.76** (0.36)	**0.11** (0.03)	**-1.41** (0.4)
	8	0.45 (1.13)	0.1 (0.05)	-1.99 (1.12)
	12	0.68 (1.01)	**0.11** (0.05)	-1.99 (0.89)
	16	0.29 (1.62)	0.1 (0.09)	-2.29 (1.27)
CTM	4	-3.54 (1.86)	-0.17 (0.07)	-4.4 (1.79)
	8	-1.07 (2.14)	-0.01 (0.12)	-2.85 (1.56)
	12	**-0.49** (1.78)	**0.03** (0.11)	**-2.17** (0.85)
	16	-0.69 (1.81)	**0.03** (0.1)	-2.39 (0.99)

Table 4.2: Big data scenario. Mean and standard deviation (in parenthesis) of coherence measures stratified by model and number of topics. Best values are reported in bold.

Furthermore, in order to label a text, also two different combinations of the two methods have been tested. More precisely, if the two methods strictly agree on the topic identification, then that topic is assigned to the text, otherwise no topic is assigned (i.e., the topic is labelled as pseudo-topic). From now on, this combination is called $D \bigwedge T$ approach. A further combination is also considered, which assigns a topic whenever the two methods agree on the topic (as in the $D \bigwedge T$ approach), but also when one method identifies a topic while the other method detects a pseudo-topic. In all other cases (i.e., the two methods disagree on the topic or they both identify a pseudo-topic), the topic is labeled as pseudo-topic. This approach is denoted by $D \bigvee T$.

The performance of the D and T methods depends on different parameter choices. Among these choices, one of the most relevant is the selection of the thresholds for defining whether a text is associated with a real topic or a pseudo-topic. For this reason, a sensitivity study has been performed to ascertain the influence of parameters t_d and t_{tw} defined in Algorithms 1 and 2. More precisely, the classification performances considering $t_d \in \{0, 0.1, 0.2, 0.3\}$ for the D method, $t_{tw} \in \{1, 2, 3\}$ for the T method, and $K \in \{4, 8, 12, 16\}$ have been compared.

Table 4.3 shows the global accuracy of D and T methods. It emerges that considering the LDA as latent topic engine helps the whole set of methods to get larger (i.e., better) values with respect to the use of the CTM model for any values of t_d and t_{tw}. The performance of the D method together with LDA seems to be independent with respect to either K and t_d, with the only exception $K = 4$. When considering the T method combined with LDA, a careful selection of t_{tw} should be done: a larger value of t_{tw} leads to smaller values of global accuracy. This is even more evident if we also consider a larger value of K. More generally, the farther K from

		Distribution				Top-words		
		t_d				t_{tw}		
	K	**0**	**0.1**	**0.2**	**0.3**	**1**	**2**	**3**
LDA	4	0.72	0.77	0.77	0.77	0.77	0.77	0.77
	8	**0.85**	**0.85**	0.84	0.78	0.76	0.73	0.73
	12	0.82	**0.85**	**0.85**	**0.85**	0.77	0.73	0.49
	16	0.84	0.84	**0.85**	**0.85**	**0.79**	0.76	0.46
CTM	4	0.43	0.00	0.00	0.00	0.10	0.10	0.00
	8	0.46	0.36	0.00	0.00	0.37	0.48	0.36
	12	0.49	0.40	0.54	0.00	0.45	0.44	0.44
	16	0.58	0.59	**0.60**	0.58	**0.62**	0.45	0.37

Table 4.3: Global accuracy for D and T methods. Best values are reported in bold.

the ground truth, the more strict the threshold t_{tw} should be.

		Distribution				Top-words		
		t_d				t_{tw}		
	K	**0**	**0.10**	**0.20**	**0.30**	**1**	**2**	**3**
LDA	4	0.57	0.58	0.58	0.58	0.75	0.75	0.75
	8	**0.84**	**0.84**	0.81	0.75	0.75	0.73	0.73
	12	0.79	0.83	0.82	0.83	0.76	0.76	0.50
	16	0.83	0.83	**0.84**	**0.84**	**0.78**	0.76	0.47
CTM	4	0.25	0.00	0.00	0.00	0.08	0.05	0.00
	8	0.32	0.13	0.00	0.00	0.38	0.40	0.13
	12	0.40	0.29	0.31	0.00	0.45	0.28	0.28
	16	0.43	0.47	**0.48**	0.34	**0.66**	0.43	0.36

Table 4.4: Macro *F1-score* for D and T methods. Best values are reported in bold.

The values of macro *F1-score* are also reported in Table 4.4. Focusing on the LDA model as latent topic engine, it can be noticed that the D method gets larger values of macro *F1-score* with $K = 8$ and $K = 16$. If we focus on the T method, the situation is slightly different. In fact, larger values are reached only if $K = 16$ is considered. Globally, the D method obtains higher values of macro *F1-score* with respect to the T method.

Given these results, the focus has been fallen on the comparison of $D \bigvee T$ and $D \bigwedge T$ only considering LDA as latent topics engine. From the results in Table 4.5, the combination of the D and T methods does not lead to better performance. $D \bigvee T$ reaches the same results as the D method, therefore there is no any advantage in using the two methods together. Moreover, the results lead us to assume that the D is the method ruling the final decision of the $\bigvee$ logical

		D ⋁ T				D ⋀ T			
		t_d				t_d			
K	**t_{tw}**	**0**	**0.1**	**0.2**	**0.3**	**0**	**0.1**	**0.2**	**0.3**
4	1	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**
	2	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**
	3	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**
8	1	**0.85**	**0.85**	0.84	0.78	**0.76**	**0.76**	**0.76**	**0.76**
	2	**0.85**	**0.85**	0.84	0.78	0.73	0.73	0.73	0.73
	3	**0.85**	**0.85**	0.84	0.78	0.73	0.73	0.73	0.73
12	1	**0.85**	0.84	0.83	0.73	**0.77**	**0.80**	**0.80**	**0.80**
	2	0.82	**0.85**	**0.85**	**0.85**	0.83	0.73	0.73	0.73
	3	0.82	**0.85**	**0.85**	**0.85**	0.49	0.49	0.49	0.49
16	1	0.84	0.84	**0.85**	**0.85**	**0.79**	**0.79**	**0.79**	**0.79**
	2	0.84	0.84	**0.85**	**0.85**	0.76	0.76	0.76	0.76
	3	0.84	0.84	**0.85**	**0.85**	0.46	0.46	0.46	0.46

Table 4.5: Global accuracy for $D \bigvee T$ and $D \bigwedge T$ methods for the LDA model. Best values are reported in bold.

operator.

4.2.3 Test on the identification of mixed real topics

Since a document may be described by more than one topic, there is an urgent need to develop suitable approaches that recognize mixed real topics. For this purpose, distribution-based and T methods are tested as multi-class classification tools. Similarly to Section 4.2.2, also here LDA and CTM are used as latent topic engines in both methods. $D \bigwedge T$ and $D \bigvee T$ are also tested. Again, all the combinations between $K \in \{4, 8, 12, 16\}$, $t_d \in \{0, 0.1, 0.2, 0.3\}$, and $t_{tw} \in \{1, 2, 3\}$ are tested. As reported in Table 4.1, two mixed topics have been considered, that are "Health" - "Animals" and "Celebrities" - "Fashion".

Table 4.6 summarizes the main results in terms of global accuracy. Similar to the previous case, the performance of the D method with LDA as the topic engine is independent of K and t_d, except for $K = 4$. This confirm the idea that K has to be greater than the number of real topics (at least double as it is shown here). For the T methods the behaviour is different: increasing the number K of latent topics affects the performance of the T method (with LDA). When a value of K higher than 4 is set, results start to deteriorate, because of the limits of T which counts the words and falls into difficulties when latent topic number increases in presence of mix of topics.

		Distribution				Top-words		
		t_d				t_{tw}		
	K	**0**	**0.1**	**0.2**	**0.3**	**1**	**2**	**3**
LDA	4	0.44	0.44	0.44	0.44	**0.80**	**0.80**	**0.80**
	8	0.95	0.95	0.96	0.64	0.60	0.61	0.61
	12	0.92	0.97	**0.98**	0.95	0.55	0.61	0.17
	16	0.91	0.91	0.92	0.92	0.52	0.54	0.13
CTM	4	**0.45**	0.00	0.00	0.00	0.55	0.00	0.00
	8	**0.45**	0.00	0.00	0.00	0.70	0.41	0.00
	12	**0.45**	**0.45**	**0.45**	0.00	0.48	0.00	0.00
	16	**0.45**	**0.45**	**0.45**	**0.45**	**0.94**	0.36	0.47

Table 4.6: Global accuracy for D and T methods. In this case, two mixed topics are considered. Best values are reported in bold.

		D $\bigvee$ T				D $\bigwedge$ T			
		t_d				t_d			
K	t_{tw}	**0**	**0.1**	**0.2**	**0.3**	**0**	**0.1**	**0.2**	**0.3**
4	1	0.44	**0.80**	**0.80**	**0.80**	**0.44**	**0.44**	**0.44**	**0.44**
	2	0.44	**0.80**	**0.80**	**0.80**	**0.44**	**0.44**	**0.44**	**0.44**
	3	0.44	**0.80**	**0.80**	**0.80**	**0.44**	**0.44**	**0.44**	**0.44**
8	1	0.95	0.95	**0.96**	0.64	0.60	0.60	0.60	0.60
	2	0.95	0.95	**0.96**	0.64	**0.61**	**0.61**	**0.61**	**0.61**
	3	0.95	0.95	**0.96**	0.64	**0.61**	**0.61**	**0.61**	**0.61**
12	1	0.97	0.93	0.93	0.89	**0.61**	**0.61**	**0.61**	**0.61**
	2	0.92	0.97	**0.98**	0.95	**0.61**	**0.61**	**0.61**	**0.61**
	3	0.92	0.97	**0.98**	0.95	0.17	0.17	0.17	0.17
16	1	0.91	0.91	**0.92**	**0.92**	0.52	0.52	0.52	0.52
	2	0.91	0.91	**0.92**	**0.92**	**0.54**	**0.54**	**0.54**	**0.54**
	3	0.91	0.91	**0.92**	**0.92**	0.13	0.13	0.13	0.13

Table 4.7: Global accuracy for $D \bigvee T$ and $D \bigwedge T$ methods for the LDA model. In this case, two mixed topics are considered. Best values are reported in bold.

Also in this case, the use of CTM as the topic engine is not advisable.

As in Section 4.2.2, the two approaches $D \bigvee T$ and $D \bigwedge T$ have been considered only with LDA as latent topic engine. Table 4.7 reports the values of global accuracy. Differently from the previous case, the combination of the D and T methods leads to slightly better performance with respect to using only one of the two methods. In conclusion, it seems that $D \bigvee T$ clearly outperforms $D \bigwedge T$, and a larger value of K is preferable for classifying documents characterized by a mix of real topics.

Chapter 5

Experiments of Paper 2

This chapter presents experiments described in the second paper and it concludes the research.

All the elements to perform comparisons between different TMs are available: a very rich dataset made of documents with known mix of topics and new metrics suitable to evaluate performances on such a dataset.

For computational reason the first three experiments are performed on a random subset of 200'000 documents selected from the 6 millions of documents of the original dataset.

In the first experiment, before making the final comparison, it has been measured if the new automatic supervised labelling methods proposed in this research work correctly or not. Therefore, in the first experiment, it has been tested if and how much they are distinct from a random selection and even from a "cheated" random selection where for each document one of the real topic is "suggested" to the random model (called from now on "doped random model"). As a further insight an analogue evaluation of a range of SLMs is performed (including linear regression, AdaBoost regression, bagging regression, gradient boosting regression, random forest regression, and K-nearest neighbor regression). This test can carry two important achievements: validation of all new metrics and satisfy the curiosity to see of how much these specialized method are better then the proposed automatic labelling methods (with the hope to find that they are in the same order of magnitude).

In the second experiment, it has been applied the complete methodology to compare the four latent topic models presented in chapter 2.2.1 using all labelling engines introduced in chapter 3.2. The ultimate goal is to use for the first time our new comparative framework for generative

topic models.

In the third experiment a tuning of the hyperparameters of the best model identified in the second experiment, is performed.

In the fourth experiment, the selected method with the best hyperparameters found in the third experiment is tested over all the dataset of 6 millions of documents.

Both classification metrics (just on monotopic test set) and new metrics (on all test set) are used to evaluate results.

All experiments have been carried out with an AMD Ryzen 5 1500X Quad-Core Processor with 8 logical units, and RAM 64GB.

5.1 Datasets

The dataset has been created downloading text from English wikipedia dump (see section 3.1.2 for more details) and assigning to each article a mixture of the 39 main topics (see section 3.1.2). The overall amount of texts is about 6 million, about 1 million of which are mono-topic and the remaining about 5 million are mixed-topic. Then the mono-topic dataset has been extracted to find the top words for each topic. Then each text has been processed with the method described in section 3.4.1, keeping only the words in the dataset built with the 30 top words of each topics. Meaning that a maximum of 30x39 = 1'170 words are considered. In particular, the total number of unique top words used has been 649.

The focus of this research is on the process and algorithms, therefore for the first three experiments a subset of 200'000 articles have been used[1]. Since the number of monotopic texts is about 15% of the overall it has been decided to compose this subset with 85% of texts with topic mix and 15% of monotopic texts (stratifying by topic) to reflect reality. In cases in which the number of texts for a topic was not sufficient, all the texts of that topic have been considered, and the remaining number of texts needed has been spread over other topics. When only mono-topic texts need to be considered (for example for distribution and topwords baptism or to measure accuracy, precision and recall), only the mono-topic part of this subset is considered. In essence about 30'000 texts.

[1]The fourth experiments has tested the complete procedure on the full dataset.

To perform the tests dataset has been divided in train e test with proportion respectively of 80% and 20% with a random selection stratified by mixed-topic set and each monotopic set.

5.2 Experimental approach: cross validation

All experiments are performed using a k-fold cross validation. The corpus is divided in 5 subset (20% each) and then evaluated five times, each time using one of the 5 subset as test set and the other 4 as training set. So for each metrics there are 5 values and it is calculated the mean and the standard deviation and box plot graphics are produced.

To evaluate if a value is significantly better than another one taken as benchmark, the distance between the two means in terms of standard deviation of the latter is measured, calling it ”distance in standard deviations” (DSTD). We have to consider that usually two standard deviations is considered enough to state that the two values are statistically different and that in CERN the Higgs Boson discovery has been validated with 5 standard deviations (see [9]).

All models are employed with default parameters, and detailed parameters are available in Appendix B.

5.3 Experiment 1: New methodologies using LDA

In the first experiment, tests to validate the proposed baptism methods (regression-based, correlation-based, distribution-based, top words-based, and D $\bigvee$ T methods) as valid real topic identification methodologies have been performed. It is used LDA as the latent topic engine. The selection of LDA as the latent topic engine is based on results obtained in chapter 4. The number of iterations is set to 1000.

The main idea is to compare the results of our methods both with the Random, to see if they are far from casual topic selection, and with Doped Random to see if they are better even compared with a random model helped by a suggestion. If both statements are true the new methods can be considered valid methods to identify real topic and can be used to compare different topic models in their ability to identify real topics in texts (that will be done in the second experiment in section 5.4).

Furthermore, it has been decided to verify even correspondent performances of SLMs to see how much they are better than our methods and if values of the new metrics are coherent with the classical classification metrics as an additional validation of the new metrics them self.

All methods used (random models, the new methods proposed and the SLMs) have the same output: the distribution of predicted real topics over each document.

5.3.1 Models

As the focus of this research is not to find the best model but to test the methodological framework, all models are implemented with the default parameters except for few parameters which enables probability estimation or reduce computational load to acceptable values (the parameters are described in Appendix B, the ones different from defaults are highlighted with the symbol "*").

Random Models

As said at the beginning of this chapter, two different random models are used as "validator" benchmarks for proposed labelling methods.

Pure random: This first method takes, for each document, an integer N randomly between 1 and 39 (which will be the "predicted" number of topic) and then it takes randomly N topics and assign to each of them a random value between 0 and 1 so that the sum of the N values will be 1[2]. It will be called just "Random" from now on.

Doped random: In the second random method a cheating is performed by giving it the one of the knows real topic of each document. So for each document the model take randomly one of the known real topics of the document, then choose randomly an integer M between 0 and 38 and than choose randomly M topics between remaining ones adding them the known topic given at the beginning. Therefore having a the list of M+1 topics, it assigns to them numbers between 0 and 1 (so that they sum to 1) as the Pure Random.

[2]It is better not to let just assign a random values to all 39 topics because it would carry too diluted values for all topics.

Topic Model

The focus of this work is not to find the best topic model in the world so it is not so important what topic model is selected for this comparison. LDA (Tomotopy Python library [3] with default parameters as shown in Appendix B) is selected because it has a significantly lower computational load and it demonstrated to be better in the first paper (see chapters 4.2.2 and 4.2.3). It is used with 1'000 iterations. It is configured with the number k of topics equal to four times the number of real topics (there are 39 real topics) following the idea of the first paper (see chapter 4.2). So k = 4x39 = 156.

For LDA, bag-of-words vectors are given as input, so words' weights only depend on the frequency of each word inside each document. Details of the steps are given below:

1. LDA has been trained with k = 156 topics over the mixed training set with default parameters and 1000 iterations.

2. Based on the five aforementioned methods, still working on training set (just mono-topic for methods 1, 2 and 3 and all the training set for methods 4 and 5), each latent topic has been baptized as a specific real topic or as pseudo topic.

3. Model has been applied on a test set (with also mixed texts) to predict latent topics for each unseen document. In this way a vector of probabilities assigned to each latent topic has been obtained.

4. For each unseen document all latent topic baptised with the same real topic have been summed together and all the pseudotopics have been removed. So a distribution of real topic for each document has been obtained. As far as the pseudotopics have been removed, the sum was not 1 anymore, so a simple renormalization to 1 (dividing each element by the sum of all elements) has been performed. In this way for each baptism method a vector of probabilities assigned to each real topic has been obtained. See chapter 3.2.5 for more details.

Notice that even if distribution and topwords baptisms is based on the mono-topic subset, the LDA has been trained on all the mixed train set.

[3] https://bab2min.github.io/tomotopy/

Baptism methods

All 4 baptism methods have been compared, together with the mixed method $D \bigvee T$[4] (chapter 3.2.3). In details here the list of baptism methods used:

1. Topwords-based (T)
2. Distribution-based (D)
3. Distribution OR Topwords ($D \bigvee T$)
4. Regression-based (R)
5. Correlation-based (C)

The selection of the threshold of each method is calculated dynamically as part of the method as explained in chapter 3.2.4.

Regressors

To perform a test using well known statistical learning models a subset of regressors has been selected. In detail, below there is the list of the used one. Between brackets the Python library used is given while the details about hyperparameters can be found in Appendix B).

- Linear Regression (Scikit-learn)
- Ada Boosting (Scikit-learn)
- Bagging Regressor (Scikit-learn)
- Gradient Boosting Regressor (Scikit-learn)
- Random Forest Regressor (Scikit-learn)
- KNN Regressor (Scikit-learn)

[4] It has been proved to give good results in the first paper (in chapters 4.2.2 and 4.2.3)

For supervised methods, text data is transformed into a common document-term matrix with *tf-idf* weights (refer to Section 3.4.2 for specifics). In each supervised regression model, an ensemble is employed. For example, in the case of gradient boosting regression, the ensemble size is set to 39, corresponding to the number of real topics. Each member of the ensemble receives the document-term matrix and the proportion of each real topic (dependent variable). Essentially, a model is estimated for each real topic. The ensemble is subsequently used for prediction. As a result, predictions are normalized to the range [0, 1], yielding a vector of proportions with a size of 39.

Then for each document of the train set the tf-idf vectors are given as input to regressors models (independent variables) together with the vector of quotas of real topics (dependent variables). After the training, the model will produce for each document of the test set a vector of predicted quotas of real topics to be compared with the vector of actual quotas of real topics.

5.3.2 Results with classification metrics

Firstly, the automatic topic models are compared with the two random models using all classification metrics, keeping in mind that for global accuracy, precision, and recall just monotopic documents are used as test set. To correctly use these performance metrics, in all cases, the real topic with the highest value in the predicted vector of proportion is assigned as the unique topic that describes the unseen document.

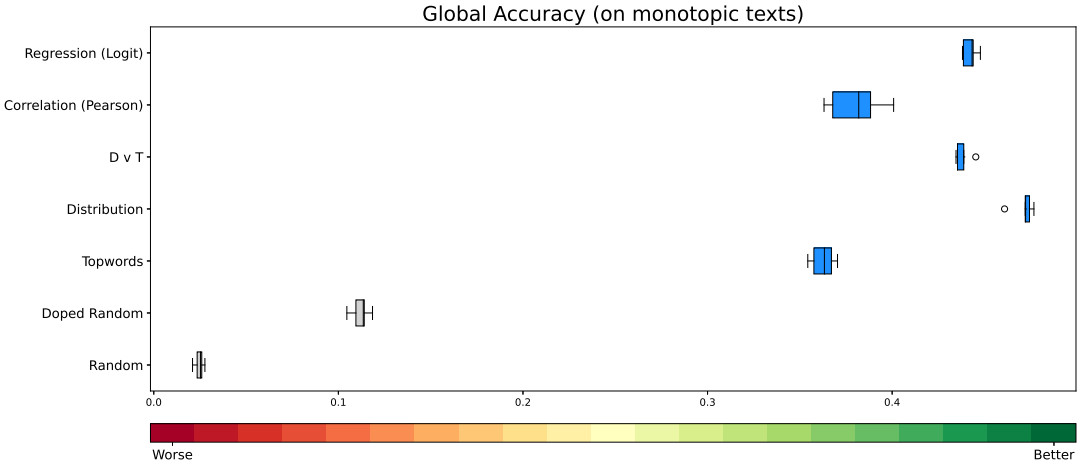

Figure 5.1: Global accuracy (computed on monotopic texts) boxplots

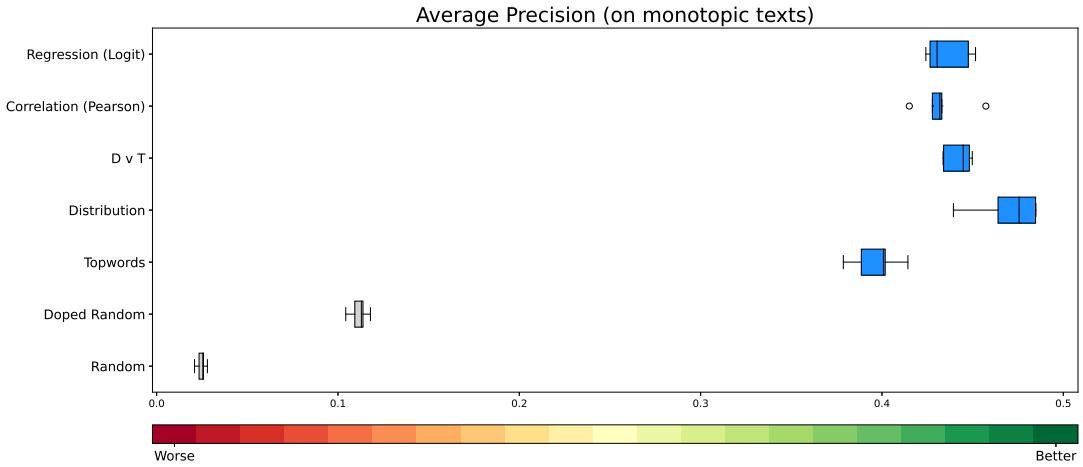

Figure 5.2: Average precision (computed on monotopic texts) boxplots

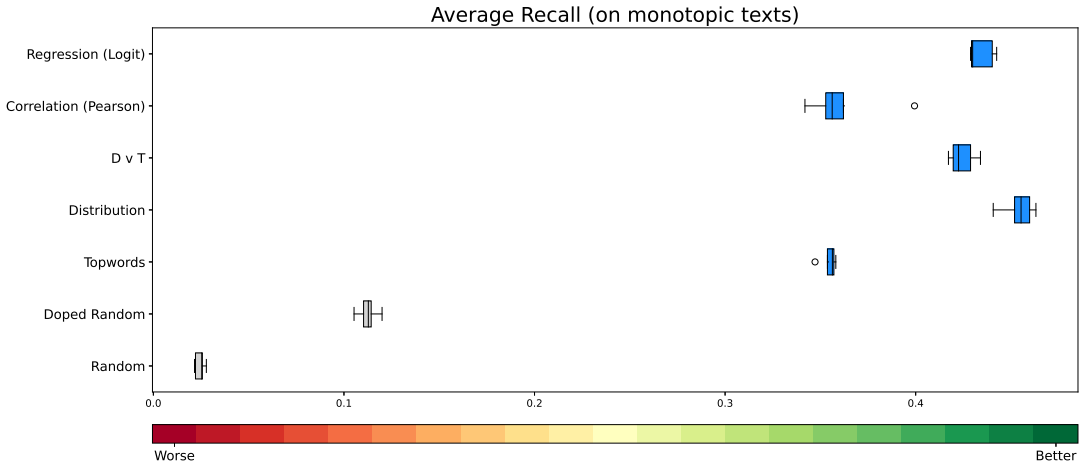

Figure 5.3: Average recall (computed on monotopic texts) boxplots

The Figures 5.1, 5.2, 5.3 summarize the results showing a huge difference, not only with the pure random model (i.e. the model don't make random choices) but even with the doped random model, in which one topic is passed directly to the results. So the proposed approach shows to be very powerful to identify topics.

To measure the gap the standard deviations ais used s a unit of measure.

Results are very strong. The distance from the pure Random model is more than one hundred of standard deviations. Furthermore, even the distance from doped Random is remarkable, always more than forty standard deviations.

	Global Accuracy		Average Precision		Average Recall	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Regression	0,44	0,004	0,44	0,0126	0,43	0,0066
Correlation	0,38	0,0153	0,43	0,0153	0,36	0,0219
D ⋁ T	0,44	0,0044	0,44	0,0077	0,42	0,0069
Distribution	0,47	0,0061	0,47	0,019	0,45	0,0087
Topwords	0,36	0,0066	0,4	0,0136	0,35	0,0044
Doped Random	0,112	0,0052	0,112	0,0051	0,113	0,0054
Random	0,025	0,0026	0,025	0,0027	0,025	0,0026

Table 5.1: Global Accuracy, Average Precision and Average Recall - mean and standard deviation of baptism methods and random models

	DSTD					
	Global Accuracy		Average Precision		Average Recall	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Regression	163,25	63,13	153,12	63,38	156,77	59,65
Correlation	139,03	51,29	151,99	62,79	129,35	46,36
D ⋁ T	161,51	62,28	155,37	64,56	153,06	57,86
Distribution	174,57	68,67	165,65	69,96	164,43	63,37
Topwords	132,06	47,8	138,52	55,7	126,28	44,8

Table 5.2: Global Accuracy, Average Precision and Average Recall - distance between the mean of baptism method and the mean of random (or doped random), in terms of random (or doped random) standard deviations

5.3.3 Results with the new metrics

Now the same analysis can be preformed with the metrics introduced in this work which allow to deal with documents with mix of topics too.

As we can see in Figures from 5.4 to 5.11, even with the new metrics the distance remains very wide.

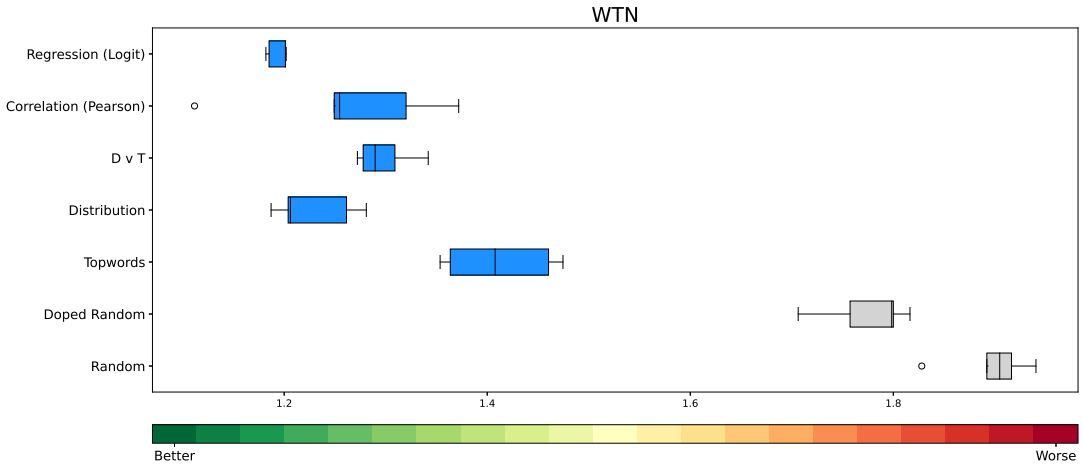

Figure 5.4: WTN boxplots

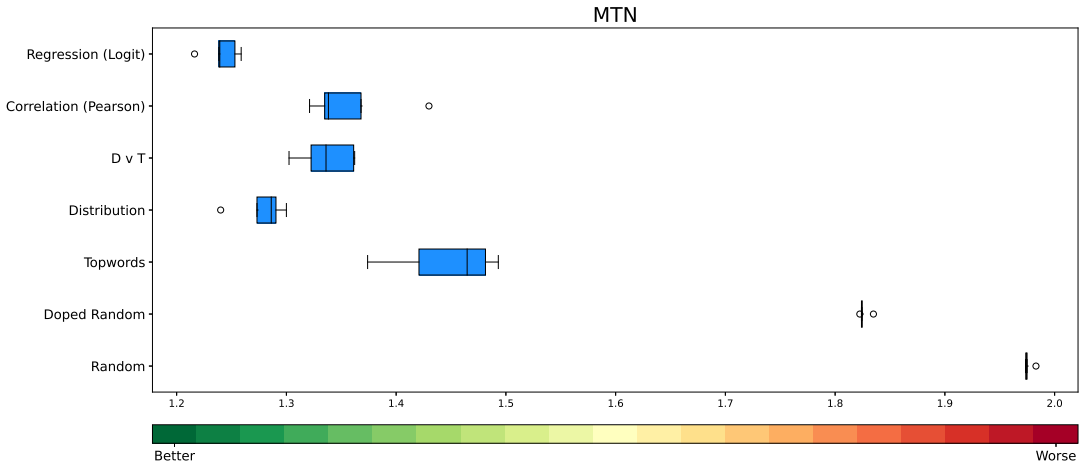

Figure 5.5: MTN boxplots

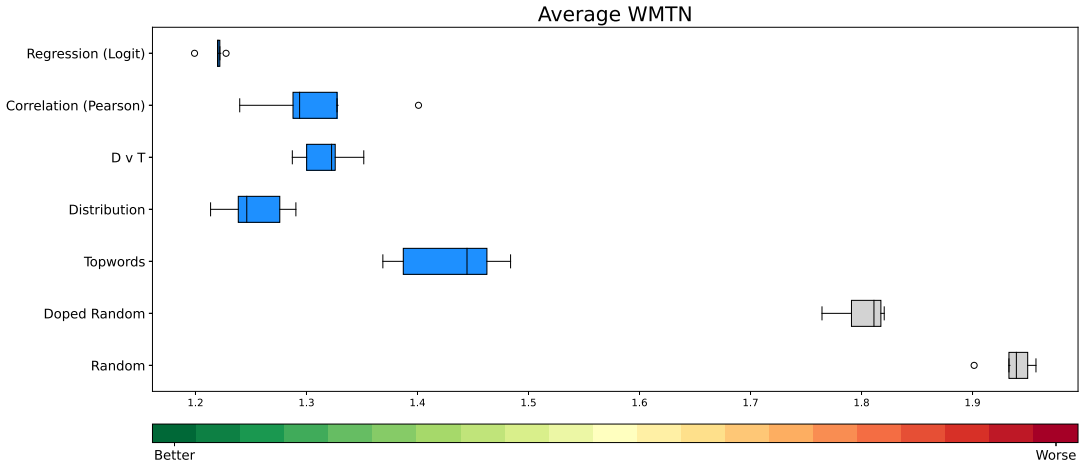

Figure 5.6: WMTN boxplots

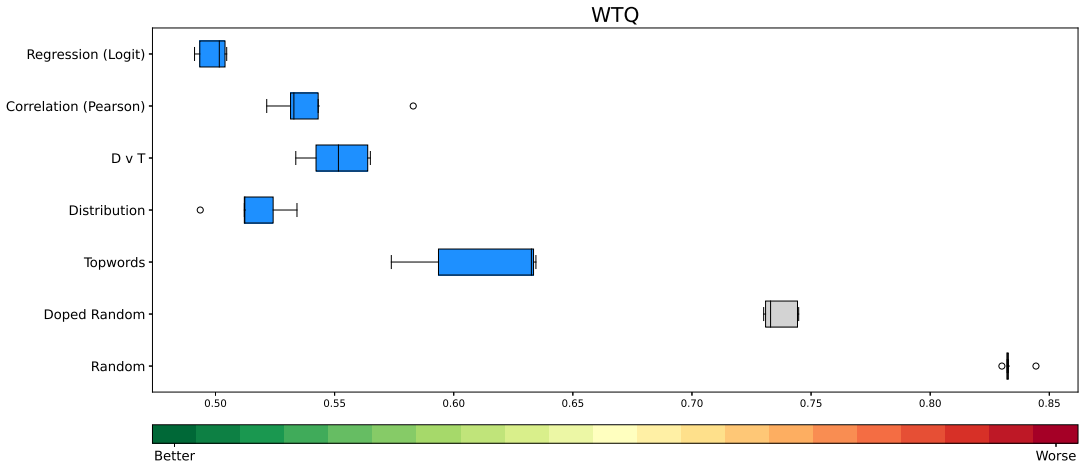

Figure 5.7: WTQ boxplots

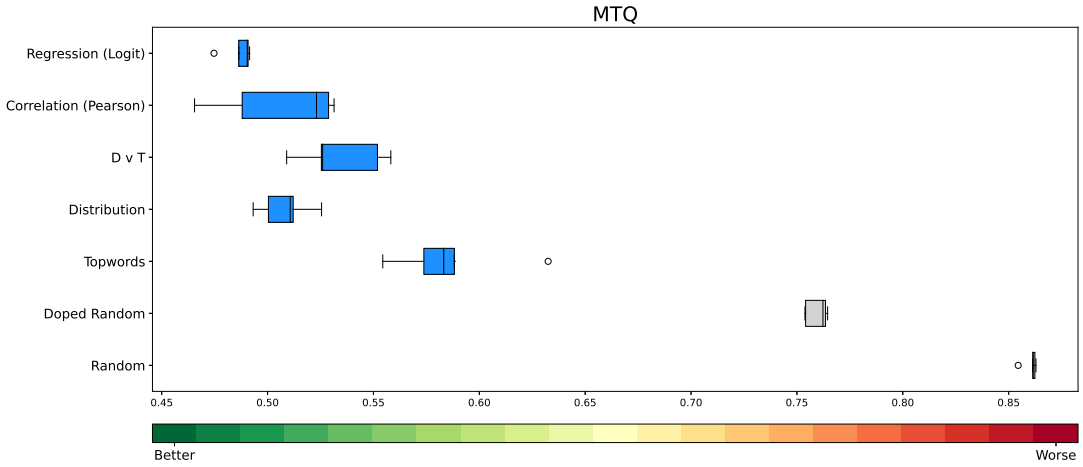

Figure 5.8: MTQ boxplots

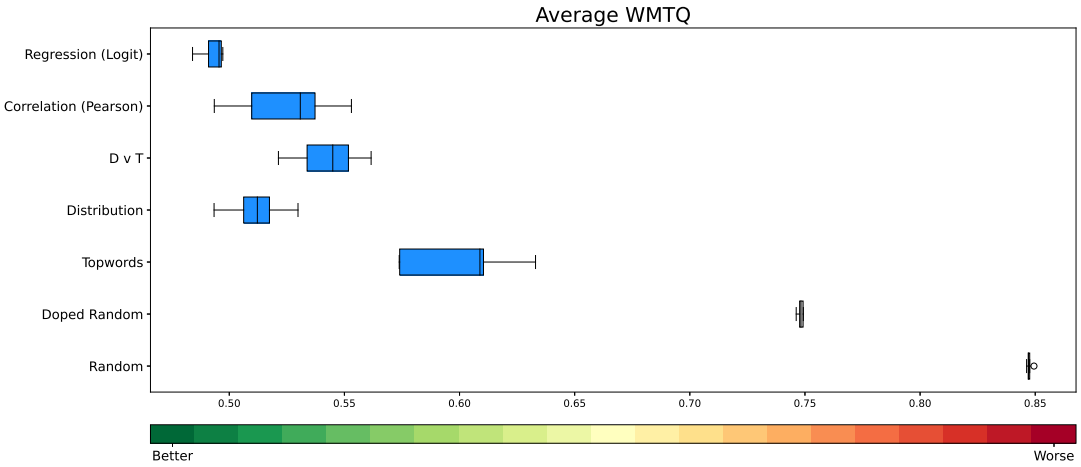

Figure 5.9: WMTQ boxplots

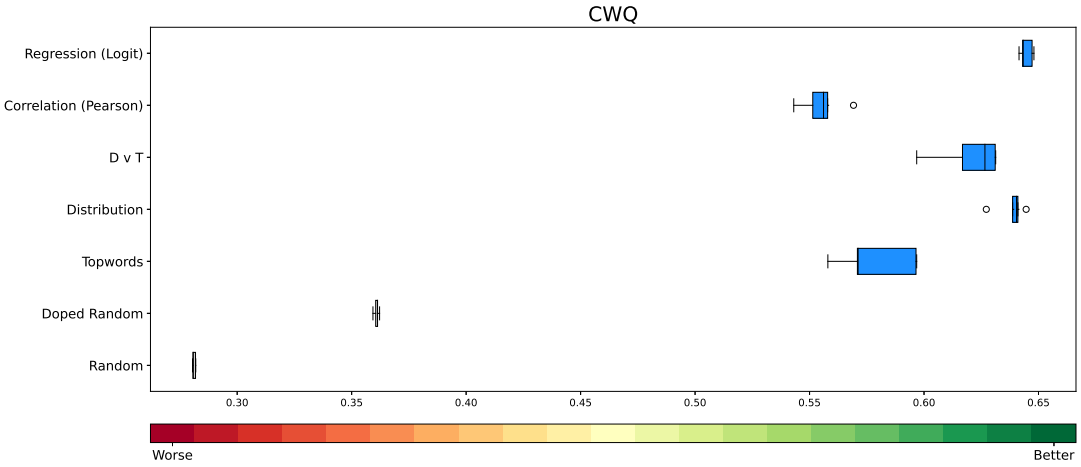

Figure 5.10: CWQ boxplots

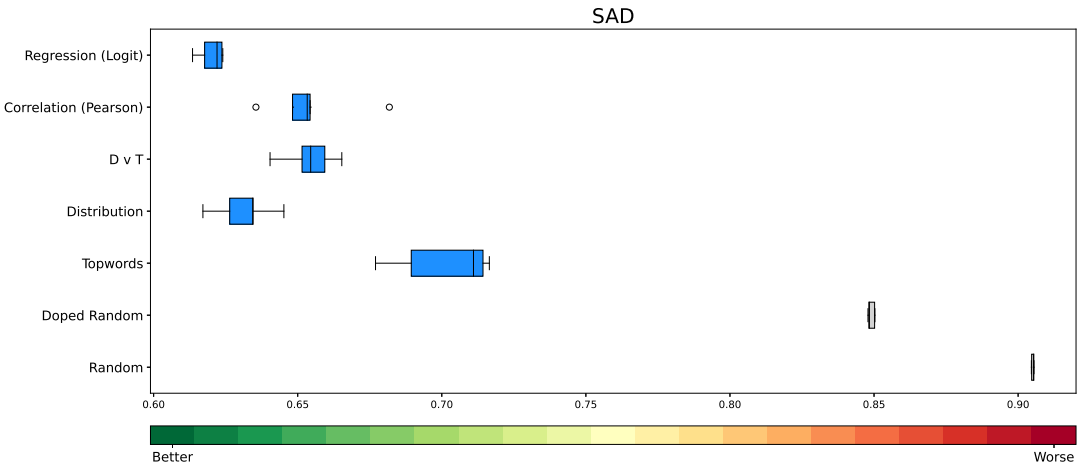

Figure 5.11: SAD boxplots

Even the quantification of the distance in standard deviations shows the efficiency of the models. We can notice looking at tables 5.4, 5.6, 5.8 that with the first metrics (WTN, MTN and WMTN) that just count the topics the difference is smaller because it doesn't take into account the score assigned.

	WTN		MTN		WMTN	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Regression	1,19	0,0099	1,24	0,0164	1,22	0,0108
Correlation	1,26	0,0978	1,36	0,0435	1,31	0,0597
D ⋁ T	1,3	0,0283	1,34	0,0257	1,32	0,0249
Distribution	1,23	0,0408	1,28	0,0233	1,25	0,0306
Topwords	1,41	0,0548	1,45	0,0491	1,43	0,0494
Doped Random	1,776	0,0445	1,826	0,005	1,801	0,0234
Random	1,896	0,0422	1,976	0,004	1,936	0,0215

Table 5.3: WTN, MTN and WMTN - mean and standard deviation of baptism methods and random models. Low values indicate better performances.

	DSTD					
	WTN		MTN		WMTN	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Regression	16,65	13,07	183,0	117,02	33,39	24,9
Correlation	15,06	11,56	153,8	93,56	29,11	20,96
D ⋁ T	14,19	10,74	159,16	97,87	28,76	20,64
Distribution	15,85	12,32	173,84	109,66	31,76	23,4
Topwords	11,49	8,1	131,79	75,8	23,55	15,8

Table 5.4: WTN, MTN and WMTN - distance between the mean of baptism method and the mean of random (or doped random), in terms of random (or doped random) standard deviations

	WTQ		MTQ		WMTQ	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Regression	0,5	0,0063	0,49	0,007	0,49	0,0055
Correlation	0,54	0,024	0,51	0,0292	0,52	0,0234
D ⋁ T	0,55	0,0136	0,53	0,0204	0,54	0,0156
Distribution	0,52	0,0152	0,51	0,0123	0,51	0,0135
Topwords	0,61	0,0282	0,59	0,0288	0,6	0,0257
Doped Random	0,737	0,0073	0,76	0,0053	0,748	0,0013
Random	0,834	0,0057	0,861	0,0034	0,847	0,0012

Table 5.5: WTQ, MTQ and WMTQ - mean and standard deviation of baptism methods and random models. Low values indicate better performances.

	DSTD					
	WTQ		MTQ		WMTQ	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Regression	58,86	32,47	108,62	51,94	293,65	201,2
Correlation	51,24	26,54	102,63	48,02	267,14	175,97
D $\bigvee$ T	49,68	25,33	94,86	42,93	252,39	161,93
Distribution	56,0	30,24	102,34	47,83	277,94	186,26
Topwords	38,75	16,8	79,66	32,9	204,92	116,7

Table 5.6: WTQ, MTQ and WMTQ - distance between the mean of baptism method and the mean of random (or doped random), in terms of random (or doped random) standard deviations

	CWQ		SAD	
Method	**Mean**	**Std**	**Mean**	**Std**
Regression	0,64	0,0028	0,62	0,0045
Correlation	0,56	0,0095	0,65	0,0169
D $\bigvee$ T	0,62	0,0145	0,65	0,0093
Distribution	0,64	0,0066	0,63	0,0105
Topwords	0,58	0,0172	0,7	0,0175
Doped Random	0,361	0,0011	0,849	0,0012
Random	0,281	0,0007	0,905	0,0004

Table 5.7: CWQ and SAD - mean and standard deviation of baptism methods and random models. In CWQ, better performances are indicated by high values, whereas, in SAD low values indicate better performances.

	DSTD			
	CWQ		SAD	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**
Regression	552,38	260,95	658,92	198,04
Correlation	417,0	179,01	579,28	168,23
D $\bigvee$ T	515,75	238,78	580,2	168,58
Distribution	542,9	255,21	632,7	188,23
Topwords	452,17	200,2	470,54	127,5

Table 5.8: CWQ and SAD - distance between the mean of baptism method and the mean of random (or doped random), in terms of random (or doped random) standard deviations

5.3.4 Performances of supervised Statistical Learning Models

To verify how much SLMs are better than our approach (that is designed allow to compare performances of topic models) they are evaluated in the same context. As a matter of fact it is suprising to see that the results (Tables from 5.9 to 5.16) are of the same order of magnitude of our approach both with classification metrics and with our new metrics. This indirectly further confirms the validity of the new metrics.

	Global Accuracy		Average Precision		Average Recall	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Linear Regression	0,52	0,0104	0,59	0,0077	0,51	0,01
Linear Regression (Logit)	0,53	0,0079	0,58	0,0076	0,52	0,0066
Ada Boost	0,4	0,0086	0,54	0,0079	0,39	0,0074
Ada Boost (Logit)	0,42	0,018	0,55	0,0169	0,41	0,0208
Gradient Boosting	0,56	0,0061	0,64	0,006	0,55	0,0057
Gradient Boosting (Logit)	0,56	0,0069	0,62	0,0088	0,55	0,0067
Random Forest	0,47	0,006	0,61	0,0067	0,46	0,0064
Random Forest (Logit)	0,46	0,0044	0,57	0,0033	0,45	0,005
KNN	0,55	0,0066	0,57	0,006	0,54	0,0056
KNN (Logit)	0,55	0,0075	0,57	0,0058	0,54	0,0063
Bagging Regressor	0,56	0,0094	0,58	0,0104	0,55	0,0101
Bagging Regressor (Logit)	0,56	0,0063	0,58	0,0061	0,55	0,0073
Doped Random	0,112	0,0052	0,112	0,0051	0,113	0,0054
Random	0,025	0,0026	0,025	0,0027	0,025	0,0026

Table 5.9: Global Accuracy, Average Precision and Average Recall - mean and standard deviation of regressors and random models

	DSTD					
	Global Accuracy		Average Precision		Average Recall	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Linear Regression	193,67	78,0	209,89	93,18	185,65	73,65
Linear Regression (Logit)	198,9	80,56	207,67	92,02	190,31	75,91
Ada Boost	148,11	55,73	191,11	83,32	140,82	51,92
Ada Boost (Logit)	153,4	58,31	194,34	85,02	146,09	54,48
Gradient Boosting	207,53	84,78	227,66	102,51	199,98	80,6
Gradient Boosting (Logit)	208,14	85,08	222,16	99,62	200,69	80,94
Random Forest	175,06	68,91	219,19	98,06	167,9	65,05
Random Forest (Logit)	169,84	66,35	203,32	89,74	162,86	62,6
KNN	204,92	83,5	203,8	89,99	198,74	80,0
KNN (Logit)	205,64	83,86	202,15	89,12	199,08	80,16
Bagging Regressor	208,58	85,3	208,3	92,35	201,76	81,46
Bagging Regressor (Logit)	208,18	85,1	207,58	91,97	200,98	81,08

Table 5.10: Global Accuracy, Average Precision and Average Recall - distance between the mean of regressors and the mean of random (or doped random), in terms of random (or doped random) standard deviations

	WTN		MTN		WMTN	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Linear Regression	0,81	0,0038	0,84	0,0048	0,82	0,0018
Linear Regression (Logit)	0,8	0,0048	0,87	0,0036	0,84	0,0039
Ada Boost	0,94	0,2009	1,25	0,0445	1,09	0,0876
Ada Boost (Logit)	1,13	0,0535	1,19	0,039	1,16	0,0433
Gradient Boosting	0,84	0,0042	0,81	0,003	0,83	0,0034
Gradient Boosting (Logit)	0,76	0,0031	0,85	0,004	0,81	0,0027
Random Forest	0,89	0,0061	0,96	0,0039	0,92	0,0042
Random Forest (Logit)	0,82	0,0948	0,98	0,022	0,9	0,0366
KNN	0,86	0,0064	0,85	0,004	0,86	0,0037
KNN (Logit)	0,88	0,0051	0,89	0,0042	0,88	0,0036
Bagging Regressor	0,78	0,0034	0,79	0,0041	0,78	0,0029
Bagging Regressor (Logit)	0,76	0,006	0,82	0,0047	0,79	0,0048
Doped Random	1,776	0,0445	1,826	0,005	1,801	0,0234
Random	1,896	0,0422	1,976	0,004	1,936	0,0215

Table 5.11: WTN, MTN and WMTN - mean and standard deviation of regressors and random models. Low values indicate better performances

	DSTD					
	WTN		MTN		WMTN	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Linear Regression	25,85	21,79	282,55	196,97	51,7	41,72
Linear Regression (Logit)	25,96	21,9	275,56	191,36	51,15	41,22
Ada Boost	22,66	18,77	181,82	116,07	39,18	30,22
Ada Boost (Logit)	18,23	14,57	196,16	127,59	36,17	27,45
Gradient Boosting	25,05	21,03	289,52	202,58	51,57	41,6
Gradient Boosting (Logit)	26,88	22,77	280,36	195,22	52,5	42,46
Random Forest	23,83	19,88	254,19	174,2	47,07	37,47
Random Forest (Logit)	25,43	21,39	247,81	169,07	48,04	38,36
KNN	24,51	20,52	280,16	195,05	50,16	40,31
KNN (Logit)	24,18	20,21	271,01	187,7	48,99	39,23
Bagging Regressor	26,49	22,4	295,92	207,71	53,58	43,44
Bagging Regressor (Logit)	27,0	22,88	287,74	201,14	53,31	43,2

Table 5.12: WTN, MTN and WMTN - distance between the mean of regressors and the mean of random (or doped random), in terms of random (or doped random) standard deviations

	WTQ		MTQ		WMTQ	
Method	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Linear Regression	0,32	0,0015	0,31	0,0017	0,32	0,0013
Linear Regression (Logit)	0,3	0,0052	0,28	0,0136	0,29	0,0044
Ada Boost	0,44	0,0759	0,5	0,0305	0,47	0,0267
Ada Boost (Logit)	0,47	0,0198	0,47	0,0182	0,47	0,0184
Gradient Boosting	0,31	0,0082	0,31	0,0084	0,31	0,0007
Gradient Boosting (Logit)	0,29	0,0011	0,27	0,0014	0,28	0,0011
Random Forest	0,38	0,0012	0,36	0,0021	0,37	0,0016
Random Forest (Logit)	0,35	0,0065	0,35	0,0197	0,35	0,0068
KNN	0,35	0,0013	0,33	0,0075	0,34	0,0033
KNN (Logit)	0,34	0,0046	0,31	0,0179	0,32	0,0068
Bagging Regressor	0,3	0,0053	0,29	0,0048	0,3	0,0008
Bagging Regressor (Logit)	0,28	0,0063	0,27	0,0198	0,28	0,007
Doped Random	0,737	0,0073	0,76	0,0053	0,748	0,0013
Random	0,834	0,0057	0,861	0,0034	0,847	0,0012

Table 5.13: WTQ, MTQ and WMTQ - mean and standard deviation of regressors and random models. Low values indicate better performances

	DSTD					
	WTQ		MTQ		WMTQ	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**	**Random**	**Doped Random**
Linear Regression	89,42	56,26	159,35	85,18	438,02	338,63
Linear Regression (Logit)	92,91	58,98	169,38	91,75	460,56	360,08
Ada Boost	69,13	40,46	103,53	48,61	310,61	217,36
Ada Boost (Logit)	63,1	35,77	114,03	55,49	311,34	218,05
Gradient Boosting	92,13	58,37	160,58	85,98	446,16	346,37
Gradient Boosting (Logit)	95,18	60,75	172,74	93,95	470,69	369,73
Random Forest	79,41	48,47	146,3	76,63	395,81	298,45
Random Forest (Logit)	84,63	52,54	149,78	78,91	413,09	314,9
KNN	84,65	52,55	155,18	82,45	420,83	322,27
KNN (Logit)	86,97	54,36	161,16	86,36	434,83	335,59
Bagging Regressor	92,92	58,99	165,46	89,18	454,98	354,77
Bagging Regressor (Logit)	96,64	61,89	171,4	93,07	472,23	371,19

Table 5.14: WTQ, MTQ and WMTQ - distance between the mean of regressors and the mean of random (or doped random), in terms of random (or doped random) standard deviations

	CWQ		SAD	
Method	**Mean**	**Std**	**Mean**	**Std**
Linear Regression	0,69	0,001	0,45	0,0011
Linear Regression (Logit)	0,7	0,0009	0,47	0,0012
Ada Boost	0,58	0,0027	0,66	0,01
Ada Boost (Logit)	0,67	0,0044	0,58	0,0124
Gradient Boosting	0,72	0,0007	0,43	0,0012
Gradient Boosting (Logit)	0,7	0,0011	0,45	0,0009
Random Forest	0,68	0,0007	0,48	0,0011
Random Forest (Logit)	0,68	0,0016	0,5	0,0017
KNN	0,67	0,0014	0,44	0,0014
KNN (Logit)	0,71	0,0014	0,47	0,0016
Bagging Regressor	0,71	0,0014	0,41	0,0015
Bagging Regressor (Logit)	0,73	0,0016	0,43	0,0023
Doped Random	0,361	0,0011	0,849	0,0012
Random	0,281	0,0007	0,905	0,0004

Table 5.15: CWQ and SAD - mean and standard deviation of regressors and random models. In CWQ, better performances are indicated by high values, whereas, in SAD low values indicate better performances.

	DSTD			
	CWQ		SAD	
Method	**Random**	**Doped Random**	**Random**	**Doped Random**
Linear Regression	617,06	300,09	1048,07	343,68
Linear Regression (Logit)	630,92	308,48	1014,03	330,94
Ada Boost	453,6	201,16	559,34	160,77
Ada Boost (Logit)	583,56	279,82	762,2	236,69
Gradient Boosting	659,73	325,92	1093,53	360,7
Gradient Boosting (Logit)	640,46	314,26	1041,31	341,16
Random Forest	613,07	297,68	973,74	315,87
Random Forest (Logit)	609,18	295,33	934,52	301,19
KNN	588,88	283,04	1066,29	350,5
KNN (Logit)	658,67	325,28	996,54	324,4
Bagging Regressor	649,4	319,67	1141,49	378,65
Bagging Regressor (Logit)	678,51	337,29	1094,22	360,96

Table 5.16: CWQ and SAD - distance between the mean of regressors and the mean of random (or doped random), in terms of random (or doped random) standard deviations

5.3.5 Computational load

One significant advantage of LDA is the minimal computational load required for training and predicting mixed topics for unseen documents. Figure 5.12 illustrates the computational load, measured in minutes, for our methods as well as for SLMs for each of the five run of the cross validation.

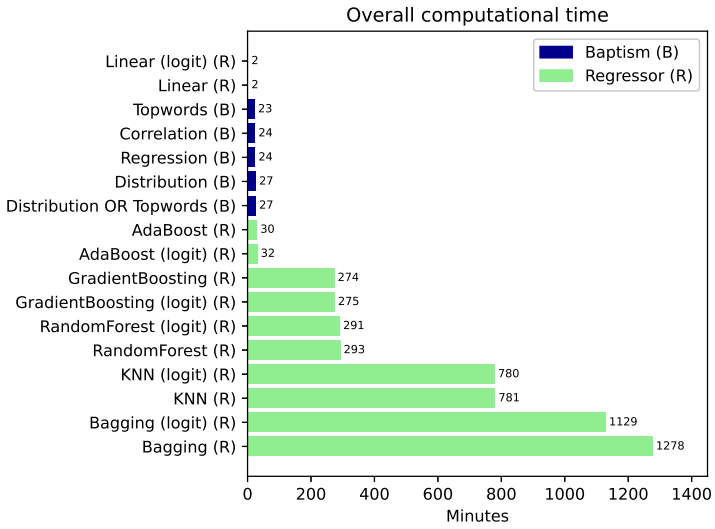

Figure 5.12: Graphic representation of computational time efforts.

5.4 Experiment 2: Comparison of four topic models

Having the new rich corpus of documents with known mix of topics and validated the new methods to identify real topic using topic models and verified the reliability of new metrics everything is ready to use the new methodological framework to compare topic models in their ability to identify real topic.

5.4.1 Models

In this experiment, the four selected latent topic models (LDA, CTM, HDP, PAM) with all the baptism methods are compared. The overarching objective is to evaluate these topic models within the proposed comparative framework.

All the considerations and hypotheses made in the first experiment remain applicable here as well. The only difference is that, due to the heavy computational requests of CTM and PAM,the iterations of models are reduced from the 1.000 of the first experiment to 100.

Let's remember that the known number H of real topic is 39. Again for LDA and CTM k (the number of latent topic to find) is set equal to 156 (=39x4). For HDP the library asks a value of topics from which the algorithm starts to evaluate the final number. In the experiment this value is set to 39. PAM instead asks for the number of super topic and the number of subtopic, for coherence the choice has been 39 for the first and 156 for the second.

5.4.2 Results with classification metrics

As we can see in Figures 5.13,5.14 and 5.15 and Table 5.17, 5.18 and 5.19 using classification metrics LDA and HDP seems to be clearly better than the others.

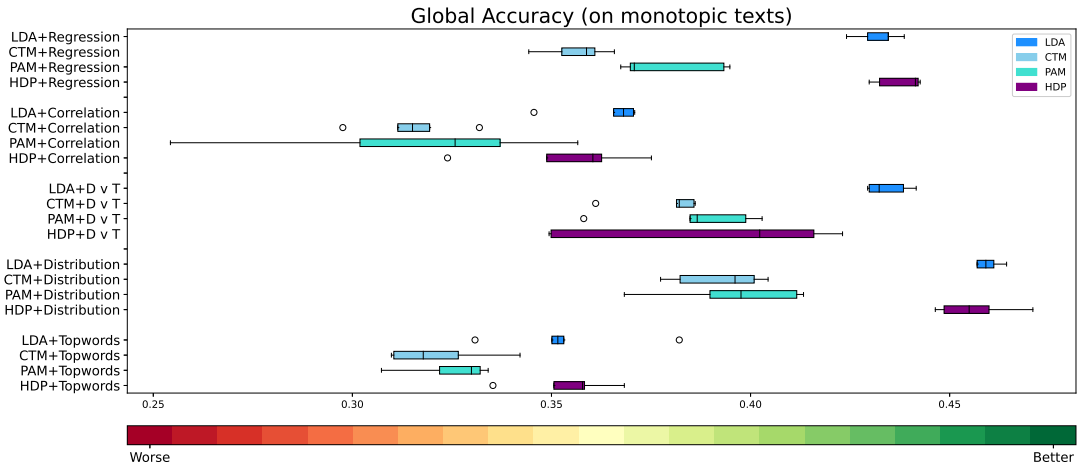

Figure 5.13: Global accuracy (computed on monotopic texts) boxplots

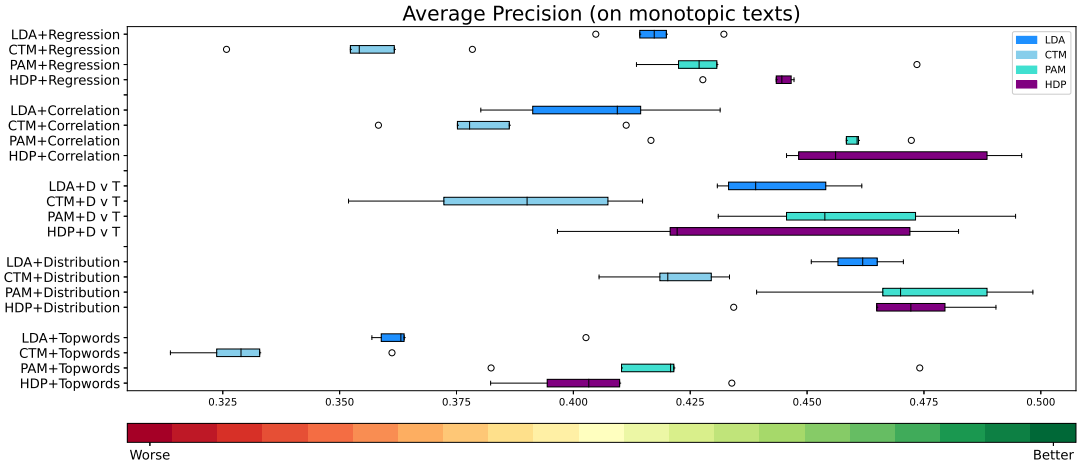

Figure 5.14: Average precision (computed on monotopic texts) boxplots

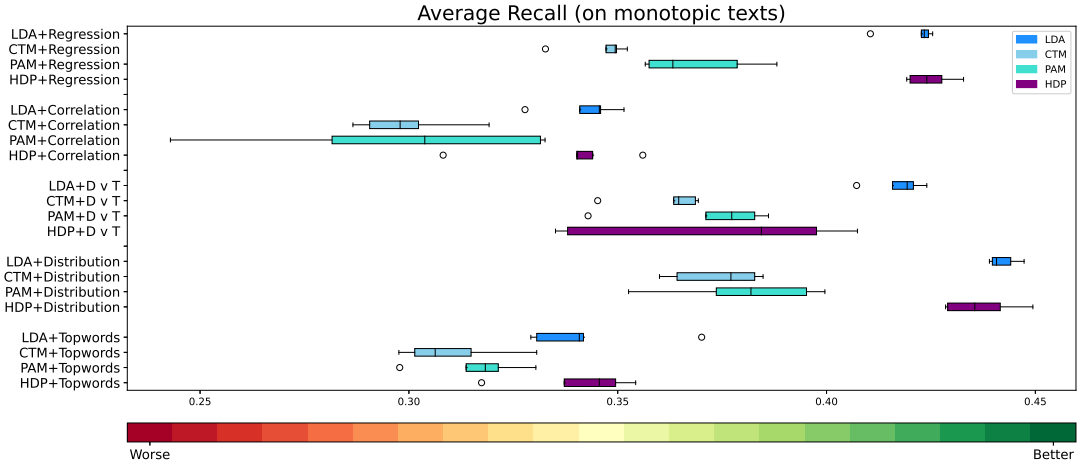

Figure 5.15: Average recall (computed on monotopic texts) boxplots

	Global Accuracy							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,43	0,36	0,44	0,38	0,0056	0,0083	0,0061	0,0136
Correlation	0,36	0,32	0,35	0,32	0,0106	0,0125	0,0193	0,0393
D ⋁ T	0,43	0,38	0,39	0,39	0,0054	0,0104	0,0359	0,0175
Distribution	0,46	0,39	0,46	0,4	0,0031	0,0118	0,0098	0,0184
Topwords	0,35	0,32	0,35	0,33	0,0184	0,0135	0,0122	0,011

Table 5.17: Global accuracy - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM.

	Average Precision							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,42	0,35	0,44	0,43	0,0099	0,0191	0,0081	0,0233
Correlation	0,41	0,38	0,47	0,45	0,02	0,0194	0,0236	0,0215
D ⋁ T	0,44	0,39	0,44	0,46	0,0135	0,0257	0,0367	0,0248
Distribution	0,46	0,42	0,47	0,47	0,0076	0,0109	0,0212	0,0228
Topwords	0,37	0,33	0,4	0,42	0,019	0,0178	0,0193	0,0332

Table 5.18: Average Precision - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM.

	Average Recall							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,42	0,35	0,42	0,37	0,0061	0,0078	0,0056	0,0139
Correlation	0,34	0,3	0,34	0,3	0,009	0,0127	0,0177	0,0376
D ⋁ T	0,42	0,36	0,37	0,37	0,0064	0,0098	0,0338	0,0173
Distribution	0,44	0,37	0,44	0,38	0,0035	0,0111	0,0088	0,0188
Topwords	0,34	0,31	0,34	0,32	0,0165	0,0131	0,0145	0,012

Table 5.19: Average Recall - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM.

5.4.3 Results with new metrics

Using this metrics results are more uncertain, even if skipping for a while the first 6 metrics that are less sophisticated (as they counts the wrong topics) and that depends on the choice of a threshold and using CWQ and SAD LDA seems to be the best both as value and as variability.

So LDA coupled with the regression based baptism method is the winner.

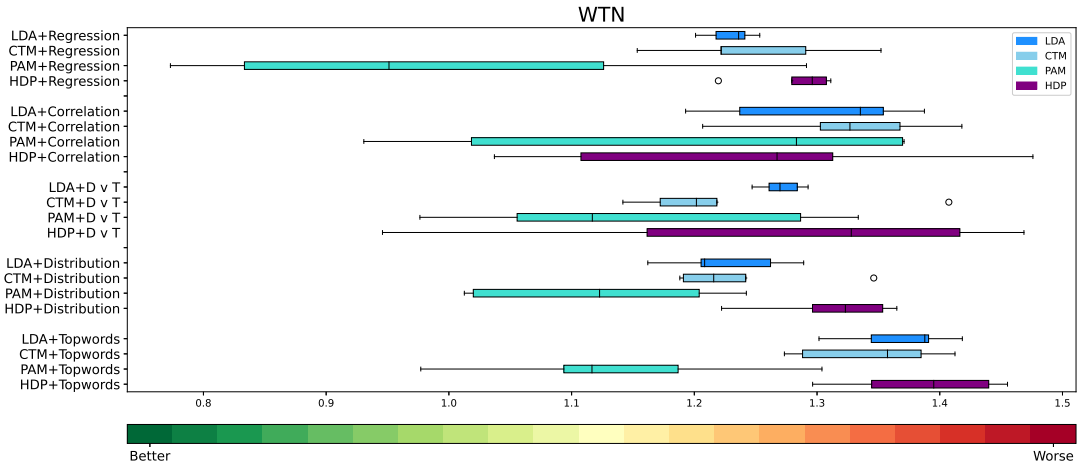

Figure 5.16: WTN boxplots

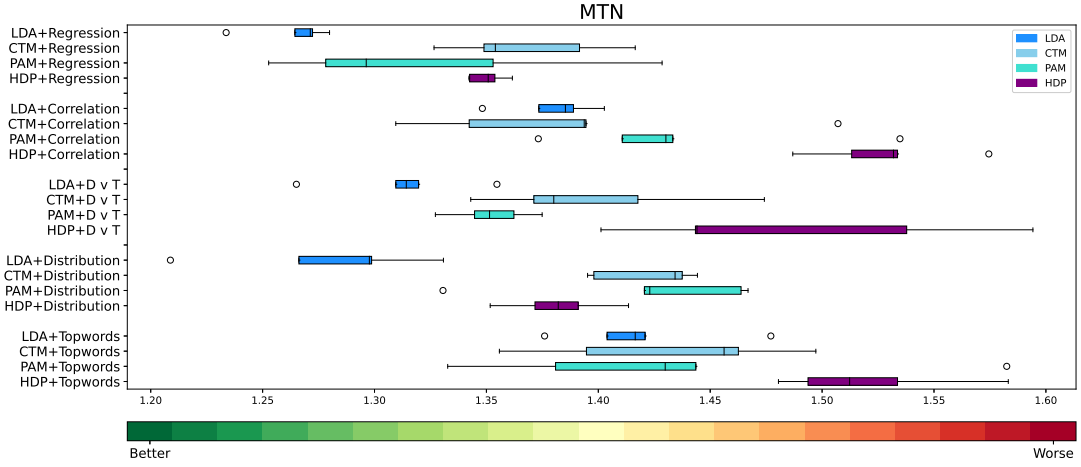

Figure 5.17: MTN boxplots

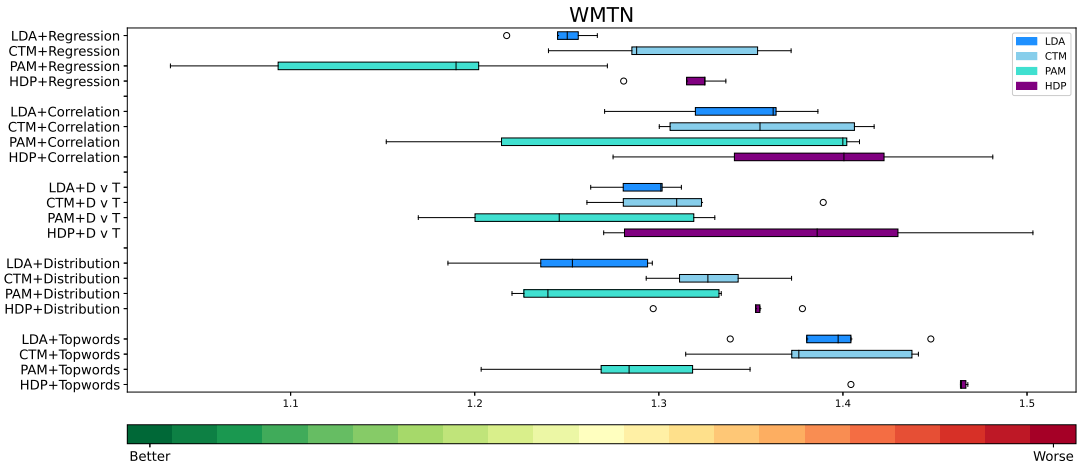

Figure 5.18: WMTN boxplots

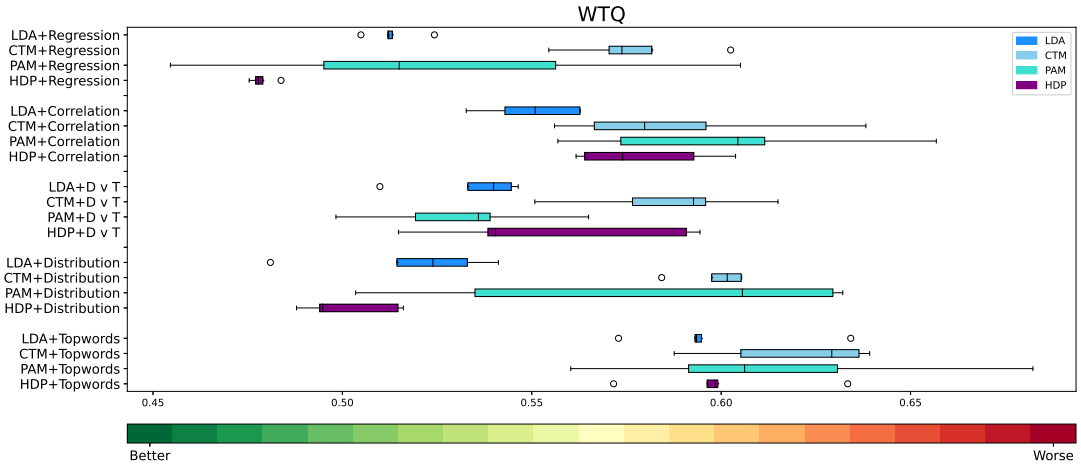

Figure 5.19: WTQ boxplots

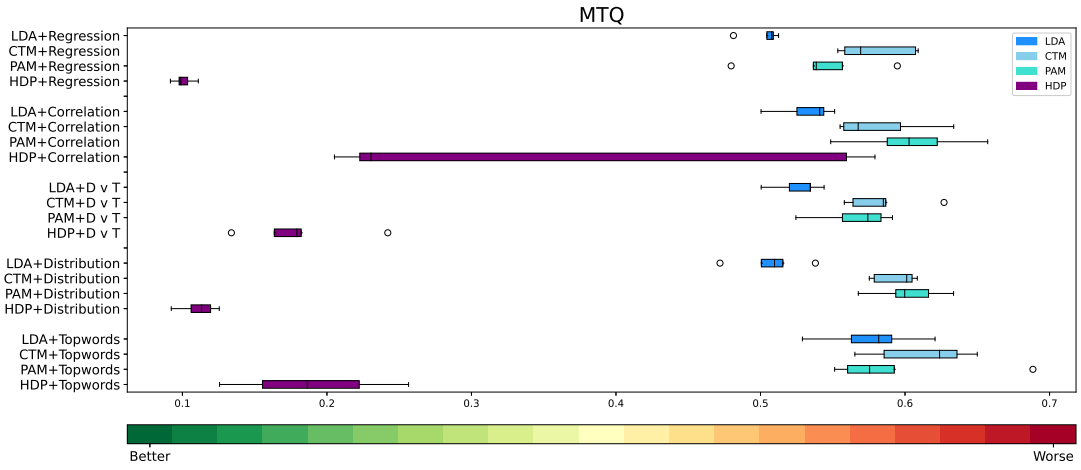

Figure 5.20: MTQ boxplots

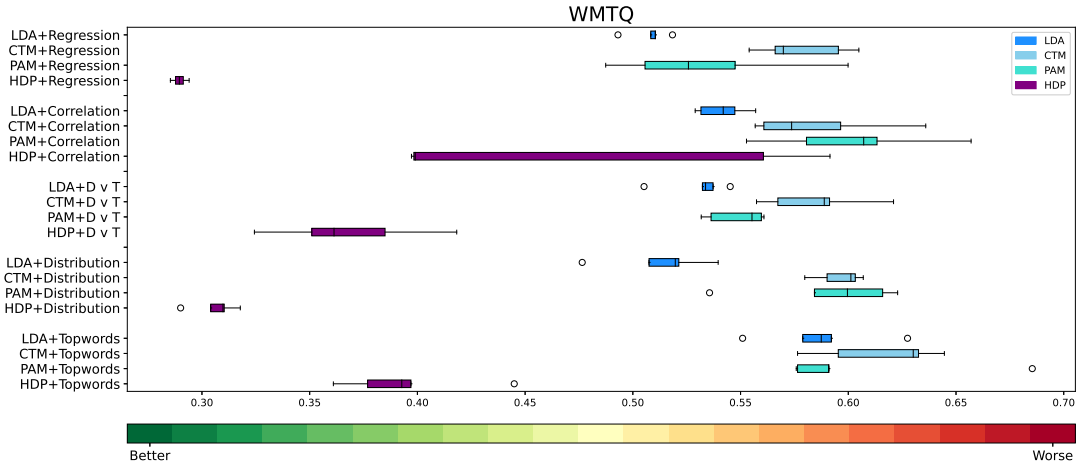

Figure 5.21: WMTQ boxplots

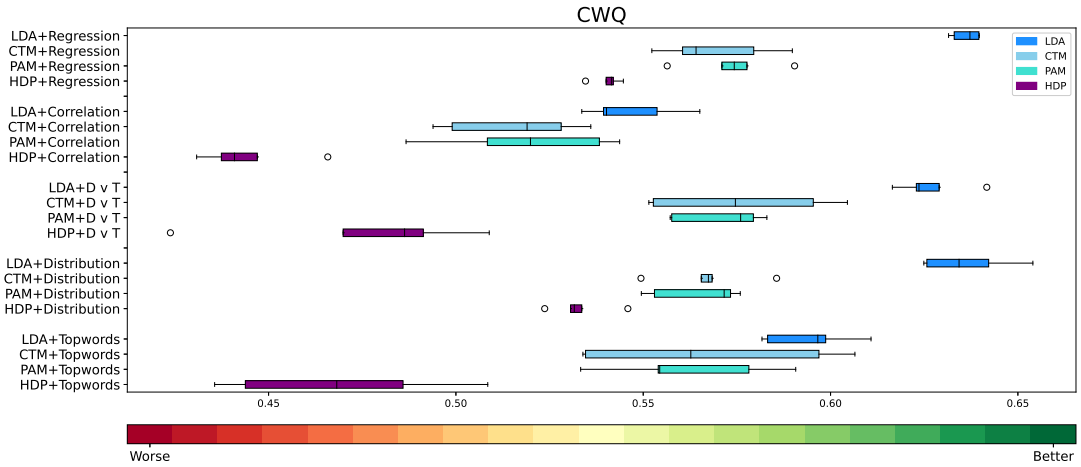

Figure 5.22: CWQ boxplots

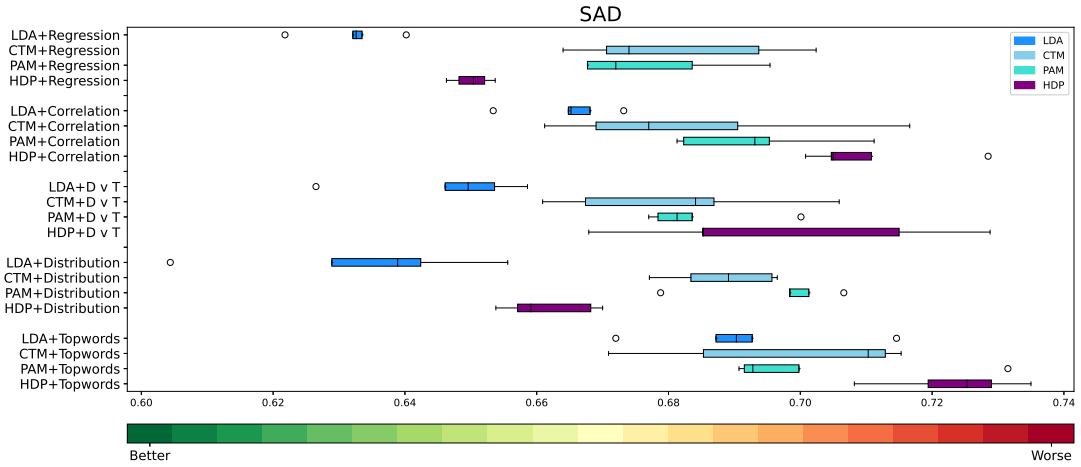

Figure 5.23: SAD boxplots

	WTN							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	1,23	1,25	1,28	1,0	0,0206	0,0758	0,0374	0,2135
Correlation	1,3	1,32	1,24	1,19	0,0825	0,079	0,1735	0,2063
D ⋁ T	1,27	1,23	1,26	1,15	0,0181	0,1043	0,2126	0,152
Distribution	1,23	1,24	1,31	1,12	0,0502	0,065	0,0569	0,1044
Topwords	1,37	1,34	1,39	1,14	0,0459	0,0605	0,0662	0,1206

Table 5.20: WTN - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In WTN, better performances are indicated by low values

	MTN							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	1,26	1,37	1,35	1,32	0,018	0,036	0,0082	0,0702
Correlation	1,38	1,39	1,53	1,44	0,0205	0,075	0,0321	0,06
D ⋁ T	1,31	1,4	1,48	1,35	0,032	0,0506	0,0793	0,018
Distribution	1,28	1,42	1,38	1,42	0,0461	0,0234	0,0229	0,055
Topwords	1,42	1,43	1,52	1,43	0,037	0,0569	0,0403	0,0939

Table 5.21: MTN - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In MTN, better performances are indicated by low values

	WMTN							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	1,25	1,31	1,32	1,16	0,0185	0,0541	0,0213	0,094
Correlation	1,34	1,36	1,38	1,32	0,0459	0,0544	0,0789	0,1228
D ⋁ T	1,29	1,31	1,37	1,25	0,0197	0,0492	0,0992	0,071
Distribution	1,25	1,33	1,35	1,27	0,0459	0,0303	0,03	0,0576
Topwords	1,39	1,39	1,45	1,28	0,0395	0,0525	0,0275	0,0552

Table 5.22: WMTN - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In WMTN, better performances are indicated by low values

	WTQ							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,51	0,58	0,48	0,53	0,007	0,0175	0,0032	0,0578
Correlation	0,55	0,59	0,58	0,6	0,013	0,0322	0,0184	0,0385
D ⋁ T	0,53	0,59	0,56	0,53	0,0148	0,024	0,0351	0,0247
Distribution	0,52	0,6	0,5	0,58	0,0233	0,0087	0,013	0,0585
Topwords	0,6	0,62	0,6	0,61	0,0223	0,0223	0,022	0,0458

Table 5.23: WTQ - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In WTQ, better performances are indicated by low values

	MTQ							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,5	0,58	0,1	0,54	0,0124	0,0268	0,0072	0,0416
Correlation	0,53	0,58	0,36	0,6	0,0202	0,0332	0,192	0,0403
D ⋁ T	0,53	0,58	0,18	0,57	0,017	0,0271	0,0395	0,0266
Distribution	0,51	0,59	0,11	0,6	0,024	0,0155	0,0129	0,0248
Topwords	0,58	0,61	0,19	0,59	0,034	0,0355	0,052	0,0553

Table 5.24: MTQ - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In MTQ, better performances are indicated by low values

	WMTQ							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,51	0,58	0,29	0,53	0,0092	0,0213	0,0033	0,0435
Correlation	0,54	0,58	0,47	0,6	0,0115	0,0325	0,098	0,0389
D ⋁ T	0,53	0,59	0,37	0,55	0,0151	0,0246	0,0355	0,0137
Distribution	0,51	0,6	0,31	0,59	0,0234	0,0111	0,0103	0,0347
Topwords	0,59	0,62	0,39	0,6	0,0275	0,0287	0,0316	0,0461

Table 5.25: WMTQ - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In WMTQ, better performances are indicated by low values

	CWQ							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,64	0,57	0,54	0,57	0,0038	0,0151	0,0037	0,0123
Correlation	0,55	0,52	0,44	0,52	0,0128	0,0182	0,0133	0,0231
D ⋁ T	0,63	0,58	0,48	0,57	0,0095	0,0242	0,0323	0,0123
Distribution	0,64	0,57	0,53	0,56	0,0122	0,0128	0,0081	0,0124
Topwords	0,59	0,57	0,47	0,56	0,012	0,034	0,03	0,0225

Table 5.26: CWQ - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In CWQ, better performances are indicated by high values

	SAD							
	Mean				Std			
Method	**LDA**	**CTM**	**HDP**	**PAM**	**LDA**	**CTM**	**HDP**	**PAM**
Regression	0,63	0,68	0,65	0,68	0,0066	0,0163	0,003	0,012
Correlation	0,66	0,68	0,71	0,69	0,0073	0,0218	0,011	0,0121
D ⋁ T	0,65	0,68	0,7	0,68	0,0123	0,0177	0,0248	0,0093
Distribution	0,63	0,69	0,66	0,7	0,0191	0,0082	0,0071	0,0106
Topwords	0,69	0,7	0,72	0,7	0,0153	0,0198	0,0102	0,0173

Table 5.27: SAD - mean and standard deviation of each baptism method made on LDA, CTM, HDP and PAM. In SAD, better performances are indicated by low values

5.4.4 Computational Load

As explained in chapter 2.2.1 CTM and PAM models needs more computational load than LDA. And the experiment confirms this. Taking into consideration just the training time (with 100 iterations on the 200'000 documents subset), as the baptism is the same for all models, here are there is the list in order of speed:

- LDA: about 2 minutes
- HDP: about 4 minutes
- PAM: about 3.5 hours
- CTM: about 14.5 hours

5.5 Experiments 3: Hyperparameters tuning

With Experiments 2 the aims of this research can be considered successfully achieved as a new methodological framework has been well tested and it is ready to be used to compare performances of Topic Models in their ability to identify real topics.

Nevertheless at this point it worth to go further and try to test the new methodology on all the bigdata dataset of 6 millions of documents using the best model according to the chapter 5.4: LDA+regression.

Before doing this (it will be done in the next chapter (5.6) it could be a good idea to tune the hyperparameters of that model. In particular a test on the combination of these three hyperparameters is performed, always using cross validation approach described in 5.2 and with 100 iteration:

1. **Term Weighting**. This define the approach used to weight the words inside a document according to [30]. Possible values are:
    - ONE: Consider every term equal (it is the default value)
    - PMI: Use Pointwise Mutual Information term weighting, introduced in [8]. It compares the probability of two events occurring together to what this probability would be if the events were independent.

- IDF: Use Inverse Document Frequency term weighting (3.4.2). Thus, a term occurring at almost every document has very low weighting and a term occurring at a few document has high weighting.

2. **Alpha**. It is described in chapter 2.2.1. The default value is 0.1. In several papers (as [11]) it is suggested to put this equal to 50/k (where k is the number of latent topic, as everywhere in this thesis). Therefore three values are tested:

   - alpha=0.1 (the default value)
   - alpha=0.32 ($\sim$50/H, that is more than three time the default value 0.1)
   - alpha=0.02 (that is 1/5 of the default value 0.1)

3. **Eta**. It is described in chapter 2.2.1 calling it Beta according to in the original LDA paper. The default value is 0.1 so one half and two times this values are taken as other value to compare:

   - eta=0.01 (the default value)
   - eta=0.02 (two times the default value)
   - eta=0.005 (half the default value)

5.5.1 Used metrics

Results are shown with the three classification metrics (accuracy, precision and recall) and the two new metrics that doesn't need a threshold to be defined (CWQ and SAD).

5.5.2 Results

In figures 5.28, 5.27, 5.24, 5.25 and 5.26 there are results of different metrics giving the mean of the cross validation values and the standard deviation. For the mean better values are in green and worse in orange passing through yellow, the best value is put in bold to find it easily. For the standard deviation darker pink means worse values.

	accuracy	Mean			Standard deviation		
tw	alfa / eta	0.005	0.01	0.02	0.005	0.01	0.02
ONE	0.02	0.437	0.428	0.437	0.005	0.009	0.004
ONE	0.1	0.430	0.432	0.428	0.008	0.006	0.006
ONE	0.32	**0.437**	**0.437**	0.436	0.014	0.012	0.013
PMI	0.02	0.387	0.388	0.382	0.006	0.006	0.011
PMI	0.1	0.410	0.412	0.412	0.011	0.008	0.006
PMI	0.32	0.412	0.420	0.418	0.008	0.005	0.012
IDF	0.02	0.388	0.387	0.387	0.009	0.011	0.007
IDF	0.1	0.405	0.409	0.407	0.007	0.006	0.010
IDF	0.32	0.427	0.431	0.428	0.007	0.006	0.007

Figure 5.24: Matrix of performances of different values of the hyperparameters using accuracy.

	precision	Mean			Standard deviation		
tw	alfa / eta	0.005	0.01	0.02	0.005	0.01	0.02
ONE	0.02	0.424	0.408	0.422	0.012	0.018	0.008
ONE	0.1	0.414	0.418	0.407	0.011	0.010	0.011
ONE	0.32	0.422	0.430	**0.431**	0.010	0.024	0.015
PMI	0.02	0.366	0.374	0.366	0.012	0.007	0.012
PMI	0.1	0.395	0.393	0.384	0.017	0.006	0.008
PMI	0.32	0.390	0.402	0.410	0.016	0.008	0.017
IDF	0.02	0.372	0.362	0.357	0.016	0.026	0.016
IDF	0.1	0.381	0.399	0.377	0.005	0.014	0.011
IDF	0.32	0.413	0.420	0.412	0.013	0.004	0.015

Figure 5.25: Matrix of performances of different values of the hyperparameters using precision.

	recall	Mean			Standard deviation		
tw	alfa / eta	0.005	0.01	0.02	0.005	0.01	0.02
ONE	0.02	0.426	0.414	0.425	0.008	0.007	0.004
ONE	0.1	0.421	0.421	0.417	0.005	0.006	0.006
ONE	0.32	**0.428**	0.426	0.427	0.011	0.011	0.015
PMI	0.02	0.379	0.377	0.376	0.003	0.006	0.010
PMI	0.1	0.401	0.404	0.403	0.011	0.008	0.004
PMI	0.32	0.405	0.413	0.413	0.007	0.009	0.013
IDF	0.02	0.377	0.376	0.377	0.010	0.010	0.007
IDF	0.1	0.396	0.401	0.400	0.009	0.010	0.011
IDF	0.32	0.420	0.423	0.421	0.009	0.006	0.010

Figure 5.26: Matrix of performances of different values of the hyperparameters using recall.

	CWQ	Mean			Standard deviation		
tw	alfa / eta	0.005	0.01	0.02	0.005	0.01	0.02
ONE	0.02	0.625	0.622	0.630	0.004	0.007	0.001
ONE	0.1	0.634	0.636	0.634	0.005	0.004	0.007
ONE	0.32	**0.640**	0.638	0.639	0.009	0.006	0.008
PMI	0.02	0.474	0.481	0.471	0.005	0.007	0.007
PMI	0.1	0.530	0.531	0.526	0.005	0.006	0.010
PMI	0.32	0.566	0.567	0.551	0.005	0.011	0.012
IDF	0.02	0.452	0.461	0.468	0.014	0.011	0.010
IDF	0.1	0.498	0.489	0.500	0.009	0.015	0.010
IDF	0.32	0.538	0.535	0.535	0.008	0.009	0.004

Figure 5.27: Matrix of performances of different values of the hyperparameters using CWQ metric.

	SAD	Mean			Standard deviation		
tw	alfa / eta	0.005	0.01	0.02	0.005	0.01	0.02
ONE	0.02	0.631	0.634	0.633	0.005	0.006	0.002
ONE	0.1	0.629	0.632	0.634	0.010	0.007	0.009
ONE	0.32	**0.627**	0.628	**0.627**	0.004	0.011	0.007
PMI	0.02	0.679	0.662	0.675	0.010	0.006	0.004
PMI	0.1	0.654	0.661	0.651	0.007	0.005	0.004
PMI	0.32	0.648	0.644	0.652	0.006	0.008	0.005
IDF	0.02	0.681	0.674	0.674	0.004	0.002	0.010
IDF	0.1	0.664	0.664	0.659	0.008	0.002	0.006
IDF	0.32	0.652	0.651	0.651	0.007	0.004	0.003

Figure 5.28: Matrix of performances of different values of the hyperparameters using SAD metric.

Looking at the results emerges that differences are not big. The best values are:

- TW = ONE. This is probably because text are already prepared as explained in chapter 3.4.1, so further elaborations introduce more noise.
- Alpha = 0.32. This confirms that the suggestion of [11] seems to works well.
- Eta = 0.005. This can be due again to the preparation of text that create more sparisty of words among topics (lower values of Eta).

5.6 Experiments 4: Best model on the complete dataset

Finally all is ready to run a text on the complete dataset of 6 millions of documents using what has demonstrated to perform better (LDA in combination of regression-based baptism method) with the optimal hyperparameters identified in chapter 5.5 (TW=TermWeight.ONE, Alpha=0.32 and Eta = 0.005). This test is performed using 100 iterations to maintain coherence with the last two experiments. The test has been performed on the same hardware to prove the feasibility of the proposed methodology even in an extremely big corpus of documents without the need of super cluster of powerful servers.

5.6.1 Results

As we can see results are in line with the results of the previous chapters.

Metric	Mean	Std
Global accuracy	0.42	0.0289
Average Precision	0.35	0.0074
Average Recall	0.36	0.0049

Table 5.28: Global accuracy, average precision and average recall results based on baptism regression method made on optimized LDA model on the complete set of 6 millions of documents

Metric	Mean	Std
CWQ	0.59	0.0119
SAD	0.70	0.0124

Table 5.29: CWQ and SAD results based on baptism regression method made on optimized LDA model on the complete set of on 6 millions of documents

5.6.2 Computational load

To perform this test a PC with 8 cores, RAM of 64GB, works for 20 hours proving the computational efficiency of LDA models.

Chapter 6

Conclusions

In the field of Natural Language Processing (NLP), the development of automatic topic identification is vital. With the exponential growth of textual data, there is an urgent need for a common methodology to develop and compare models that predicts real topics. This necessity arises due to the overwhelming volume of text data generated daily and its applications in content recommendation, sentiment analysis, information retrieval, and more.

On the other hand a rapid growth of topic models which identify latent topics in document has led to great results. But there is no direct connection between the discovered latent topic and the real topics of documents. Moreover different models aren't easily comparable and there is the need for methodology of standardization. Establishing such a methodology is essential for collaborative research, benchmarking, and the advancement of accurate and adaptable topic modeling techniques in NLP.

To address the challenges outlined in the previous paragraph, the key solution is to establish a dedicated methodology for the development and, critically, the comparative evaluation of topic modeling methods. This methodology should incorporate three essential components. First, it requires a sizable, diverse corpora of text data that reflects the complexity of real-world textual information, encompassing a broad spectrum of topics and subjects present even in mix on the single document. These corpora will serve as the foundation for testing the generative models. Second, a collection of automatic real topic identification algorithms based on latent topic engines. Finally, an array of comprehensive performance metrics is indispensable to assess the effectiveness, efficiency, and interpretability of these models. By integrating these elements

into a unified methodology, researchers and practitioners can systematically develop, test, and compare automatic topic modeling techniques, fostering progress in the field and ensuring their applicability in diverse domains.

To achieve the previous objective, I have made several significant contributions to my research endeavor.

First and foremost, it is developed a methodology to dynamically create an up-to-date big data corpora in all the most common languages, ready to be used as a supervised dataset for real topic identification engines. These corpora are notable for their unique characteristics, as it can be labelled using different taxonomies and the each document belongs to more than one topic with known proportions. With this methodology a corpus of 6 million texts in English language is created. This diverse collection of texts serves as the foundation for our research, enabling us to explore the complexities of mixed-topic textual data.

In addition, four innovative automatic real topic identification engines has been introduced, that use the capabilities of topic models and work with any topic model. These novel engines exploit the unique nature of our corpus, providing a robust framework for topic extraction and analysis.

Furthermore, the need for specialized performance metrics in scenarios involving mixed topics has been tackled. To address this, a set of novel performance metrics have been designed to evaluate the effectiveness and accuracy of real topic identification engines in the presence of documents with known mix of topics. These metrics are instrumental in quantifying the quality of results and can be utilized to fine-tune and improve the performance of models.

As a byproduct the real topics identification engines developed on top of topic models in this research candidate them self as alternatives to consolidated machine learning techniques as they proved to have results beyond expectations:

1. the approach demonstrates to be robust and to compete well with SLMs. So it can be used not just to compare topic models, but even as a supervised engine itself as an alternative.

2. it shows to be very efficient allowing analysis in less time and computational resources than the many other methodologies.

6.1 Operative applications

As it is stressed several times, the methodology proposed in this work is focused on the creation of a valid tool to compare generative topic models. Nevertheless, there are many immediate applications in real situations.

6.1.1 Selection of the best-unsupervised document classification model

It is common to use generative topic models to categorize document corpora into homogeneous classes. Till now the choice was based on their capacity to remain stable in their choices (coherence [23]) and/or their capacity to face new documents (perplexity [15]). But if someone wants to categorize documents, they will be most likely interested in the capacity to identify real topics so that the categorization has a sense in the real world. And this is what the new methodologies do.

6.1.2 Mix of real topic identification in big data with low computational request

New methods have proven to give good results even when used as a topic identification tool. Moreover, if we exclude the Linear Regression (which could be cornered by increasing the number of features) it has a much lower computational load than SLMs, which is very important in a big data environment.

6.1.3 Evaluation of the correctness of the used taxonomy

When we use SLMs we are prone to the limits of the taxonomy selected. Instead using the proposed methodology, that intercepts the hidden and intrinsic structure of topics before trying to map them on the taxonomy, anomalies in the taxonomy itself could emerge:

1. The need to introduce a further segmentation of topics that are too general (relevant subtopics)

2. The need to unify two or more topics because boundaries are not so definable (relevant supertopics)

3. The need to use different taxonomies (relevant pseudotopics)

6.1.4 Outlier identification in corpora of documents

Always because the proposed methodology operates at the beginning in an agnostic way looking for latent topics in documents, it can be used to find ”anomalies”, i.e. document that should belongs to a particular real topic but have a very different pattern of latent topics than the other of that real topic.

6.2 Next steps

Certainly, there are numerous possibilities for further improvement and investigation in our work. These directions encompass a wide array of possibilities aimed at enhancing the robustness and applicability of these results. For instance diversifying our testing to include different languages beyond English can help us understand the universality of our models. Exploring alternative taxonomies could shed light on more domain-specific applications. Comparing more topic models, even tuning hyperparameters of all of them, could lead to a lot of interesting insights on these models. Lastly, further refining our topic modeling methods, including the exploration of different latent topic engines, can push the boundaries of what we can achieve in this domain. These potential improvements promise to make our research more versatile and impactful.

In conclusion, our contributions in the form of a diverse big data corpus, innovative real topic identification engines, and specialized performance metrics represent significant steps forward in the realm of topic modeling and textual data analysis. These advancements not only advance our understanding of mixed-topic textual data but also provide valuable tools and insights for researchers and practitioners in various domains.

Appendix A

Terminology

Real topic is what we refer to when we read a document and say it speaks about this or that. The ”semantic” topic of a text. Real topics could be general as ”sport” or ”finance” but even specific as ”badminton” or ”public investments”

Taxonomy it is a hierarchical structured tree of topics. In a taxonomy we have a first level of topics that covers all the human knowledge. Each topics of the first level could have subtopics. Each subtopics could have sub-subtopics and so on. In this way each topic node of the taxonomy is related just to one first level topic.

Monotopic and mix of topics With monotopic document we refer to a document that speaks about just one topic. In general this event is rare and usually documents are about a mix of topics in different proportions (to sum to 1).

Latent topic is one of the topics identified by generative topic models, starting from a corpus of texts. They are created not referring to a real topic but based on the frequency of words.

Subtopic is a topic that is a part of another topic more ”general” that is higher in the taxonomy. In the examples before ”badminton” and ”public investments” are subtopic of ”sport” and ”finance” respectively.

Supertopic is a latent topic which is strongly related to more than one real topic. For example we can have a latent topic strongly related to the real topics 'Mathematics' and 'Physics'. This latent topic is a supertopic of those real topics and could be something like ”Science”.

Transversal topic is a latent topic not related to any real topic of the actual taxonomy. In fact generative topic models (that uses the distribution of word to identify latent topic) could

intercept other possible text classifications related to word distribution (such as sentiments, or cultural levels).

Pseudotopic is every latent topic that is not related to one specific real topic, is called pseudotopic. Supertopic and Transversal topic are example of pseudotopics.

Baptism is process of labeling a latent topic with the related real topic, in essence the process of assigning a real topic to a latent topic. When more than one latent topic are related to a real topic it means that the models separated this real topic in different subtopics. If we would like to separate them in different real topic we should use a lower level in the taxonomy increasing the granularity of the set of real topics used.

It is also possible that the model identifies ”latent topic” that cannot be related to a singular particular real topic. In this case we speak about ”pseudo-topic”. We are not interested in this pseudo-topic so we remove this latent topics from the analysis. In Figure A.1 an example is shown.

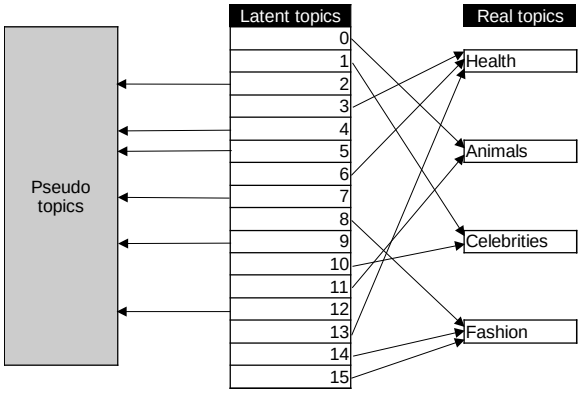

Figure A.1: Example of baptism of latent topics as real topics or pseudo-topics.

Appendix B

Hyperparameters

B.1 LDA, CTM, HDP, PAM and Baptism methods

- LDA:
  - term weighting (tw) = TermWeight.ONE (all words have the same weight)
  - number of topics =39*4
  - number of top words removed = 0
  - minimum document frequency of words = 0
  - minimum collection frequency of words = 0
  - hyperparameter of Dirichlet distribution for document-topic *alpha* = 0.1
  - hyperparameter of Dirichlet distribution for topic-word *eta* = 0.01
- CTM:
  - term weighting = TermWeight.ONE (all words have the same weight)
  - number of topics =39*4
  - number of top words removed = 0
  - minimum document frequency of words = 0
  - minimum collection frequency of words = 0

  - small smoothing value for preventing topic counts to be zero = 0.1
  - hyperparameter of Dirichlet distribution for topic-word = 0.01

- HDP:
  - term weighting = TermWeight.ONE (all words have the same weight)
  - initial number of topics = 39
  - number of top words removed = 0
  - minimum document frequency of words = 0
  - minimum collection frequency of words = 0
  - concentration coeficient of Dirichlet Process for document-table *alpha* = 0.1
  - hyperparameter of Dirichlet distribution for topic-word *eta* = 0.01
  - concentration coeficient of Dirichlet Process for table-topic *gamma* = 0.1

- PAM:
  - term weighting = TermWeight.ONE (all words have the same weight)
  - the number of super topics between = 39
  - the number of sub topics between = 39*4
  - number of top words removed = 0
  - minimum document frequency of words = 0
  - minimum collection frequency of words = 0
  - initial hyperparameter of Dirichlet distribution for document-super topic *alpha* = 0.1
  - initial hyperparameter of Dirichlet distribution for super-sub topic *alpha* = 0.1
  - hyperparameter of Dirichlet distribution for sub topic-word *eta* = 0.01

- Topwords based Baptism:
  - number of unique top words selected from each real topic = 5
  - number of top words considered in each latent topic = 10

B.2 Regressors

- Linear Regression (Scikit-learn)
  - fit intercept = True
  - force coefficients to be positive = False
- Ada Boosting (Scikit-learn)
  - base estimator from which the boosted ensemble is built = None
  - maximum number of estimators at which boosting is terminated = 50
  - weight applied to each classifier at each boosting iteration = 1.0
  - algorithm to use = 'SAMME.R'
- Bagging Regressor (Scikit-learn)
  - base estimator to fit on random subsets of the dataset. If None, then the base estimator is a DecisionTreeRegressor = None
  - number of base estimators in the ensemble = 10
  - number of samples to draw from X to train each base estimator = 1.0
  - number of features to draw from X to train each base estimator = 1
  - whether samples are drawn with replacement = True
  - whether features are drawn with replacement = False
  - whether to use out-of-bag samples to estimate the generalization error. Only available if bootstrap=True. = False
  - warm start (reuse the solution of the previous call to fit as initialization)=False
- Gradient Boosting Regressor (Scikit-learn)
  - loss function to be optimized = 'log_loss'
  - learning rate = 0.1
  - number of estimators = 100

  - fraction of samples to be used for fitting the individual base learners = 1
  - function to measure the quality of a split = 'friedman_mse'
  - minimum number of samples required to split an internal node = 2
  - minimum number of samples required to be at a leaf node = 1
  - minimum weighted fraction of the sum total of weights required to be at a leaf node = 0
  - maximum depth of the individual regression estimators = 3
  - decrease of the impurity required to split = 0
  - object used to compute the initial predictions = None
  - number of features to consider when looking for the best split = None
  - alpha-quantile of the huber loss function and the quantile loss function = 0.9
  - maximum of leaf nodes = None
  - warm start (reuse the solution of the previous call to fit as initialization) =False
  - proportion of training data to set aside as validation set for early stopping = 0.1
  - whether to use early stopping to terminate training when validation score is not improving = None
  - tolerance for the early stopping = 0.0001
  - complexity parameter used for Minimal Cost-Complexity Pruning = 0.0

- Random Forest Regressor (Scikit-learn)
  - number of estimators = 100
  - criterion to measure the quality of a split = 'squared_error'
  - maximum depth of the tree = 5 (*)
  - minimum number of samples required to split an internal node = 2
  - minimum number of samples required to be at a leaf node = 1
  - minimum weighted fraction of the sum total of weights required to be at a leaf node = 0

  - number of features to consider when looking for the best split = 1
  - maximum number of leaf nodes = None
  - decrease of the impurity required to split = 0
  - whether bootstrap samples must be used when building trees = True
  - whether to use out-of-bag samples to estimate the generalization score = False
  - warm start (reuse the solution of the previous call to fit as initialization)=False
  - complexity parameter used for Minimal Cost-Complexity Pruning = 0
  - number of samples to draw from X to train each base estimator if boostrap is True= None

- KNN regressor (Scikit-learn)
  - number of neighbors to use by default for kneighbors queries = 5
  - weight function used in prediction = 'uniform'
  - algorithm used to compute the nearest neighbors = 'auto' (it attempts to decide the most appropriate algorithm based on the values passed to fit method)
  - leaf size passed to BallTree or KDTree = 30
  - power parameter for the Minkowski metric = 2
  - metric to use for distance computation = 'minkowski'

Appendix C

Websites crawled in the first project

Health

- http://www.benessere-news.it/
- https://notiziebenessere.it/
- https://www.corriere.it/salute/
- https://www.silhouettedonna.it/

Celebrities

- https://www.biccy.it/
- https://www.gossip.it/
- https://www.ilgossip.net/

Fashion

- https://www.corriere.it/moda/
- https://www.elle.com/it/moda/
- https://www.iodonna.it/moda/
- https://www.marieclaire.com/it/moda

- https://www.vogue.it/

Animals

- https://imieianimali.it/
- https://www.amoreaquattrozampe.it/
- https://www.animalipucciosi.com/
- https://www.animalpedia.it/
- https://www.corriere.it/animali/
- https://www.notizieanimali.com/
- https://www.passione-animali.it/

Animals and Health

- https://www.amoreaquattrozampe.it/cani/salute/
- https://www.amoreaquattrozampe.it/gatti/salute-gatti/
- https://www.amoreaquattrozampe.it/cavalli/salute-cavalli/

Celebrities and Fashion

- https://www.elle.com/it/moda/street-style/

Appendix D

Tables for the first project

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
	me	and	products	dog	cases	hair	ciprofloxacin	project
	said	to	water	dogs	swab	legs	kids	fashion
	says	is	oil	animals	diet	makeup	vardenafil	vogue
	friends	music	face	pedigree	deaths	color	vaccine	brand
	say	for	ingredients	cats	food	feet	covid	job
	have	that	products	puppy	virus	lips	tumor	milan
	want	on	hair	pedigrees	foods	yoga	cancer	design
	father	album	cream	need	none	exercises	disease	accross
	maybe	with	properties	owner	eat	towards	tumors	city
	that	you	natural	rabbit	phase	arms	virus	designer
	singer	festival	naturals	names	meat	face	patients	fashion
	instagram	it	minutes	animal	protein	up	test	future
	love	students	oils	size	positives	nail	patient	research
	social	we	help	nature	fats	makeup	data	new
	mother	my	hamster	company	quantity	position	study	culture
	written	this	seeds	shepherd	number	leg	national	director
	person	video	shampoo	play	region	haircut	doctors	beauty
	story	musical	sugar	kids	beginning	eyes	alprazolam	space
	no	trump	water	fear	deaths	movement	vaccines	exhibition
	you	are	sun	hello	check	back	therapies	projects
Real Topic	Celebrities	—	—	Animals	Health	—	Health	Fashion
Label	Social Media	Music	Beauty (Products)	Dogs	COVID	Beauty (Body)	Diseases	Fashion

	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16
	star	fashion	photo	vip	patients	sleep	species	cat
	cinema	look	daughter	natalia	mg	lansoprazole	animals	dog
	red	portfolio	rome	episode	treatment	blood	fishes	cats
	carpet	style	kaia	mastrota	must	pain	insects	vet
	series	clothes	lex	big_brother	dose	stress	males	fur
	main_character	jeans	enne	brother	paragraph	disease	specimen	horse
	kate	fashion	mum	giulia	effects	oxygen	birds	kitty
	harry	spring	social	men	data	cells	females	might
	queen	clothes	sanremo	photo	administration	symptom	animal	animal
	jennifer	winter	cindy	rodriguez	reaction	issue	turtle	food
	meghan	pants	milan	d'urso	drug	capable	female	teeth
	oscar	brand	birthday	share	renal	factor	male	symptoms
	new	shoes	couple	belen	therapy	brain	big	dogs
	venice	runway	beautiful	tempation	risk	level	turtles	feline
	hollywood	summer	tv	island	drugs	activity	live	paw
	york	maison	chiara	francesco	supervisory_report	physical	mammal	animal
	director	chanael	francesco	balivo	concomitant	cause	little	diseases
	prince	wear	laura	caterina	adverse	system	hunt	puppies
	lady	jacket	son	famous	side	disease	size	issues
	plot	accessories	marco	read	use	wealth	bees	animal
Real Topic	Celebrities	Fashion	Celebrities	Celebrities	Health	Health	Animals	Animals
Label	Gossip	Fashion show	VIP	Reality	Adverse Events	Health	Animals (non-pet)	Cats

Table D.1: Section 4.2.1: big data scenario. 20 most probable words for each of the $K = 16$ latent topics detected by the LDA model. Please note that words are translated from the Italian language.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
	fashion	story	cat	kaia	audience	dog	patients	equal
	fashion	francesco	dogs	cindy	must	hair	treatment	dog
	brand	that	cats	know	species	dogs	mg	share
	designer	ilary	dogs	news	degree	cat	data	clarithromycin
	project	says	animals	both	meat	hair	paragraph	system
	vogue	mara	symptoms	friend	check	vet	drug	nifedipine
	style	emma	species	smartphone	products	animals	drugs	report
	costanza	francesca	swab	gregoraci	none	vitamin	reaction	general
	hair	hand	food	free	can	diet	therapy	information
	caracciolo	book	must	crawford	reported	animal	dose	effects
	collection	millions	species	hours	effects	quantity	administration	animals
	designer	blasi	renal	gossip	ramipril	foods	effects	doctor
	gucci	hourses	cases	work	effect	benefits	cases	grater
	vieri	travel	puppies	app	size	fruit	side	kids
	boots	marrone	diseases	that_is	sustainable	product	infections	must
	new	all	pet	some	safety	ears	adverse	specialist
	model	little	animal	reality	oil	paw	concomitant	check
	loves	past	kitty	android	prescription	fish	ciprofloxacin	study
	couture	told	animal	download	oil	diabete	eo	data
	haircut	big	test	best	massage	protein	doctor	use
Real Topic Label	Fashion Beauty	VIP Gossip	Animali domestici Salute Animale				Drugs/Desease	

	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16
	tail	temptation	natalia	men	look	vip	photo	evening
	virus	colors	mastrota	belen	collection	enne	video	heart
	horse	and	issues	rodriguez	fashion	son	instagram	back
	disease	legs	article	marco	clothes	lex	marriage	daughter
	nails	to	BigBrother	stefano	jeans	social	york	christmas
	vaccine	fabric	episode	d'urso	suit	couple	cinema	week
	fundamental	effect	tells	andrea	spring	wife	birthday	friends
	respect	euro	giorgio	luca	black	mum	star	milan
	birds	position	read	me	pants	husband	september	alessanfra
	towards	island	caterina	brother	accessories	show	main_character	seems
	eyes	high	balivo	name	style	fan	alessia	photo
	age	yoga	giulia	said	red	children	Sunday	few
	need	michael	face	michelle	clothes	maria	daughter	job
	death	leg	sky	also	shoes	elisabetta	tv	evaluation
	risk	neck	repeated	gemma	white	rome	beautiful	sara
	lips	point	birth	federica	models	fabrizio	evening	girlfriends
	cells	oxygen	seventh	say	colors	corona	model	have
	animal	shirt	immediately	crisis	pair	new	super	come
	come	suit	accepts	throne	pink	red	milan	arrived
	covid	simple	navigation	given	color	actress	party	known
Real Topic Label		Clothes		VIP				

Table D.2: Section 4.2.1: big data scenario. 20 most probable words for each of the $K = 16$ latent topics detected by the CTM model. Please note that words are translated from the Italian language.

Bibliography

[1] J. Aitchison. 'The Statistical Analysis of Compositional Data'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 44.2 (1982), pp. 139–160. DOI: https://doi.org/10.1111/j.2517-6161.1982.tb01195.x.

[2] J. Aitchison. *The Statistical Analysis of Compositional Data*. Second. London: The Blackburn Press, 2003. ISBN: 9789401083249.

[3] Mehdi Allahyari et al. 'A Knowledge-based Topic Modeling Approach for Automatic Topic Labeling'. In: *International Journal of Advanced Computer Science and Applications* 8.9 (2017).

[4] R. Artusi, P. Verderio and E. Marubini. 'Bravais-Pearson and Spearman Correlation Coefficients: Meaning, Test of Hypothesis and Confidence Interval'. In: *The International Journal of Biological Markers* 17.2 (2002), pp. 148–151. DOI: 10.1177/172460080201700213.

[5] D.M. Blei and J.D. Lafferty. 'A correlated topic model of science'. In: *The Annals of Applied Statistics* 1(1) (2007), pp. 17–35. DOI: 10.1214/07-AOAS114.

[6] D.M. Blei, A.Y. Ng and M.I. Jordan. 'Latent Dirichlet Allocation'. In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022. DOI: 10.5555/944919.944937.

[7] Teh Jordan Beal Blei. 'Hierarchical Dirichlet processes'. In: *Journal of the American Statistical Association* (2005), pp. 1566–1581. DOI: 10.1198/016214506000000302.

[8] Kenneth Church and Patrick Hanks. 'Word Association Norms, Mutual Information, and Lexicography'. In: *Computational Linguistics* 16 (July 2002). DOI: 10.3115/981623.981633.

[9] David A. van Dyk. 'The Role of Statistics in the Discovery of a Higgs Boson.' In: *Annual Review of Statistics and Its Application* 1.1 (2013), pp. 41–59. DOI: 10.1146/annurev-statistics-062713-085841.

[10] Thomas L. Griffiths and Mark Steyvers. 'Finding scientific topics'. In: *Proceedings of the National Academy of Sciences of the United States of America* 101.SUPPL. 1 (2004), pp. 5228–5235. DOI: 10.1073/pnas.0307752101.

[11] Thomas L. Griffiths and Mark Steyvers. 'Finding scientific topics'. In: *Proceedings of the National Academy of Sciences* 101.suppl_1 (2004), pp. 5228–5235. DOI: 10.1073/pnas.0307752101.

[12] Dongbin He et al. 'Automatic topic labeling using graph-based pre-trained neural embedding'. In: *Neurocomputing* 463 (2021), pp. 596–608. ISSN: 0925-2312.

[13] Rafiq H Hijazi and Robert W Jernigan. 'Modelling compositional data using Dirichlet regression models'. In: *Journal of Applied Probability & Statistics* 4.1 (2009), pp. 77–91.

[14] Ioana Hulpus et al. 'Unsupervised Graph-Based Topic Labelling Using Dbpedia'. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. WSDM '13. Rome, Italy: Association for Computing Machinery, 2013, pp. 465–474.

[15] F. Jelinek et al. 'Perplexity—a measure of the difficulty of speech recognition tasks'. In: *The Journal of the Acoustical Society of America* 62.S1 (Aug. 2005), S63–S63. ISSN: 0001-4966. DOI: 10.1121/1.2016299.

[16] G. Jingxian and Q. Yong. 'Selection of the optimal number of topics for LDA topic model – Taaking patent policy analysis as an example'. In: *Entropy* 23 (2021), p. 1301.

[17] Jey Han Lau et al. 'Automatic Labelling of Topic Models'. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 1536–1545.

[18] Jey Han Lau et al. ‘Best Topic Word Selection for Topic Labelling’. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. COLING ’10. Beijing, China: Association for Computational Linguistics, 2010, pp. 605–613.

[19] Wei Li and Andrew McCallum. ‘Pachinko allocation: DAG-structured mixture models of topic correlations’. In: *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 577–584.

[20] Davide Magatti et al. ‘Automatic Labeling of Topics’. In: *2009 Ninth International Conference on Intelligent Systems Design and Applications*. 2009, pp. 1227–1232. DOI: 10.1109/ISDA.2009.165.

[21] Marco Maier. *DirichletReg: Dirichlet Regression for Compositional Data in R*. English. WorkingPaper 125. Apr. 2014.

[22] 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*. KDD ’07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 490–499.

[23] D. Mimno et al. ‘Optimizing semantic coherence in topic models’. In: *Proceedings of Empirical Methods in Natural Language Processing*. 2011, pp. 262–272.

[24] Rishabh Misra and Jigyasa Grover. *Sculpting Data for ML: The first act of Machine Learning*. Jan. 2021.

[25] David Newman et al. ‘Distributed Algorithms for Topic Models’. In: *Journal of Machine Learning Research* 10.62 (2009), pp. 1801–1828.

[26] Frank Nielsen. ‘Hilbert’s simplex distance: A non-separable information monotone distance’. In: (Oct. 2021). DOI: 10.13140/RG.2.2.31240.96001.

[27] M. Röder, A. Both and A. Hinneburg. ‘Exploring the space of topic coherence measures’. In: *Proceedings of the eighth ACM international conference on Web search and data mining*. 2015, pp. 399–408.

[28] S. Sbalchiero and M. Eder. ‘Topic modeling, long texts and the best number of topics. Some problems and solutions’. In: *Quality and Quantity* 54 (2020), pp. 1095–1108.

[29] S. Syed and M. Spruit. ‘Full-text or abstract? Examining topic coherence scores using latent Dirichlet allocation’. In: *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*. 2017, pp. 165–174.

[30] Andrew Wilson and Peter Chew. ‘Term Weighting Schemes for Latent Dirichlet Allocation.’ In: Jan. 2010, pp. 465–473.

[31] L. Yao, D. Mimno and A. McCallum. ‘Efficient Methods for Topic Model Inference on Streaming Document Collections’. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009, pp. 937–946.