University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Applying Topic Modeling Techniques for Semantic Search in Public Administration Documents

Supervisor
Prof. Roberto Confalonieri
University of Padova

Master Candidate
Giovanni Zerbo

Student ID
2089386

Academic Year
2023-2024

To my parents.

To everyone who made me smile in the last two years.

(These were not mutually exclusive, most of the time)

Abstract

As an everyday Italian citizen, finding information about the structure and functioning of public administration could be challenging at times, due to the amount of bureaucracy and different regulations. A Virtual Assistant that guides the user through the thousand of documents could be useful to help with the problem. This thesis studies how, given an user input (query), it's possible to improve the semantic search over the public administration documents in order to return the most relevant ones. This is done by applying topic modeling to the database of documents, in order to sort and categorize them in a more meaningful and interpretable structure for the search.

Contents

Abstract v

List of figures ix

List of tables xi

Listing of acronyms xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objective . . . 2
- 1.3 Structure of the thesis . . . 2

2 Context 3
- 2.1 IVA4PA . . . 3
- 2.2 Architecture . . . 3
  - 2.2.1 Embedding Model . . . 4
  - 2.2.2 Semantic Database . . . 4
  - 2.2.3 Retriever . . . 5
  - 2.2.4 Generative Model . . . 5

3 Review of the state of the art 7
- 3.1 Survey on topic modeling algorithms and applications . . . 7
- 3.2 Review of topic modeling methods . . . 9
- 3.3 Use of Topic Modeling for Improvement of Quality in the Task of Semantic Search . . . 10
- 3.4 Semantic Search and Summarization of Judgments Using Topic Modeling . . . 11
- 3.5 Evaluation of semantic web search . . . 11
- 3.6 Review of the state of the art . . . 12

4 Problem Statement 13

5 Problem Solution 15
- 5.1 Text Pre-Processing . . . 16
  - 5.1.1 Cleaning . . . 17
  - 5.1.2 Stop Word Removal . . . 17

5.1.3 Stemming and Lemmatization . . . 17
5.1.4 Tokenization . . . 18
5.2 Latent Dirichlet Allocation . . . 18
5.3 Non-negative Matrix Factorization . . . 19
5.3.1 Topic Coherence . . . 20
5.3.2 Model Choice . . . 21
5.3.3 Topics . . . 21
5.4 Filtering Search Results with Topic Modeling . . . 23
5.4.1 Keywords . . . 23
5.4.2 Query Expansion . . . 24
5.4.3 Filtering . . . 25
5.4.4 Potential practical issues . . . 25
5.5 Ranking Search Results with Topic Modeling . . . 26
5.5.1 Code . . . 26

6 Evaluation **29**
6.1 IVA4PA Semantic Search . . . 29
6.2 Normalized Discounted Cumulative Gain . . . 30
6.3 Ground Truth and Relevance Values . . . 31
6.4 Results . . . 32
6.5 Results Discussion . . . 32

7 Conclusion **35**
7.1 Challenges and Future Work . . . 35

References **37**

Acknowledgments **39**

Listing of figures

2.1 General Structure of IVA4PA 4
2.2 Functional Diagram of IVA4PA 6

3.1 Evaluation of topic modeling algorithms against the M10 Dataset 8
3.2 Decision tree for topic modeling algorithm selection 10

4.1 Example header of an "Amministrazione Trasparente" Document 14

5.1 General structure of the solution (in red) 16
5.2 LDA Algorithm Structure [1] 19
5.3 NMF on topic modeling [2] 20
5.4 The Filtering stage 22
5.5 The Filtering stage 24
5.6 The Re-Ranking Stage 27

Listing of tables

3.1 Categories of Topic Modeling Algorithms 8
3.2 Review of Topic Modeling Algorithms 9

6.1 Example output of the original semantic search 30
6.2 Example output after filtering and re-ranking 31
6.3 Performances on one run of test-queries 34

Listing of acronyms

IVA4PA . . . . . Intelligent Virtual Assistant for Public Administration

LDA . . . . . Latent Dirichlet Allocation

NMF . . . . . Non-negative Matrix Factorization

NLP . . . . . Natural Language Processing

DCG . . . . . Discounter Cumulative Gain

NDCG . . . . . Normalized Discounted Cumulative Gain

NPMI . . . . . Normalized Pointwise Mutual Information

LLM . . . . . Large Language Model

1 Introduction

Semantic search is a technique in information retrieval that focuses on understanding the contextual meaning behind a user's query when searching over a set of documents or texts, rather than simply using a matching keyword logic. It's useful to deliver more relevant and accurate results by capturing deeper relationships between words, concepts, and the query context. In a digital era where users demand precise and personalized answers to increasingly complex queries, semantic search has become especially valuable in areas like e-commerce, healthcare, and knowledge retrieval systems.

1.1 Motivation

This thesis is inspired by the author's work in the curricular stage experience at the Padua municipality, on a project aiming to develop a Virtual Assistant that could help the citizens to easily acquire any information related to the public administration. Semantic search was a key feature of this project, since the Virtual Assistant response has to take into account, from the thousands of public administration documents, the documents that are relevant to the user query in the most accurate way possible.

1.2 Objective

With the goal in mind to improve the performances of a semantic search over the documents, we incorporated a topic modeling algorithm into the document search process. Topic modeling is a widely use technique in natural language processing, and the idea is to take advantage of this method to increase the Virtual Assistant capability of understanding the context and the concepts of the query and the public administration documents. This idea is dictated from the fact that the Virtual Assistant will make use of Large Language Models to generate its answer, therefore the semantic value of the input documents it will use as a generating basis is key to the success of the project.

1.3 Structure of the thesis

First of all, in chapter 2 we are going to explain the context of the project from which this thesis is inspired. Then we will discuss the state of the art 3 on topic modeling and its relationship with semantic search, in order to accurately formulate the problem 4. After that the solution is presented in 5, involving two main steps, whose results will be evaluated 6 and discussed in the last chapter before the conclusion, where we will discuss future improvements and how the work of this thesis could be implemented in the municipality project.

2 Context

2.1 IVA4PA

The work done on this thesis was motivated by the author internship experience at the Municipality of Padua, where he contributed to the first phase of the project IVA4PA (Intelligent Virtual Assistant for Public Administration).

This project, coordinated by Akera S.R.L., has the goal to develop a virtual assistant for the website of the Padua Municipality (and possibly other municipalities in the future) that could help the citizens navigate through the huge amount of public information that is spread across the website. The Virtual Assistant has a conversational approach, and will return an answer to the user using natural language by summarizing the documents found in the website that are correlated to the user question.

In the first phase of the project, the database of documents is limited to a specific area in the website, called "Amministrazione Trasparente", where about 20.000 public documents are stored. This same data will be used in the thesis in later developments.

2.2 Architecture

The Architecture of IVA4PA can be split in 4 main components:

- Embedding Model

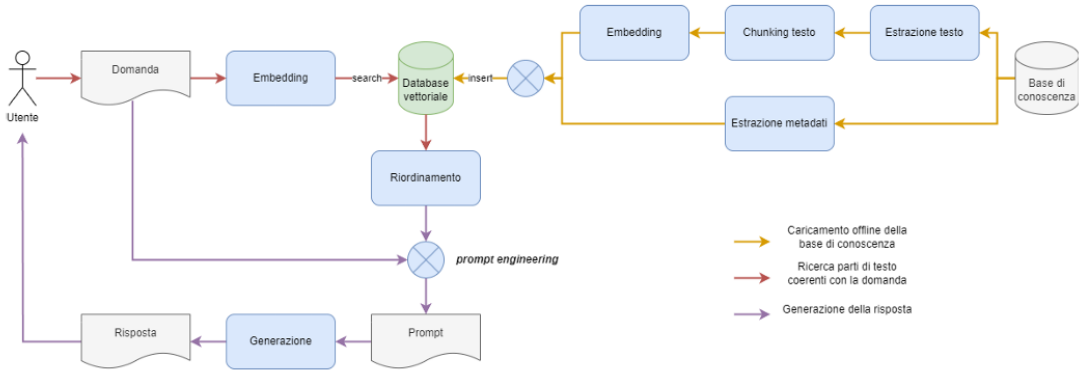


Figure 2.1: General Structure of IVA4PA

- Semantic Database
- Retriever
- Generative Model

The first two components are the ones that that will be majorly employed in this thesis, while the last two will not be analyzed in depth since they mainly regard the development of the conversational aspect of the Virtual Assistant.

2.2.1 Embedding Model

An embedding model is a machine learning model that has the goal to convert different types of data (text data, in this case) into continuous numerical vectors that aim to capture its semantic meaning. Ideally, when two sentences (or words) have similar meaning, their respective embedding representation in the vector space should be also similar, with the angle between the vectors being smaller as the vectors get close to each other. The model chosen for IVA4PA is one of the best open source embedding models in the task of semantic search [3] and utilizes vectors in a 768 dimensions space.

The best embedding models don't just capture the meaning of words in a sentence, but also the relative position of the words, thanks to the transformers technology used during the training.

2.2.2 Semantic Database

Due to the limitation on the number of characters (512) that the embedding model can take as input, in order to store all the "Amministrazione Trasparente" documents in a database, a pro-

cess called "Chunking" is performed. Every document is split into chunks of fixed length, that are then passed to the embedding model and stored in a vector database. The technology used to implement the database is Weaviate, an open source vector database, that offers the option to attach useful metadata to the data unit, such as the Document ID or the Document Text, and allows a mixture of semantic search (search in the vector space) and keyword search inside the database. This way, from the 20.000 documents, a bigger database containing approximately 357.000 chunks is created, with the following metadata:

- **id**: an unique code that identifies the chunk
- **ChunkText**: a string of maximum 512 characters that contains the text of the chunk
- **DocSourceUrl**: the URL that re-directs to the original document of the chunk
- **nChunks**: the number of chunks in which the original document was splitted into

2.2.3 Retriever

The Retriever is a component that is linked to the generative model, with the function of providing it with all the information relative to the documents, and making sure that only that information is used to formulate an answer to the user. This is crucial to make sure that the generative model generates an appropriate answer, without utilizing the external knowledge that the model itself has acquired during its original training.

2.2.4 Generative Model

The job of the generative model is to return an appropriate response to the user query, by using the information of the documents obtained in the search stages. This task can be performed by a Large Language Model (LLM). A crucial restriction on the choice of the model is the capability of understanding the Italian language, which mainly depends from the dimension of the model and the quality of the data used in the training phase.

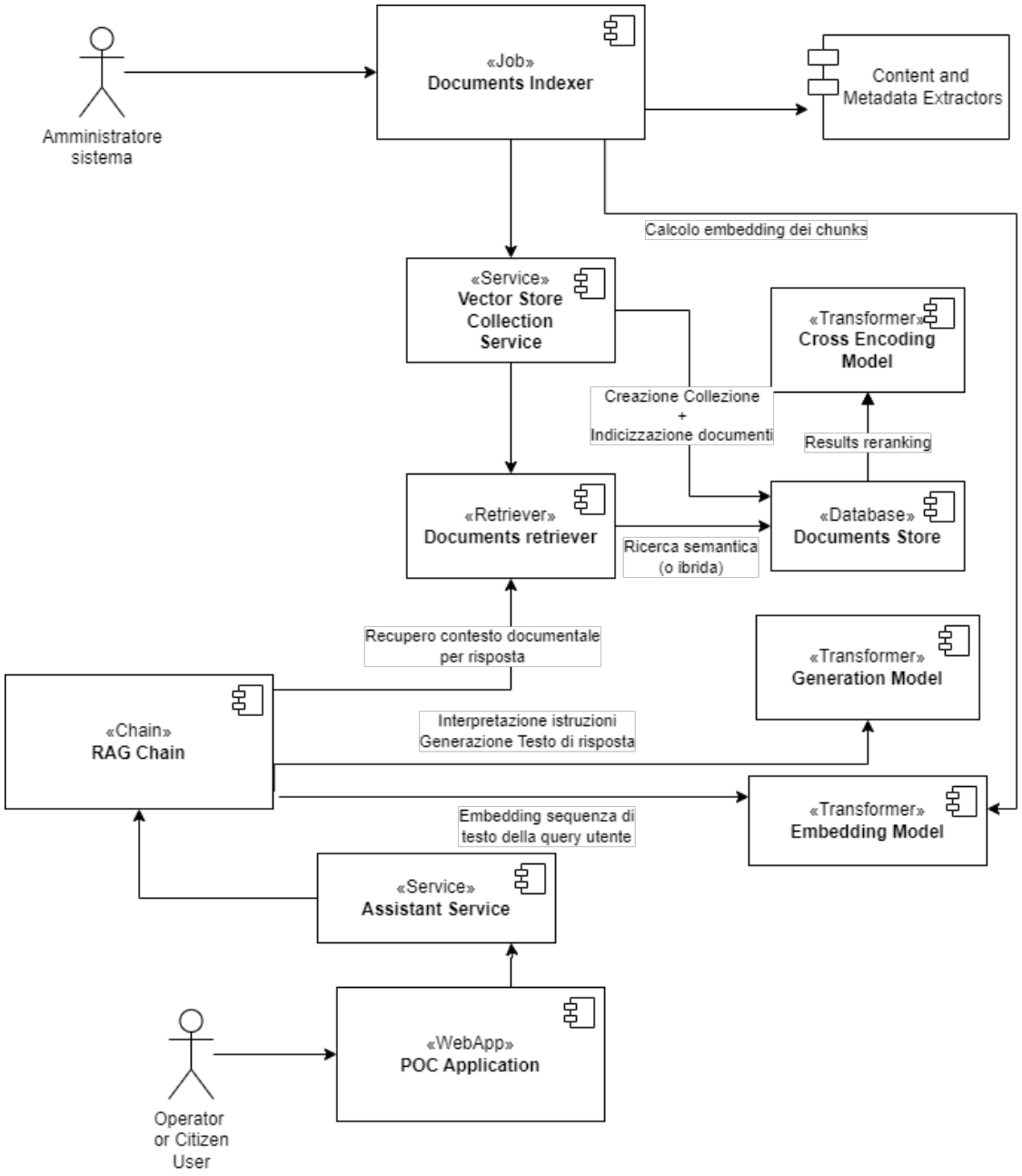


Figure 2.2: Functional Diagram of IVA4PA

3

Review of the state of the art

3.1 Survey on topic modeling algorithms and applications

Abdelrazek et al. [4] present a literature survey on the different categories of topic modeling algorithms: Algebraic, Fuzzy, Probabilistic and Neural. They talk about the main application of topic modeling and exhibit the metrics to evaluate the performance of a topic modeling algorithm, from the human interpretability of the topics to the computational complexity of the algorithm. In table 3.1 it's possible to see the positives and the negatives about the categorization mentioned above.

In the results of the paper the authors show the performances of 7 different topic modeling algorithms on two datasets, measured using 4 different metrics:

Coherence: Measures the interpretability of the topics from an human point of view, by observing the lexical similarity between pairs of words that belong to the same topic.

Diversity: It describes the topics uniqueness by calculating the percentage of unique words in the top K words of each topic. Diversity is highly influenced by the parameter of the number of topics, since the greater this parameter is, the more likely will be that 2 or more topics will have similar meaning and therefore a low uniqueness in their top words.

Stability: Measures how much the topics change over different runs of the algorithm. A stability value lower than 1 indicates that the topics can change if the input ordering of the

Category	Strengths	Limitations
Algebraic	Simple, intuitive, and computationally relatively efficient. Some adaptations can handle short text documents	Provides no solid statistical foundation. And does not define a generative data model
Fuzzy	It can handle sparsity in short text documents (for example, tweets)	Most of the applications focus on medical data
Probabilistic	Simple, intuitive, extensible, and interpretable	Inference becomes complicated with increased model complexity
Neural	Flexibility of joint training, optimizing for topic coherence, attaining complex models, scalability	interpretability of model parameters. Also, it generally cannot handle sparsity

Table 3.1: Categories of Topic Modeling Algorithms

data is different from one run to another. It's measured by calculating the similarity of the top n words of all the topics in 2 separate runs of the algorithm.

Time: The amount of time the algorithm took to execute. It is an indicator of the algorithm efficiency and its computational complexity

Figure 3.1: Evaluation of topic modeling algorithms against the M10 Dataset

Model	Coherence	Diversity	Stability	Time (Seconds)
LSI	-0.0235	0.546	1	1.6136
NMF	-0.0311	0.713	0.6014	2.3121
LDA	-0.0604	0.627	0.692	2.2156
HDP	-0.5049	0.668	0.3194	7.3764
ETM	-0.0014	0.383	0.8641	32.9827
CTM	0.0197	0.98	0.7739	12.0966
ProdLDA	-0.0017	0.971	0.7881	12.7665

3.2 Review of topic modeling methods

Vayansky [5] covers a comprehensive review of the most relevant topic modeling methods, starting from the classic LDA (Latent Dirichlet Allocation) to more advanced methods like Dynamic Topic Models and Correlated Topic Models. They examine the differences in these algorithms and how to select the better algorithm given the different type of task that the user needs to accomplish.

Algorithm	**Key Features**	**Strengths**	**Limitations**
Latent Dirichlet Allocation (LDA)	Probabilistic generative model using Dirichlet priors for topic distributions	Simple, interpretable results	Struggles with short texts; assumes topics are independent
Non-Negative Matrix Factorization (NMF)	Matrix decomposition technique using non-negative constraints to identify topics	Deterministic, robust to sparse data, low computational cost	Requires preprocessing, lacks probabilistic interpretation
Correlated Topic Modeling (CTM)	Extends LDA by modeling topic correlations	Captures topic correlations effectively	Increased complexity, computationally expensive
Pachinko Allocation Model (PAM)	Models hierarchical topic structures with a directed acyclic graph	Captures topic hierarchies and correlations, flexible	Computationally expensive, hard to implement compared to simpler models
Dynamic Topic Modeling (DTM)	Extends LDA to model temporal evolution of topics over time	Tracks how topics change over time, useful for time-series data	Requires temporal data, high computational demands
Self-Aggregating Topic Modeling (SATM)	Uses clustering and graph-theoretic approaches to self-organize topics without priors	Flexible, avoids strict assumptions of probabilistic models, suitable for large datasets	Results depend on dataset characteristics

Table 3.2: Review of Topic Modeling Algorithms

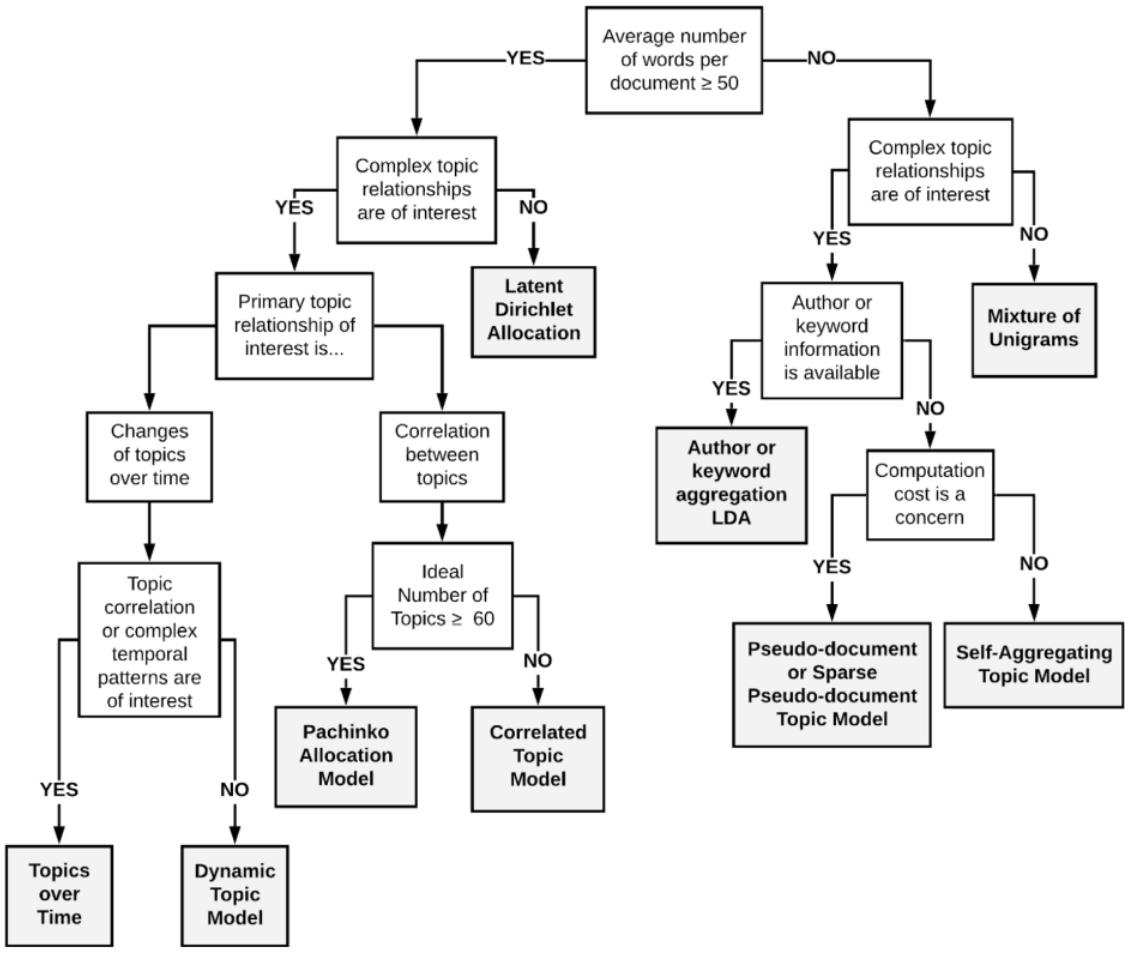


Figure 3.2: Decision tree for topic modeling algorithm selection

3.3 Use of Topic Modeling for Improvement of Quality in the Task of Semantic Search

Nikolaev et al. [6] propose an approach to improve a semantic search algorithm on educational courses by using a filtering mechanism of the documents that relies on probabilistic topic modeling. They investigate the impact of different hyperparameters in the topic modeling algorithm and how they affect the quality of the semantic search. The topic modeling, differently from the approach used by Wu et al., is incorporated after the semantic search has already took place, and a list of ordered documents is returned from a given query. Here the topic distribution of each document in the ranked list is examined, and the main topics of each document are

presented to an expert in the form of the topics keywords. The job of the expert is to select the keywords that are most correlated with the original query, in order to filter out every document in which the main topics are not represented by any of the selected keywords. This is done in order to improve the interpretability of the semantic search results, by having an additional control over the search that follows an human perception of the data.

3.4 Semantic Search and Summarization of Judgments Using Topic Modeling

Wu et al. [7] Introduce a new method that implements the concept of topic modeling inside the procedure of semantic search, in a vector space containing documents of legal judgments. They show how, following user evaluation, their method results highly effective in matching legal judgments to user queries, not just on a keyword level but also on a semantic level. The core concept of the idea is to measure the similarity between the user query q and the documents by how much the topics of those entities overlap.

After training an LDA model on the database of legal judgments (documents from now on) the authors obtain a set of N topics, where each topic T_i is represented by a "top words" vector $[w_{i1}, w_{i2}..., w_{iN}]$. Every top words has a score assigned to it, indicating how strong its association with the topic is. By using a simple embedding model (word2vec), it's possible to formulate a vector representation of these top words, and by taking a weighted mean of this top words' vectors, with respect to their scores, a set of N topic semantic vectors $[v_{T1}, v_{T2}..., v_{TN}]$ is created. A similar process is applied to the query, thus obtaining a query semantic vector. By then calculating the cosine similarity between the query semantic vector and Topic semantic Vectors a new vector of probabilities is returned, that indicates how much every topic is correlated with the query. Finally it's possible to confront this probability vector with the vectors that represent the topic distribution of the documents, returned from the LDA model. The cosine similarity between the two probability vectors returns a final score for each document, in order to measure the relevance of each document with respect to the query.

3.5 Evaluation of semantic web search

Elbedweihy et al. [8] review the methodologies, challenges, and opportunities in evaluating semantic search systems. They distinguish two main evaluation approaches: System-Oriented

Evaluation and User-Oriented Evaluation.

System-Oriented Evaluation is useful to test algorithmic performance on test collections, evaluating the performance of the search system itself and focusing on how well it retrieves relevant information based on a given query. The measures used for this goal are either Binary-Relevance measures, like precision, recall and MMR (Mean Reciprocal Rank), that assume documents to be either relevant or not relevant, or Graded-Relevance measures. The latter assign varying levels of relevance to documents, enabling more nuanced metrics like Normalize DCG (Discounted Cumulative Gain) and Expected Reciprocal Rank (EER).

User-Oriented Evaluation evaluates search systems based on the user experience, focusing more on user satisfaction and effectiveness in achieving the user's goal. These type of measures usually require gathering of user feedback through questionnaires or observation of user behaviour. They are more capable of capturing contextual factors that may elude System-Oriented evaluation, but on the other hand they are more expensive in terms of time and resources, and are subjective to experimental conditions.

3.6 Review of the state of the art

Summarizing, the state of the art today proposes a huge variety of options to implement topic modeling, with algorithms that can adapt to a lot of different tasks, depending on the size of the dataset, the goal, the length of the documents, and other parameters. The more widely used and baseline models are Latent Dirichlet Allocation and Non-negative Matrix Factorization, that are indicated for a general approach on a large dataset, while the more complex model are more nuanced and are perfectly tailored for specific applications. The different possible goals and applications also imply different measures of evaluation for the models, going from measures that indicate human interpretability of the topics, to more a statistical inspired evaluation like perplexity.

Then we saw that some work has already be done in connecting topic modeling with semantic search: Nikolaev [6] and Wu [7] propose two different approaches in their articles, from which this thesis will take inspiration. The former introduces a way to filter the semantic search results using topic modeling, while the latter incorporates topic modeling in the score value used to define the rankings of the results.

Finally we saw the ways in which the results of the semantic search can be effectively evaluated, why is the evaluation still a challenge in the semantic search field and how to select the proper evaluation for our goal.

4

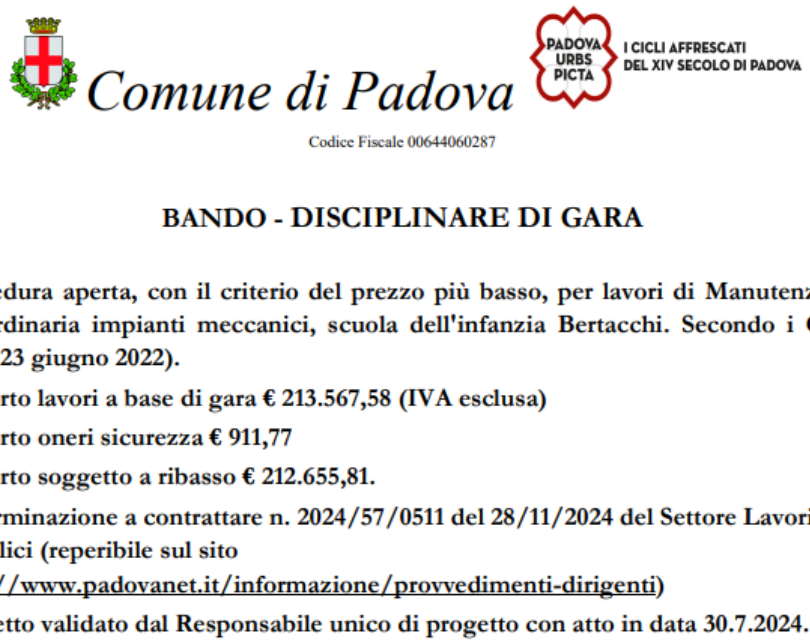
Problem Statement

The ”Amministrazione Trasparente” database contains lots of documents where most of the text repeats itself or gives bureaucratic information that is lacking actual content. It would be better to exclude these types of chunks of documents from the semantic search.

Example of a bureaucratic chunk of information: *Settore Polizia Locale e Protezione Civile RIFERIMENTI: Istanza n.2021-1511 Richiedente-DISTRIBUZIONE X VIA SAN FRANCESCO 11 25 NULLA OSTA condizione che vengano adottati gli accorgimenti per la sicurezza della circolazione mantenendoli in perfetta efficienza sia di giorno che di notte, ai sensi dell'art. 21/2° e 3° comma del N.C.d.S. a salvaguardia della pubblica incolumità. La data d'istruzione di eventuali divieti temporanei, regolarmente segnalati secondo quanto prescritto dal N.C.d.S., dovrà essere comunicata a mezzo e-mail agli indirizzi poliziamunicipale@comune.padova.it e polizialocale@pec.comune.padova.it. La data di inizio dei lavori, l'orario ed il proseguimento dello stesso dovrà essere concordate con il Comando di Polizia Locale.*

This chunk of text is taken from an authorization for a public road construction document, and it talks about general information that is present in every document of such nature. This type of texts would be considered just an obstacle to the selection of the most relevant documents from which the Virtual Assistant should generate its answer.

It's also important to remember that the whole database of documents is composed of about


Comune di Padova

PADOVA URBS PICTA

I CICLI AFFRESCATI DEL XIV SECOLO DI PADOVA

Codice Fiscale 00644060287

BANDO - DISCIPLINARE DI GARA

Procedura aperta, con il criterio del prezzo più basso, per lavori di Manutenzione straordinaria impianti meccanici, scuola dell'infanzia Bertacchi. Secondo i CAM (DM 23 giugno 2022).

Importo lavori a base di gara € 213.567,58 (IVA esclusa)

Importo oneri sicurezza € 911,77

Importo soggetto a ribasso € 212.655,81.

Determinazione a contrattare n. 2024/57/0511 del 28/11/2024 del Settore Lavori Pubblici (reperibile sul sito http://www.padovanet.it/informazione/provvedimenti-dirigenti)

Progetto validato dal Responsabile unico di progetto con atto in data 30.7.2024.

Figure 4.1: Example header of an "Amministrazione Trasparente" Document

350.000 chunks, so a system to filter the selection of appropriate documents, or make it more specific, would be of great help to the generative model in order to answer the user question in a more focused way, and could lower the possibility of "hallucinations" in the answer.

The question, here, is how to better navigate the thousands of documents, in order to be more accurate in the search and make sure that the documents used to generate an answer are relevant to the topic that the user is asking about.

5

Problem Solution

A common tool in Information retrieval, when we are dealing with text data, is Topic modeling. Topic modeling is an unsupervised technique that has the ability to identify hidden patterns and common grounds in a collection of documents, by classifying them into different topics without any type of prior knowledge. As explained by Vayansky et. al [5], there is a wide variety of possible topic modeling algorithms, that are adaptable for different goals and different types of data.

The goal of this thesis is to help the semantic search of documents to be more accurate, in order for the Virtual Assistant to generate a better answer. A topic modeling classification could help to categorize the documents and make sure that the ones returned from the search are of the maximum interest for the generative section of the IVA4PA algorithm.

The general idea is to first create "a priori" filter, to exclude documents returned from a given user search that do not match the topic of interest of the input query, and then to re-rank the results of the search by taking into account the topics of both the documents and the query.

The dimension of the "Amministrazione Trasparente" database, which contains more than 350.000 documents, makes the discussion about the computational costs of the Topic Modeling crucial. For this reason, for the purpose of this thesis, we decided to take into consideration two topic modeling algorithms: LDA (Latent Dirichlet Association) and NMF (Non-negative Matrix Factorization).

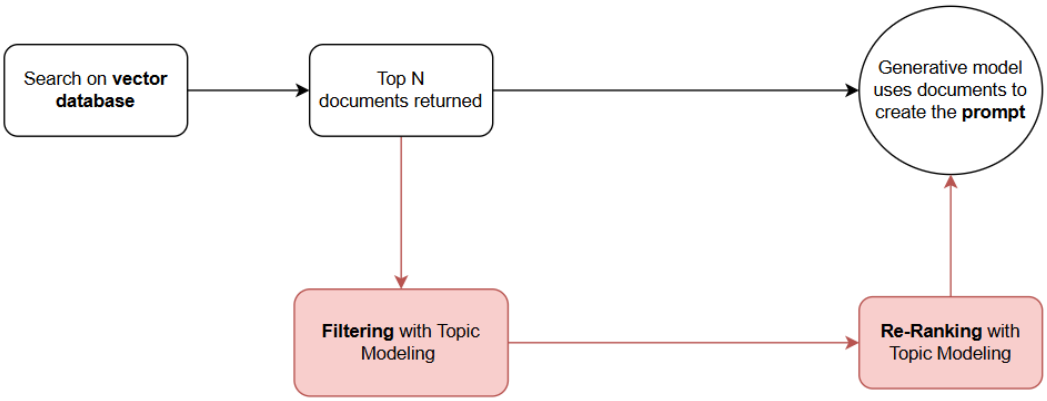


Figure 5.1: General structure of the solution (in red)

5.1 Text Pre-Processing

In order to apply a topic modeling algorithm to our database of documents, we need to create a vocabulary, containing all the words that are present in the documents' text. The creation of this vocabulary requires a careful processing of every text, in order to collect just the words carrying an actual meaning and not the ones that are uninformative and have no influence on the semantics of the text. The way that the database is built, by using scraping techniques on the "Public Administration" website, implies also a lot of text pieces that are indicative of the layout of the webpages, in the form of HTML syntax, adding an additional layer of complexity to the Pre-Processing. The whole procedure is summarized in 4 phases:

- Cleaning
- Stop word removal
- Stemming and lemmatization
- Tokenization

These phases will be described in the following sections, taking as example the text

$$\textit{Il sole splende nel cielo cittadino, ma Dicembre alle porte } < \backslash n >$$

5.1.1 Cleaning

Cleaning a text means applying this sequence of operations:

- **Lowercasing:** convert every character to lowercase, in order to ensure uniformity in the vocabulary and making sure that, for example, "November" and "november" are treated as the same word.
- **Removing punctuation and special characters:** remove every form of text that is not an integrating part of a word, such as ".,(€£/"
- **Removing numbers:** while numbers have the ability to carry potentially useful information, they have to be tied to a context in order to represent that information. Adding them to the vocabulary would introduce unnecessary randomness in the algorithm.
- **Removing HTML tags:** remove any word that is included between tags (<>) After the cleaning procedure, our example query will look like this:

il sole splende nel cielo cittadino ma dicembre alle porte

5.1.2 Stop Word Removal

Words that do not carry semantic meaning, such as articles or propositions, are called stopwords. By making use of the nlkt Python library, we can access a database of all the stopwords of a given language, in our case Italian, and remove them from the texts. The example sentence now would look like this:

sole splende cielo cittadino dicembre porte

5.1.3 Stemming and Lemmatization

The purpose of stemming and lemmatization is to incorporate together words that posses the same common root, and therefore share a similar meaning. This is done by reverting each word to its original root, or the same tense in case of verbs. To this end, we used the nlp function of the spacy library to achieve the desired result.

sole splendere cielo citta dicembre essere porta

5.1.4 Tokenization

The final pre-processing step is just splitting the text in tokens, where each token is usually equivalent to a word, to create the appropriate data unit that will structure the vocabulary of the Topic Modeling algorithm.

5.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is the most common algorithm that falls in the category of probabilistic topic modeling. It assumes that each document is represented by a vector in a "bag of words" model. This vector, in the assumptions, is generated word after word by sampling a topic from the distribution θ_i of topics of the document, and then sampling a word w from the distribution φ_i of words of the topic, where i is a document and z_i its topic assignment. The distribution θ_i is drawn from a Dirichlet distribution where each topic is independent from the others, hence the name of the model. This process defines a joint probability distribution over the training set documents and the topic structure:

$$p(\theta, z, w|\alpha, \beta)$$

where α and β are the Dirichlet priors and take the role, in this instance, of model hyperparameters.

Based on these assumption, the idea of the algorithm is that, given the observed words w, we can try to inference the posterior distribution over the variables (z, θ, φ):

$$P(z, \theta, \varphi \mid w, \alpha, \beta) = \frac{P(w, z, \theta, \varphi \mid \alpha, \beta)}{P(w \mid \alpha, \beta)}$$

This distribution, given the number of all possible configurations, is impossible to calculate exactly, so approximation techniques need to be used. The most popular ones are Gibbs Sampling, that iteratively samples the topic assignments for each word from its conditional distribution to the other parameters, and Variational Inference, that approximates the posterior distribution to a simpler one by minimizing the Kullback Leibler divergence.

While being one of the simpler topic modeling algorithms and ideal for large datasets that require topic interpretation, as mentioned by Vayansky et al. [5], in the task that we are dealing with it presents a couple of problems. The computation of an approximation of the posterior distribution for each topic, when dealing with 350.000 documents, can be computationally

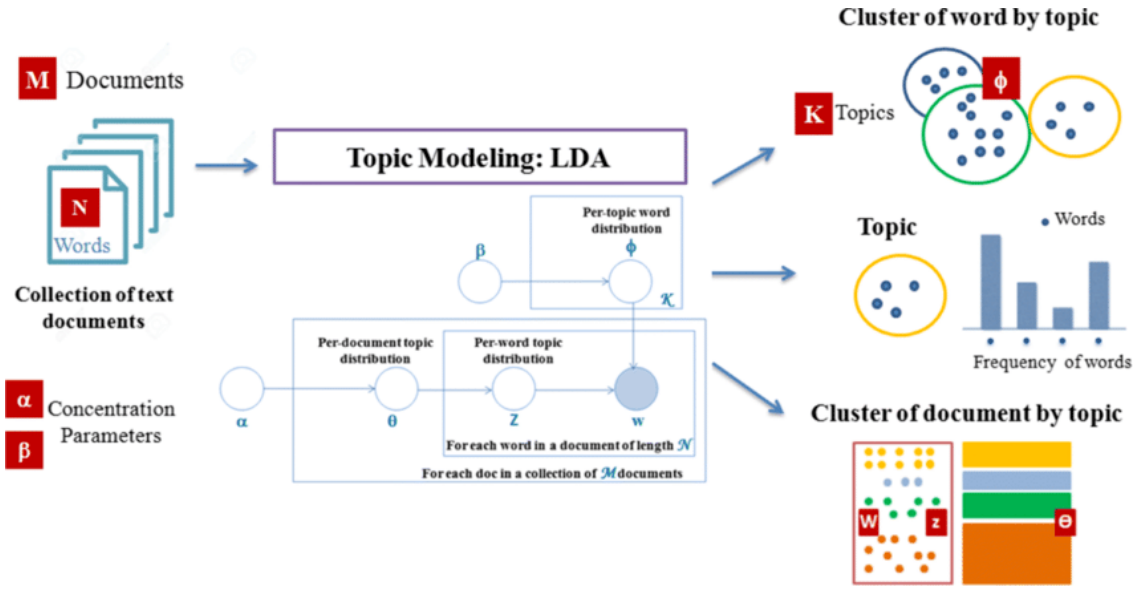


Figure 5.2: LDA Algorithm Structure [1]

very expensive. Also, the presence of the hyperparameters α and β, in addition of the hyperparameter n (number of topics), increases significantly the difficulty in the search and selection of the optimal hyperparameters.

5.3 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a Topic Modeling algorithm that belongs to a category of simpler models, the algebraic-type models. Unlike probabilistic models, NMF decomposes a Document-Term Matrix V, which represents a collection of D documents and W terms, into two lower dimensional non-negative matrices.

The first matrix H, represents the association between the documents and the topics, while the second, W, represents the association between topics and words. The main idea of this model is to minimize the error of the product between these two matrices, with respect to the original document-term matrix.

$$V \approx WH$$

The minimized quantity is the Frobenius norm of the difference between V and WH, which is the sum of all the squared differences of the elements in the matrices.

This minimization ensures that both W and H are Non-negative, and this is the key element that makes the factorization interpretable, because topics and document-topic distributions are constrained to have positive weights.

$$\min_{W,H} \|V - WH\|_F^2$$

While not computing any probability distribution over topics or words, NMF still assigns weights to each word, in order to measure the association between words and topics, or topics and documents, so it still has the capacity to help in this thesis' task. As shown by Abdelrazek et al. [4], for large datasets NMF is computationally less expensive than LDA, while performing in a similar fashion if topic interpretability is one of the goals of the model.

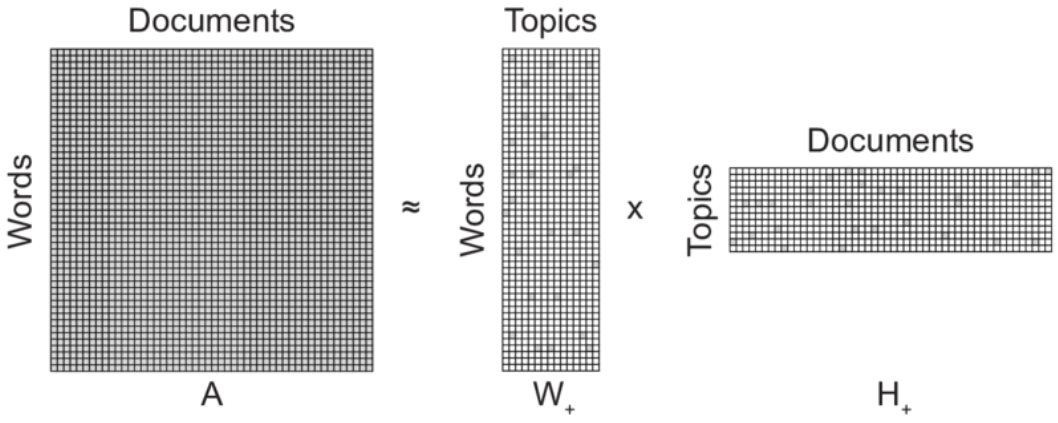


Figure 5.3: NMF on topic modeling [2]

5.3.1 Topic Coherence

The computational costs and the hyperparameters tuning, in addition to the prior mentioned factors, have led to the choice of NMF for this task, with n = 25 number of topics.

The measure used to evaluate this number is topic coherence, one of the most common indicators for the interpretability of the topics, which is key in our goal of filtering and re-ranking documents based on their semantic adherence to the query. Coherence idea is to analyse the similarity between the top keywords of each topic. A higher similarity between same topic keywords indicates a high coherence, implying a strong interpretability of the topics. The metric

used to calculate topic coherence, in our case, is the Normalized Pointwise Mutual Information (NPMI). By defining:

- w_i: the i-th ranked word in the topic's top-N list of words.
- $p(w_i)$: the probability of observing word w_i in the entire corpus.
- $p(w_i, w_j)$: the joint probability of observing both words w_i and w_j in the same document.

We can calculate the PMI [4] between two words w_i and w_j as:

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

The normalization ensures that all the values fall in a range $[-1, 1]$:

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log p(w_i, w_j)}$$

5.3.2 Model Choice

The Coherence of a topic is now defined as the average NPMI between all the pairs of top N keywords of a topic. For a topic t with N keywords, if $\{w_1, w_2, \ldots, w_N\}$ represent the keywords, then the coherence of t, $C(t)$ is equal to:

$$C(t) = \frac{1}{\binom{N}{2}} \sum_{1 \le i < j \le N} \text{NPMI}(w_i, w_j)$$

Now we can finally calculate the coherence for different values of N, where N is the number of topics. The plot in Fig. 5.4 shows how the maximum coherence value is reached in the NMF topic model with 25 topics.

5.3.3 Topics

Now that we have chosen the model, we can take a peak on what the topics we found look like. In order to represent a topic, we can list the words with a greater weight inside that topic, the keywords. This section will list a couple of topic examples next to an attempt at their interpretation.

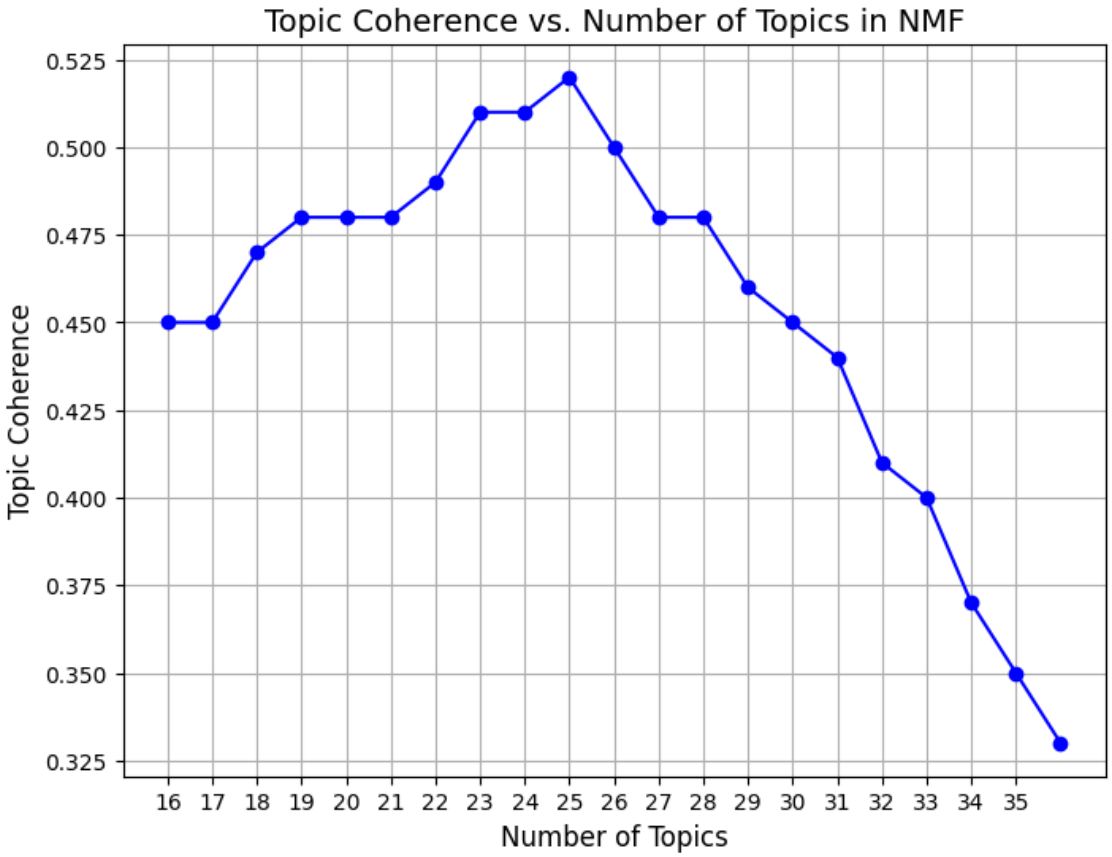


Figure 5.4: The Filtering stage

$$\text{Topic } 1 = [0.071 \times \textbf{”comma”} + 0.041 \times \textbf{”senso”} + 0.038 \times \textbf{”oggetto”} + 0.035 \times \textbf{”dlgs”} + ...]$$

We can interpret this topic as a theme of all the documents that contain bureaucratic information, since terms like ”comma” and ”dlgs” explicitly refer to Italian law syntax, while ”senso” and ”oggetto” are probably referring to ”ai sensi del” and ”oggetto di” respectively, again implying some text referring to Italian Law

$$\text{Topic } 5 = [0.131 \times \textbf{”comunale”} + 0.090 \times \textbf{”esercizio”} + 0.049 \times \textbf{”euro”} + 0.044 \times \textbf{”bilancio”} + ...]$$

This topic, instead, we can assume it’s including documents of an economic nature, proba-

bly monitoring expenses (”euro”) of the municipality (”comune”) or some other commercial activity (”esercizio”) that have an impact in the respective financial report (”bilancio”).

Topic 19 = [0.050×**”acqua”**+0.041×**”foro”**+0.030×**”realizzare”**+0.013×**”bituminoso”**+...]

In this example we can assume that the topic 19 is correlated with documents that are talking about procedures for some type of roadwork, probably authorizations for construction companies. This can be deduced by technical terms like ”bituminoso” and ”foro” that indicate characteristics of the asphalt or work procedures.

Topic 23 = [0.085×**”servizio”**+0.047×**”sociale”**+0.025×**”società”**+0.024×**”assistenza”**+...]

In the last example it’s clear how the topic includes a knowledge domain about social services and social assistance (”servizio”, ”sociale”, ”assistenza”).

5.4 Filtering Search Results with Topic Modeling

In this section we apply a filtering to the document results given by the IVA4PA searching algorithm, with the intention of eliminating the results that don’t belong to the same semantic field as the query.

5.4.1 Keywords

Now that we trained our topic model and expanded the query, we can proceed to the main idea, which, similar as in the work done by Nikolaev et al. [6], is to use the Topics of the NMF model to bridge from the documents to the query. Each document has its own topic distribution, and each topic has its own keywords. This means that, if we look at the most fitting topics for each document, we can assign a list of words representing it, those words being the keywords of the topics.

Now we ask ourselves the question: is any of those keywords correlated to the query submitted by the user? Ideally the keywords should represent the topics of interest of each document, so if none of those keywords is correlated with the query, we make the decision of filtering the document out of the results.

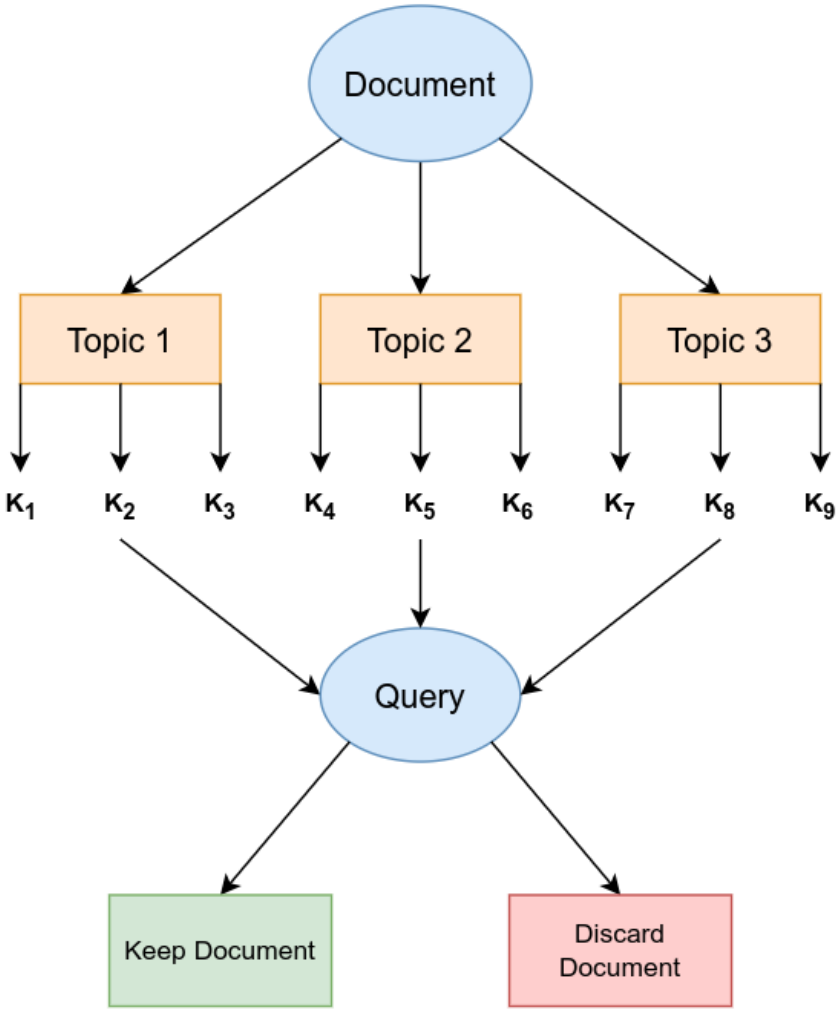


Figure 5.5: The Filtering stage

5.4.2 Query Expansion

Query expansion is a technique used in information retrieval to enhance search performance by reformulating a user's query to include additional, related terms. This ensures that more relevant results are retrieved. Before the filtering phase, where we want to measure the correlation between the keywords and the query, we decided to apply query expansion in order to make sure to capture any possible correlation to the topics, even with terms that are similar, but not identical, to the ones present in the query. For this purpose we used WordNet [9], a large lexical database that groups words into sets of synonyms called synsets, which are interlinked

by various semantic and lexical relationships. The original WordNet is tailored for the English language, so for the thesis we had to rely on a more recent related project, Open Multilingual WordNet (OMW) [10]. OMW links Wordnet to various languages, including Italian, in a free and open source manner.

5.4.3 Filtering

Nikolaev et al. [6] rely on a so called "expert" to judge the correlation between query and keywords. In this thesis we assign the "expert" role to a Large Language Model, gpt-4o-mini, for practical reason. For every search result, we submit the following prompt to the LLM:

Given this query: ***query*** *and this list of keywords:* ***keywords****, tell me if there is any keyword that is semantically correlated with the query (on a scale from 1 to 100, more than* ***alpha****). Begin your answer with either "Yes" or "No", then you can explain further.*

If the answer we get from the model is "No", then the document will be judged not relevant for the query and discarded from the semantic search. The parameter alpha is introduced to balance the severity of the filtering, since an extreme behaviour from the LLM could leave us either without any results, thus invalidating the search, or keep every result found, thus not having an impact at all in the document selection.

5.4.4 Potential practical issues

There is a consideration to be made when applying this approach to obtain a reliable filter of the documents. In the perspective to implement this algorithm in the IVA4PA project, it would be highly inefficient to interrogate a LLM multiple times, each time an user poses a question. To delegate all the semantic search phase of the project to an external LLM is not a practical solution and it's is out of the realm of possibilities, both from a computational and a financial point of view. A feasible alternative, in case of implementation, would be to use some method to calculate word similarity values, between the keywords of the query and the keywords of the topics associated with each document. If one of the similarities is above a certain value, a match is found and the document is able to pass the filter. At the moment there are plenty of libraries and embedding methods that allow to calculate word similarities, like WordNet, Spacy, etc.

5.5 Ranking Search Results with Topic Modeling

After the results filtering, it's time to calculate a new ranking that incorporates the topic model in its formulation. The baseline idea, here, is to determine the similarity between a document and a query, by taking into account how much information they share in the domain of the topics.

Every topic T_i is associated to a distribution in the vocabulary of words. We can take the 10 words (keywords), for each topic, that have the highest probability of being associated with the topic, and build a vector of keywords $[k_{i1}, k_{i2}, ..., k_{i10}]$. By using the spacy nlp embedding model, we calculate the vector representation for each of the 10 keywords, $[w_{i1}, w_{i2}, ..., w_{i10}]$. The average of these 10 vector, weighted by the strength of the association between the keyword and the topic, returns a vector which represents the topic, the **topic semantic vector** v_{T_i}.

$$v_{T_i} = \frac{\sum_{j=1}^{10} w_{ij} p_{ij}}{\sum_{j=1}^{10} p_{ij}}$$

Now, in a similar fashion, we apply this process to the query. After pre-processing the text of the query (see 5.1) and retrieving its most semantically relevant words, we perform the embedding on each one of the words, using the same embedding model as before, and by averaging the resulting vectors we obtain a single vector, v_q, the **query semantic vector**. At this point we can calculate the cosine similarities between the query vector and all the topic vectors, so for each topic T_i, $s_{qi} = v_{T_i} \cdot v_q / \|v_{T_i}\| \|v_q\|$. After all the similarities are calculated, we can store them into the **query-topic probability vector** $p_q = [s_{q1}, s_{q2}, ..., s_{q10}]$. This vector represents the probabilities of each topic to be a relevant topic for the user query. Now the job is almost done, since the output of the NMF algorithm automatically returns, for each document, a similar object, the **document-topic probability vector** p_d, that represents the probabilities of each topic to be a relevant topic for the document. The last step is to take the documents returned by the Weaviate Searching algorithm given a test query q, and calculate a query-similarity score for each of them in the form of $Score = p_d \cdot p_q$. The whole process is summarize in the diagram at 5.6.

5.5.1 Code

All the python code used to implement every step of the problem solution can be found in this project GitHub repository [11].

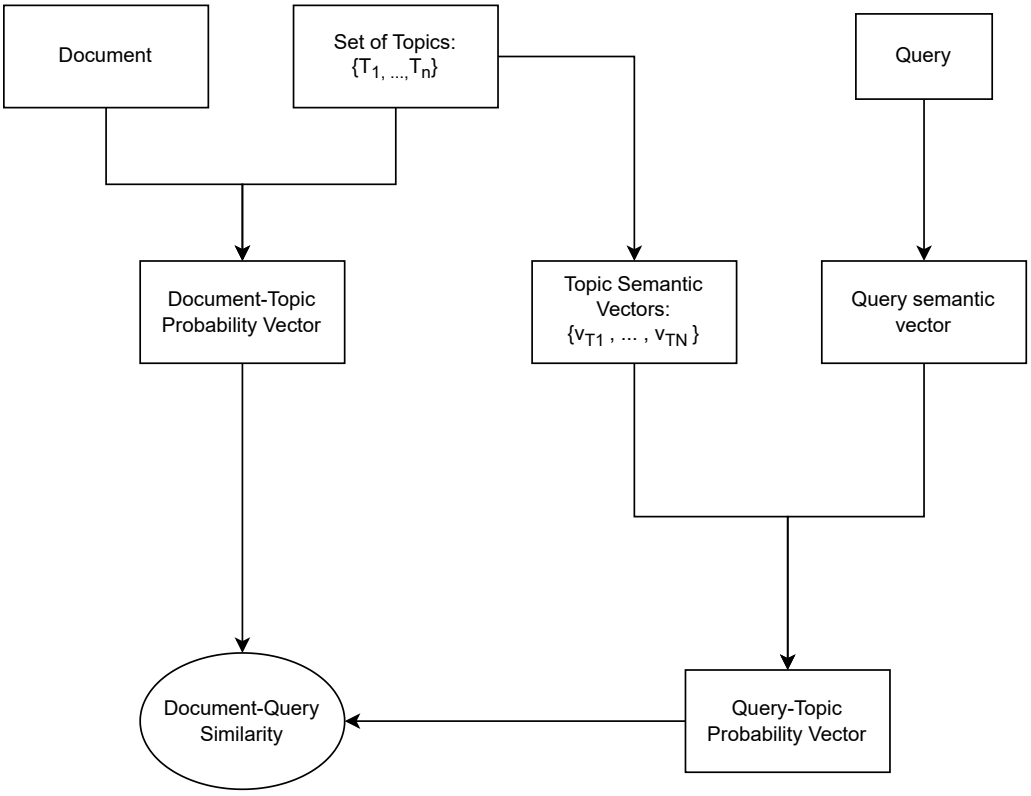


Figure 5.6: The Re-Ranking Stage

6

Evaluation

6.1 IVA4PA Semantic Search

The baseline ranking of documents, that we will compare our ranking to, is the ranking returned by the semantic search on the Weaviate database. This ranking is calculated through the LangChain Weaviate Hybrid Search Method. This is a technique that combines multiple search algorithms to improve the relevance and accuracy of search results. It's called hybrid search because it features two components, a keyword-based search component and a semantic one. When executing the search, the parameter *alpha* regulates the 2 components. If *alpha* is equal to 1, the search will be totally based on the semantic of the query and documents, while an alpha closer to 0 implies a search that relies on matching the keywords of the query and the documents. The alpha value set for the baseline search is 0.7, which was found to return the best results for the generative model to formulate an answer.

In Tables 6.1 and 6.2 we can see how the filtering and re-ranking affect the original semantic search. The returned ranking of documents is completely changed with respect to the original one, and the filtering is indicated by the score values of 0 in the second table, that indicate the documents that were excluded from the re-ranking.

Document ID	Score of the IVA4PA Semantic Search
1	1.000
2	0.350
3	0.327
4	0.320
5	0.283
6	0.283
7	0.233
8	0.212
9	0.200
10	0.194

Table 6.1: Example output of the original semantic search

6.2 Normalized Discounted Cumulative Gain

Now that the optimal ranking of documents corresponding to a given query is established, the question that remains is: how do we prove whether the Topic Modeling implementation improved or not the semantic search?

The task, here, is to confront the original ranking of documents, given by the Weaviate Score, with the new obtained ranking, that is derived by the filtering of documents and the similarity score calculated in the problem solution. One of the most frequently used measures, in the field of information retrieval, is the NDCG (Normalized Discounted Cumulative Gain), that measures the quality of a documents ranking algorithm against a single query. The requirements to calculate the NDGC are:

- A query
- A ranked list of documents returned by a search algorithm
- A ground truth ideal ranking, that resembles the optimal ranking of the documents given the query

The first step is to calculate the DCG of a search result, which is defined as:

$$DCG_p = \sum_{i=1}^{p} \frac{R_i}{\log_2(i+1)}$$

Document ID	Score after filtering and re-ranking	New Ranking
1	0.338	3
2	0.000	9
3	0.000	10
4	0.302	5
5	0.229	6
6	0.229	7
7	0.434	1
8	0.000	8
9	0.331	4
10	0.394	2

Table 6.2: Example output after filtering and re-ranking

Where i is the ranking of each of the p retrieved documents, R_i is the relevance value of the document ranked in position i by the search algorithm, which is assigned by the ground truth ranking.

This quantity indicates how strongly, given a query, the search algorithm has the ability to retrieve relevant documents, by assigning a greater score to the relevant documents in high ranked positions. It's important, although, to note that different search algorithms can return a different number of search results, therefore the DCG alone cannot be utilized to make a comparison between the algorithms. Normalizing the DCG, thus obtaining the NDCG, is the solution to this problem. It's possible to normalize a DCG value by dividing it for the DCG value of the ideal ranking established by a ground truth.

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

This way, no matter the value of p, the NDCG will always fall in a $[0, 1]$ range, making any comparison possible.

6.3 Ground Truth and Relevance Values

We built an algorithm and found an evaluation measure, now the only thing missing before evaluating the algorithm is the ground truth (ideal) ranking. This, of course, is a subjective procedure, since the same search results of an algorithm given a user query could be excellent

for one user but useless for another. The standard procedure to establish a ground truth has always been human-based, where volunteers were given some test query and the respective search result, and asked to perform a ranking based on relevance. Nowadays, with the advent of Large Language Models, another alternative is available. For the purpose of this thesis, we used the LLM GPT-4o-mini, with a detailed and customized prompt, in order to associate relevance values of the documents to each test query.

INPUT: *Given this list of 10 enumerated Italian texts in the form (index, text):* ***documents****, and this query:* ***query****, assign to all the 10 texts a relevance score representing their relevance to the query (0 = Not Relevant, 1 = Partially Relevant, 2 = Relevant, 3 = Highly Relevant). Start your answer with a list of the 10 relevance scores of the corresponding texts.*

As it's possible to see in the prompt, we opted for a discrete range of relevance values for each document, going from 0 (Not relevant at all) to 3 (Highly Relevant).

OUTPUT: *2,2,1,0,3,2,1,1,0,3*

6.4 Results

The algorithm and its performances against the original semantic search were tested on a set of 20 queries, involving plausible questions that an Italian citizen could ask to a Public administration Virtual assistant. These queries were submitted 5 times to ensure statistical stability over possible non-deterministic factors in the algorithm that are due to the use of Large Language Models.

6.5 Results Discussion

As we can see from Table 6.3, on average the semantic search returns an higher NDCG average value after the filtering and the re-ranking, meaning that the final ranked list of documents is closer to the ideal ranking produced by GPT. Since these averages are calculated over 5 runs on the 20 test queries, we can calculate, by using a simple t-test, that the difference of means is statistically significant with a p-value < 0.01. An interesting fact that can be foreseen from the values, and it's confirmed by the differences in standard deviations, is that, while the original NDCG remains pretty stable, there are some queries where the application of topic modeling actually worsens significantly the performance of the semantic search. Those are isolated cases that we assume are due to the incapability of the topic model to incorporate certain knowledge domains. If some knowledge domain is not included in the calculated topics, the filtering phase

will tend to exclude the majority of the documents from the original search, and the similarities calculated in the re-ranking phase will reflect aspects of the query that are not the ones intended by the user.

If this hypothesis is correct, however, it is an indication that the whole process could be improved even further, maybe implementing a more comprehensive topic modeling algorithm, if we dispose of more computational power. Alternatively it could be possible to train the model on a different dataset, more balanced with regard to the knowledge domains and possibly built ”ad hoc” for the task of topic modeling.

Query	Original NDCG	NDCG after Filtering and Re-Ranking
Dove posso trovare delle strutture sportive?	0.87	**0.91**
Quali sono le condizioni per ricevere i sussidi di disoccupazione?	**0.78**	0.57
Che documenti sono necessari per rinnovare la patente di guida?	0.88	**0.92**
Dove posso consultare il mio certificato di nascita?	**0.54**	**0.54**
Come posso segnalare una necessità di manutenzione stradale nel mio quartiere?	0.80	**0.99**
Quali sono i requisiti per accedere ai servizi sanitari pubblici?	**0.90**	0.82
Come posso effettuare un ricorso per la valutazione catastale sulla mia proprietà?	**0.85**	0.72
Quali sono i requisiti per effettuare una domanda di prestito per piccole imprese?	0.80	**0.82**
Quali sono le normative per avviare un'organizzazione senza scopo di lucro?	**0.83**	0.71
Dove si trovano a Padova gli uffici per la registrazione del voto?	**0.80**	0.67
Come posso segnalare un problema del servizio raccolta rifiuti?	0.85	**0.96**
Come accedere ai servizi digitali INPS?	0.84	**0.87**
Quali sono i requisiti per ottenere il bonus ristrutturazione edilizia?	0.66	**0.76**
Qual è la procedura per registrare un contratto di locazione?	0.69	**0.81**
Come posso iscrivere mio figlio alla scuola primaria?	**0.81**	0.59
Come faccio a presentare domanda per la pensione di vecchiaia?	0.86	**0.94**
Dove posso richiedere un certificato di residenza online?	0.72	**0.81**
Come richiedere un permesso di occupazione del suolo pubblico?	0.86	**0.95**
Dove posso trovare informazioni su corsi di formazione professionale?	0.93	**0.99**
Average NDCG on 5 runs	0.77	**0.82**
St.Dev. of NDCG in 5 runs	0.08	0.10

Table 6.3: Performances on one run of test-queries

7 Conclusion

After evaluating the performances of our model, we can assume, within a certain degree of safety, that implementing topic modeling into the structure of an information retrieval system based on semantic search can improve the overall results. The classification and context separation power of topic modeling have the ability to enhance the capacity of a semantic search algorithm to correctly match the user query with the most relevant documents. The whole process took a different variety of steps to be built, and in the future could be improved in different ways by better tackling its most challenging aspects. This will be key, in some way, if the topic modeling structure will need to be implemented inside the IVA4PA algorithm. At the moment of publication (december 2024) the IVA4PA project is in stand-by, but there are possibilities that it will progress to its next phase (implementation of the Virtual Assistant) in 2025.

7.1 Challenges and Future Work

In case of actual implementation, one of the first challenges to face will be substitution of the LLMs usage in the filtering aspect. As mentioned in section 5.4.4, using GPT to check the correlation between a query and the topic keyword is not affordable on a vaste scale. The major improvements, however, could be made in the topic model selection. NMF is one of the simpler existing topic models, but it was still chosen for its interpretability and low computational cost. With more computational power at disposal, it could be interesting to train more

advanced topic models on the ”Amministrazione Trasparente” database, as long as they still provide a sufficient topic interpretability.

References

[1] A. Amara, M. A. Hadj Taieb, and M. Ben Aouicha, "Multilingual topic modeling for tracking covid-19 trends based on facebook data analysis," *Applied Intelligence*, vol. 51, no. 5, pp. 3052–3073, May 2021. [Online]. Available: https://doi.org/10.1007/s10489-020-02033-3

[2] S. Ortega, https://pub.towardsai.net/topic-modeling-with-nmf-for-user-reviews-classification-65913d0b last access 2024/11/30.

[3] SBert.net, https://www.sbert.net/docs/sentence_transformer/pretrained_models.html, last access 2024/11/05.

[4] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat, and A. Hassan, "Topic modeling algorithms and applications: A survey," *Information Systems*, vol. 112, p. 102131, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437922001090

[5] I. Vayansky and S. A. Kumar, "A review of topic modeling methods," *Information Systems*, vol. 94, p. 101582, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437920300703

[6] I. Nikolaev, D. Botov, Y. Dmitrin, J. Klenin, and A. Melnikov, "Use of topic modelling for improvement of quality in the task of semantic search of educational courses," in *Proceedings of the 21st International Workshop on Computer Science and Information Technologies (CSIT 2019)*. Atlantis Press, 2019/12, pp. 104–111. [Online]. Available: https://doi.org/10.2991/csit-19.2019.18

[7] T.-H. Wu, B. Kao, F. Chan, A. S. Cheung, M. M. Cheung, G. Yuan, and Y. Chen, "Semantic search and summarization of judgments using topic modeling," ser. Frontiers in Artificial Intelligence and Applications, vol. 346, 2021, pp. 100–106.

[8] K. M. Elbedweihy, S. N. Wrigley, P. Clough, and F. Ciravegna, "An overview of semantic search evaluation initiatives," *Journal of Web Semantics*, vol. 30, pp. 82–

105, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826814001024

[9] G. A. Miller, "WordNet: A lexical database for English," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://aclanthology.org/H94-1111

[10] F. Bond, P. Vossen, J. McCrae, and C. Fellbaum, "CILI: the collaborative interlingual index," in *Proceedings of the 8th Global WordNet Conference (GWC)*, C. Fellbaum, P. Vossen, V. B. Mititelu, and C. Forascu, Eds. Bucharest, Romania: Global Wordnet Association, 27–30 Jan. 2016, pp. 50–57. [Online]. Available: https://aclanthology.org/2016.gwc-1.9

[11] G. Zerbo, https://github.com/GioZeta99/iva4patm.

Acknowledgments

First of all, I would like to express my most sincere gratitude to my supervisor, Prof. Roberto Confalonieri, for the consistent support and time he dedicated to my thesis.
I'm also extremely grateful to the Akera S.R.L. employees that worked on the IVA4PA project, first and foremost Dr. Luca Peruzzo, together with Luigi Bellio and Samuel Scarabottolo. This wouldn't have been possible without you.
Special thanks to Ing. Pietro Fontolan and Ing. Alberto Corò, for giving me the opportunity to be part of the internship.
Lastly I would like to thank my internship colleague, Dr. Josè Chacon, and all the colleagues that helped me through these last years.