Topic-based summarization to objectively analyze Central Bank statements and market sentiment

DRAFT

Carlo Abrate

Supervisor: Prof. M. Gasparini
Prof. R. Fontana

Ingegneria Matematica
Politecnico di Torino

This dissertation is submitted for the degree of
Ingegneria Matematica, Statistica e ottimizzazione su dati e reti.

July 2018

Abstract

The Riksbank (Sweden's Central Bank) releases the minutes of the Monetary Policy Meetings every two months. SEB Bank, as most of the players in the market, analyzes and creates reports about the opinion of each board member of Riksbank, to know how market changes. The goal of this work was to create automatic tools to help the Research Team of SEB. Two text mining techniques have been used: Sentiment analysis, to uncover the position (Hawkish or Dovish) of each Board Member, and Summarization, to analyze the most important statements in the minutes. In particular, a human-based topic summarization algorithm is used to summarize beliefs for each board member on different topics. Moreover, an automatic topic summarization algorithm based on Latent Semantic Analysis is proposed. The topics retrieved by the Topic Model used are similar to the ones proposed by humans. Summary evaluation has been based on human judgment: if enough sentences are retrieved, most of them are relevant, but some key points could lack. Author-based and time-dependent topic models could be a good improvement of this work.

Acknowledgements

And I would like to acknowledge ...

A mio nonno, per avermi insegnato a sognare.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Skandinaviska Enskilda Banken AB (SEB) is a Swedish financial institution with headquarters in Stockholm. In the investment division of the Bank, a lot of information are analyzed daily and report are created for clients and internal teams. Riksbank, the Central Bank of Sweden, following the examples of European Central Bank (ECD), Federal Reserve (FED) and several others central banks, has adopted a more transparency approach in the last 15 years. [18]

Every two months Riksbank publishes the minutes of the meeting attended by its Board Members. A lot of information are published. Since 2007 SEB have analyzed the position of each Board Member in the last meting. The examination of the minutes are given out on SEB's Website few hours later the minutes' publication.

The aim of our project is to automatically create a report with information of the minutes. By SEB's reports, the position of the Board Members is presented to the reader by a summary of the main concepts debated. Furthermore, the Board Members are ranked from the most Dovish to the most Hawkish.

The problem has been translated into the field of text analysis. After a review of the techniques to examine texts, two main methods have been chosen: Text Summarization and Sentiment Analysis. In order to synthesize the opinion and analyze the topics of the speech of each board member different algorithms have been proved. A weighted formula is calculated to give a relevance score to each sentence and then extract the most important. In section 3.4 is explained our approach to Text Summarization.

Sentiment Analysis techniques are frequently used to analyze reviews of products and give out a positive or negative classification of that. We expanded this idea to give an Hawkish/Dovish score with the help of the last machine learning algorithms. In section 3.5 is reported our strategies to create this sentiment score.

Both to define the position of the Board Members and summarize their key statements, a topics analysis is helpful. The summarization algorithm provided to the Research team of SEB is based on some topics that are given ex-ante by the users. In this case, the topics analysis is made by humans and the contextual knowledge is used to create the topics used by the algorithm.

A more challenging approach, it is to mbase the topic analysis

In section 3.1 are explained the different issues that have been faced to pass from the original PDF text to a more structured form of data.

1.1 Problem Formulation

1.2 Economical grounds

1.2.1 Monetary policy

The Central bank of every country is responsible for its monetary policy, in Sweden this is a task of the Riksbank. The objective for monetary policy is "to maintain price stability, that is keep the inflation close to the target of 2% per year.[1] Broadly speaking, there are two types of monetary policy, expansionary and contractionary. The first one is adopted with the intention to encourage economic growth and expand the money supply; instead the second seeks to obtain the opposite result. They are a reaction to different types of economic situation that a country faces.

Central banks have three main tools to guide the inflation: set the Repo Rate, buy or sell treasury bonds through open-market operations, and establish reserve requirements [9]. Repo Rate is used as benchmark by the others banks to set them lending rates, with a low lending rate people are induced to borrow more money and the economy is stimulated. Central bank can buy a large amount of treasury bonds to decrease the market liquidity, the prices will rise and the yields will drop. Clients will shift to other assets with higher returns and the economy will expand, thanks to the diversification of the investments. Furthermore, Riksbank can decide with percentage of costumers deposit banks must keep, the level they reserves affects the short-term interest rate banks pay to borrow and lend money from and to each other. This influence also the interest rates banks charge consumers for borrow money.

Decisions about the monetary policy are taken by the Executive Board, composed of six members that, as politicians, can have different opinions about which position to take. Monetary policy effects are not easy to measure and there is not a unique way to decide when

[1] https://www.riksbank.se/en-gb/monetary-policy/

is time to change the trend. Since the 2008 financial crisis, the monetary policy of almost each country is expansionary to stimulate the economy.

In Sweden, the current repo rate is below zero and the Riksbank has a heavy presence in bond markets. The statements of the Board members said that the strategy will remain constant until the end of the year.[2]

The sentiment of the central bank monetary policy can be describe as **hawkish** or **dovish**. The hawkish word has the same hawk root and it is a clear reference to the fact that the hawk flies as high as the Repo Rate in a hawkish position. Board members are in a hawkish position when they support a contractionary monetary policy and dovish when they are in favor of a expansionary monetary policy. The border between the two definitions is not defined and sometime is not easy understand in which position is a given strategy. [3]

1.3 Text Mining

1.4 AI in Economics

1.5 Similar Works

[2] https://www.ft.com/content/8998e16c-b15c-355c-8512-32d4b6b4b2f8

[3] //www.tradingheroes.com/hawkish-and-dovish/

Chapter 2

Dataset

In this section are described the data available. All the original data are of text format, so quite unstructured. Therefore, for each source of information available it is described both the data transformations applied, to give a more structured organization and support the examination, and the final dataset.

The first report is called Minutes, it is the statement by Riksbank of the meetings between the Board Members. The text is transformed from PDF to HTML format to enriched the text with HTML tags. Subsequently, the latter are used to extract features and create a first dataset, called Minutes Dataset.

The second text is called SEB Report, it is released by SEB few hours later the publication of the Minutes. It is a more structured table with six raw and three columns. Raws are one for each Board Member, and for each of them is reported: name-surname, summary of their argumentations, and hawkish/dovish rank. Because no future automatic reader is needed on this text and because of the complexity of the text structure, the text has been copied-pasted manually in a Text sheet.

2.1 Minutes

Minutes of Riskbank meeting have been analyzed to guide the economical decision related to monetary policy. Minutes are published and available to everyone since 2013 and they come out every two-three months.

Minutes are usually divided into four sections. The purpose of our analysis permits to focus only on the Board members discussion part, where are debated their opinions and thoughts for future changes in monetary policy.

In the most recent papers, board member's speeches are slightly summarized and grouped in a single and continued section. This operation simplifies the research for each board

member's text and helps the human or automatic reader to go easier through the paper. Unfortunately, in the minutes published before 2015, the structure was not so well defined and more sections of the same board member were allowed, probably to maintain the idea of a dialogue. Each board member's speech is usually from three to four pages long and it addresses several topics, some of them are recurrent, such as "Inflation", others could be specific in a meeting, such as "Oil".

The names of the members of the Executive Board are listed at the beginning of the first section because they can change over the years. The Board members are appointed for a period of 5/6 years according to a continuous program. They are six and all of them are usually present at every meeting, also because at least half of the members must be present for decisions to be made. Six years is a significant amount of time and this gives stability to dynamic analysis that could be performed on each member, see section 4 about Topic Modeling.

2.1.1 Structured data

All the minutes available on Riskbank website[1] have been analyzed to study patterns and extract information in the future ones. The collection obtained is made by around thirty Minutes. Two main information are investigated in this section:

- Board Member part of text;
- Topics in paragraph.

For the first task, HTML tags are used with a dictionary [2] containing the names of the Board Members. After a text transformation from PDF to HTML, better explained in section 3.1, it is possible to identify the bold type words in the text. Combining this tags with a dictionary with the names of the Board Members, it is extracted the part of text corresponding to a specific Board Member.

In text mining, there is not a shared language to identify the different documents and part of text; however, in section 3.2 is reported the most used. Transferring the notation on our data, this is how the data are organized:

- Collection C is the set of Minutes from different period of time, so d_i is the i-th Minutes, $C = \{d_1, d_2, \ldots, d_q\}$ with $q = 30$;

[1] www.riksbank.se

[2] In this context, dictionary/lexicon are used interchangeably to indicate a collection of words with some information available.

Minutes	Board Member	Topic	Paragraph
2016-02	Cecilia Skingsley	Inflation	307, 308, 309, 345, 346, 347, 348, 349
2016-02	Cecilia Skingsley	Krona	307, 308, 309, 322, 323, 324, 325, 340, 341, 342
2017-10	Per Jansson	ECB	180, 181, 182, 223, 224, 225, 226, 227, 228

Table 2.1 Three raw of Minutes Dataset.

- Document d is a single Minutes, it is composed of Part of Text pt, $d = \{pt_1, pt_2, \ldots, pt_k\}$ with $k = 6$ the number of Board Members;
- Part of Text pt is the part of text by a Board Member. It could be divided by topics $pt = \{top_1, \ldots, top_p\}$, or by paragraphs $pt = \{pg_1, pg_2, \ldots, pg_p\}$, or by sentences $pt = \{s_1, s_2, \ldots, s_p\}$, or by words $pt = \{w_1, w_2, \ldots, w_p\}$, it depends by the goal of the analysis applied;

To apply the second trick to get more structured data, it is first needed to divide the text in paragraph and text. This operation is made by combining HTML tag and Python Library named NLTK, see section 3.1. Given paragraphs and sentences in *.txt* format, it is possible to collect knowledge on where topics are discussed. The key words (Topics) are selected by a dictionary created with the help of the SEB's research team.

For the Topic Dictionary, four main topics were considered, which they always debate in every meeting, no matter in what year they are, that are:"inflation", "European Central Bank" (ECB), "Federal Reserve" (FED) and "Swedish Krona". This Topics are suggested by the Research team of SEB. The paragraph are labeled based on these four main topics, could be that a paragraph has more than one topics or no one inside.

The final dataset looks like the table above 2.1, it is combined with a dictionary data structure where it is saved the text of the sentences.

To summarized:

- each Minutes d in the collection of Minutes C is identified by the date, such as $2016-02$;
- each Board member part of text pt from a specific Minutes d is identified by the name of the Board Member;
- each topic top is identified by the name of the topic, "Krona" for instance.

In the end, for each Minutes, for each Board Member and for each Topic, the corresponding paragraphs are saved by the identification number of the sentences that compose it. The text of the equivalent sentence is retrieved by the dictionary of sentences.

2.2 SEB Reports

Around two weeks later than a Monetary Policy minutes has been published, the SEB bank provides a schematic summary of it, that is uploaded to its website[3]. For each member of the board, they provide a brief summary, consisting of a few lines of text revised by the board member's speech. The Board members are also classified from the most dovish to the most hawkish, based on what they said during the meeting. Each meeting is considered independently, the score is assigned without taking into account the past behavior of the board member. When the economical position of a member is not clear, or two or more of them express the same idea, they are considered at the same level.

It is important to observe that the information regarding the position of the Board Member and the summary provided are two different and independent information. This means that the sentences that are collected from the original Minutes to create the summary are not the most relevant to understand if the Board member is Hawkish or Dovish. This issue it is further discussed in the evaluation section [4].

The extraction of the text data from the SEB Reports is made by hand. The complexity of the format, a Power point or PDF table, makes almost impossible to create an automatic text reader able to extract the relevant part of text in a structured and standardized format. However, with this document it is not needed an automatic extraction of data for future implementation. So, the only turn off of the copy-paste approach is the human time spent to create the dataset.

2.2.1 Structured data

The summaries are used to evaluate the performances of the summarization algorithms. The rank, given by the SEB analysts, is used to build a train set for the classifications algorithms. In our work, the two highest in the ranking are considered as *dovish* (labeled as 0) and the two last are considered as *hawkish* (labeled as 1). The third and the fourth are classified as *neutral*. In the event of a tie between three members, all of them are considered as *neutral*.

Board member

The entire text of each board member is labeled using the score given by the SEB report. As said before, Board members speak for a long time and not all of the sentences in the speech

[3] www.seb.se

[4] add evaluation section: discuss the impossibility to evaluate the most H/D sentences by the SEB Report's summary.

are meaningful, to assign the Hawkish/Dovish score. In this case, this problem is ignored ***WITH GOOD RESULTS(?)***.The obtained dataset is composed of ***NUMBER*** instances.

Board member and topic

It is based on the second criteria of division used on the minutes. Each section related to an author and a specific topic is labeled. SEB rank is referred to the entire Board member text, so to every topic of the same author is assigns the same score, without considering changes of the tune. Using this technique the dataset appears four times bigger and it reaches *NUMBER* items.

Chapter 3

Algos

In this section are described the algorithms used to analyzed the data available. In the first part 3.1 are reported the different techniques used to transform the original text into different format and the text cleaning procedures, such as tokenization issue.

In the second part of this chapter, the two main algorithms used in this project are discussed: Summarization and Sentiment Analysis. For both the algorithms, it is first stated the problem, then strengths and weaknesses are examined to select the most fitting algorithms to our specific problem. In the end, it is explained our approach and how it has been evaluated.

3.1 Text Cleaning and Data Preparation

Text data are very unstructured source of information for IT purpose. In fact, for a human being it is usually straightforward to get the right meaning of a text, unfortunately it is not the same for a computer. Ones the text are available, the first issue to be solved in every Text Mining problem is the text cleaning and data preparation.

In this project, it is possible to divide this issue in two sub-problems:

- Technical text cleaning, named Data Preparation;
- Purpose-based text cleaning, named text cleaning.

The first identify the problem of transform the text's formats to be ready to be analyzed by algorithms. The input of the Data Preparation is the Minutes in a PDF format, moreover using dictionaries and HTML tags is created the Minutes Dataset in *.txt* format, presented in the section 2.1.

The second technique called Purpose-based Text Cleaning, it is applied to transform and clean the text contained in the Minutes and the Report dataset. In the following section are

reported several techniques available for this intent and it is analyzed when it is advantageous to use them.

3.2 Notations

There are several algorithms available to extract the most relevant sentences from a text. Before explaining deeply these algorithms, it is important to define some common concepts and the hypothesis behind this type of methods.

There are some important definitions that it is important to give in order to have a common notation in the following explanation:

- Collection C, it is the set of documents d, $C = \{d_1, d_2, \ldots, d_q\}$;
- Document d, it is a single unit of a Collection and it is composed of Part of Text pt, $d = \{pt_1, pt_2, \ldots, pt_n\}$;
- Part of Text pt, it is a single unit of a Document d, it can be a sentence $pt = s$ or a paragraph $pt = pg$, that, in turn, could be composed by sentences $pg = \{s_1, s_2, \ldots, s_p\}$.
- Sentences s are composed of words w, so that $s = \{w_1, ..., w_m\}$.

In particular, a general word w_i in a sentence s is not uniquely identified. In fact, the tokenization part is essential, that means to assign the right token from a collection of tokens (or labels) to a word in a sentence. The collection of tokens are generally named dictionaries or lexicons, and are collection of words where the specific meaning is defined for each shade of meaning of the word, moreover some more information could be collected in the dictionary.

There are many dictionaries available on-line for the purpose of tokenization and Part-of-Speech Tagging (POST).[1] The latter are used to infer the right meaning of a token usually based on the context of the word that surround the word tokenized. [2]

A good reference for this purpose are the book "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" from Prof. Daniel Jurafsky[17] and "Text Analytics with Python: A Practical

[1] Some of the most used Knowledge-based dictionaries available are: WordNet, SentiWordNet, AffectNet,GoogleNGrams, MicrosoftNGrams, NELL, FrameNet, ConceptNet, VerbNet, FreeBase, DBPedia, Probase, SGECKA, Per language resources, e.g. Cornetto.

[2] Spacy https://spacy.io/usage/linguistic-features is a powerful Python Library used to extract linguistic features like part-of-speech tags, dependency labels and named entities, customising the tokenizer and working with the rule-based matcher.

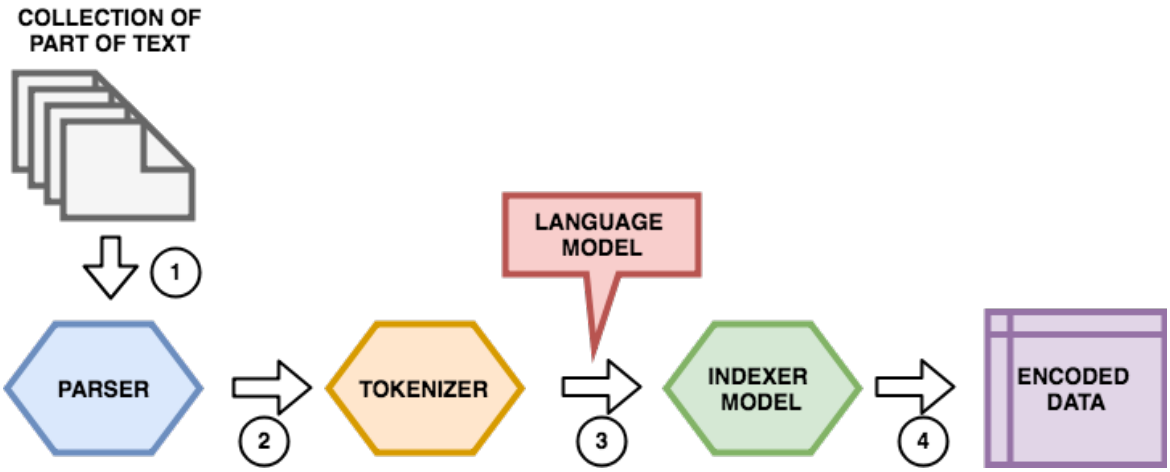

Fig. 3.1 Parsing: add a description

Real-World Approach to Gaining Actionable Insights from Your Data" from Dipanjan Sarkar. [26]

3.2.1 From PDF to Data encoded

In diagram 3.1 is explained the flow for the indexing of collection of documents. Text indexing is used to encode texts and then apply computational techniques to retrieve information.

1. Documents Parsing: documents were in pdf format, a transformation to HTML format is done. Then with Regular Expression, the information requested are extracted in the format described in the Dataset section 2. The Python library PDFMINER.Six is used in the code for this passage.

2. Tokenization: ones the text is available in a .txt format. The tokenization is applied.

3. Language Model: some language based heuristics are applied to the tokens to enrich the structure of the data. Some of the techniques available and used in the project are listed and briefly introduced: Stop-words, Lemmatization, Stemming. With a Stop-words list, it is possible to exclude from the dictionary entirely the commonest words that usually are not so discriminative in the model, such as: "the", "a", "or"... Stemming and Lemmatization allow to remove the prefixes, suffixes from a word and and change it to its "base form". It is straightforward that is a good way both the techniques above are a good way to reduce the numbers of unique terms in the collection. For a more accurate survey on these techniques a good paper available is "Preprocessing Techniques for Text Mining" from Gurusamy Vairaprakash and Kannan Subbu.[12]

4. Indexer Model: the next section introduce the most famous "Bag-of-words" Model to encode texts.

3.3 Vector Space Model

Most of the techniques in Machine Learning have a shortcoming, it is not possible to give directly unstructured information to the algorithms. In fact, features are used to have a common pattern in the informations available. With numerical data, it is quite easy to organize the data in some matrix. However, with text data it is not that trivial and some hypothesis are necessary to move from text to numbers. To this hand, there is a mandatory step after the preprocessing and the cleaning of the text, the feature extraction is useful to give a more structured dataset to the algorithms applied.

In this section, it is analyzed one of the most used feature extraction model for text data, it is called Bag-of-Words model or Vector Space Model. The first name also identifies the hypothesis behind this model: terms are considered as in a bag, the order by which the terms appear in the sentences is not relevant, only their frequencies are considered. In other words, it is a simplistic representation of a Collection $C = \{pt_1, ..., pt_n\}$, where each Part of text pt_i[3] is modeled as a vector. A dictionary DIC is used to create a space for the terms te considered, thus the vector v_{pt_i} that represents a part of text pt_i is a point in the space of the dictionary, $v_{pt_i} \in R^D$ with $D = dim(DIC)$.

The final result of the Bag-of-Words model is usually represented in a matrix M, where by rows is represented the "distribution" of a term in the Collection C and by columns is contained the representation of a Part of text pt by its terms. In other words, the matrix $M \in R^{Dxn}$ has D raw and n columns, with this in mind, raws of M give information and represent terms, meanwhile columns Part of text pt.

The Bag-of-words hypothesis is valid only if words are thought to be independent from the position where appear in the text. Therefore, it is a strong supposition and a lot of available knowledge is lose. However, from an unstructured form data are now in a matrix, where a lot of mathematical tools and theories are available to be applied on it.

As an example, let's consider the following collection of sentences $C = \{$"To be or not to be, that's the question.", "To be or not to be. That's not really a question.", "Development is about transforming the lives of people, not just transforming economies."$\}$. After the Text Cleaning step, explained in section $ADD-SECTION-LINK$, the collection is represented as $\hat{C} = \{$"to be or not to be that is the question", "to be or not to be that is not really a question", "development is about transforming the lives of people not just transforming economies"$\}$.

[3] Documents $C = \{d_1, ..., d_n\}$ or sentence $C = \{s_1, ..., s_n\}$

Starting from this cleaning collection $\hat{C} = \{s_1, s_2, s_3\}$, the first step of the Bag-of-Words model is the creation of a dictionary:

$$\begin{aligned} D = \{&\text{"}to\text{"}, \quad \text{"}be\text{"}, \quad \text{"}or\text{"}, \quad \text{"}not\text{"}, \quad \text{"}that\text{"}, \\ &\text{"}is\text{"}, \quad \text{"}the\text{"}, \quad \text{"}question\text{"}, \quad \text{"}really\text{"}, \quad \text{"}a\text{"}, \quad \text{"}development\text{"}, \\ &\text{"}about\text{"}, \quad \text{"}transforming\text{"}, \quad \text{"}lives\text{"}, \quad \text{"}of\text{"}, \\ &\text{"}people\text{"}, \quad \text{"}just\text{"}, \quad \text{"}economies\text{"}\} \end{aligned} \tag{3.1}$$

After the creation of the dictionary, it could be used an intermediate step to make a Feature Selection, in this section it is not applied and it is further described in section 3.2.1.

To get the final representation of the Collection, the measure the relevance and the discriminative power of a term has to be chosen. There are several scores as explained in section 3.3.1, in this example is counted the frequency in the single sentence. The final matrix is:

$$\mathbf{A} = \begin{array}{l} \\ \text{"to"} \\ \text{"be"} \\ \text{"or"} \\ \text{"not"} \\ \text{"that"} \\ \text{"is"} \\ \text{"the"} \\ \text{"question"} \\ \text{"really"} \\ \text{"a"} \\ \text{"development"} \\ \text{"about"} \\ \text{"transforming"} \\ \text{"lives"} \\ \text{"of"} \\ \text{"people"} \\ \text{"just"} \\ \text{"economies"} \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

From the matrix M, it is possible to get relevant information for both for sentence and terms:

- $s_3 = \{0,0,0,0,0,1,1,1,0,0,0,1,1,2,1,1,1,1,1\}$ is the final representation of the third sentence;
- "*the*" = $\{1,0,1\}$ is the final representation for the term "*the*".

There are several remarks that it is important to do on this model. The first main disadvantage is that, because of the "Bag-of-words" simplification, some important knowledge contained in the text is lost. At a sentence-level, it is lost the semantic relation between the terms in the sentence. Moreover, the logical and grammar structure of the sentence is not maintained, if it is not for Part-of-Speech (PoS) Tagging, explained in the section 3.1, where some of these structure could be take into account. At a Part of Text level, a paragraph for example, the logical and semantic information between sentences can not be considered by this model.

From a computational and memory perspective, the Bag-of-Words model has also some disadvantages. It is easy to notice that for a collection of books, for instance, it easy do have a Dictionary D with many terms and not all the terms are in common between books, consequently the final matrix M is sparse. Some cleaning techniques can improve the efficiency of this model, for instance, as explained above, with Language Models is possible to reduce the sparsity of the matrix M due to the reduction of the terms in the dictionary used D.

3.3.1 Scoring Words

The elements of the matrix M encode the information for each part of text and terms. In fact, $m_{i,j} = M(i,j)$ is the knowledge saved for the terms $tr_i \in D$ in the Part of text $s_j \in C$.[4] Having said that, the type of information saved depends on the application of the project.

Two main information are considered for each term based two level of analysis: sentence-level and collection-level. In general, terms considered in a sentence gives information about the sentence, for instance if the term "inflation" is recurrent in a sentence (or paragraph) if quite likely that the sentence is about inflation. On the other hand, a term viewed in the collection of sentences can give information about his role in the whole text considered. For instance, Stop-words are considered useless because are to frequent in the collection to be discriminative.

To summarized, different types of weighted schemes are available to chose the right score for the matrix M, some of them weighted are presented:

[4] In this explanation, the part of text are represented by sentences, to simplify the explanation.

- Boolean Model: it is a binary representation $m_{i,j}$ of the term in the sentence, if $m_{i,j} = 1$ the term $tr_i \in D$ is present in the sentence $s_j \in C$.

- Term Frequency Model: the score used to weight the presence of a term is proportional to the frequency. The length of the sentence considered can influence this type of weighting scheme, some adjustments are often made. A common measure used is, where $m_{i,j} = tf_{i,j}$ and $f_{i,j}$ is the frequency of term w_i in sentence s_j:

$$tf_{i,j} = \begin{cases} \frac{f_{i,j}}{\sum_{tr_k \in s_j} f_{k,j}} & \text{if } tf_{i,j} \neq 0 \\ tf_{i,f} & \text{if } tf_{i,j} \geq 1 \end{cases} \tag{3.2}$$

- Inverse Document Frequency:an inverse document frequency factor is incorporated in the formula to distinguish between relevant and non-relevant terms. To this hand, a common score to take care of the frequency of terms at the collection level is:

$$idf_i = log(\frac{N}{df_i}) \quad \forall te_i \in D \tag{3.3}$$

  where $N \in N$ is the number of sentences (or better Part of text) in the collection C ($N = |C|$), df_i is a measure of the informativeness of term te_i in the collection, for instance:

$$df_i = |s_j \in C | t_i \in s_j| \tag{3.4}$$

  Notice that: the highest is the df_i measure, the less discriminative is term te_i

- Tf.idf Model: it puts together the term-frequency formula with the Inverse Document Frequency one. Finally, the resulting formula is:

$$tf.idf_{i,j} = tf_{i,j} * idf_{i,j} \tag{3.5}$$

A good reference used to compare the results of with the different types of score is "Introduction to Information Retrieval" by Christopher D. Manning.

3.4 Summarization

Information overload... Intro about summarization in Text mining: - abstractive Vs extractive - Constrain-based - Context-based Vs "Bag-of-words"

3.4.1 Extraction-Based Summarization Algorithms

Extraction-based techniques to create a summary are currently the most used summarization techniques, due to a simplification hypothesis behind: there is no need to create new part of text, the resulting summary is a subset of the original text. [3] In this way, it is not necessary to involve Natural Language Generation Techniques to rewrite the main contents, instead the summary is composed by portions (sentences, paragraphs) of the original text selected by some criteria.

Following this logic, it is needed to choose a principle to select the portions of text to create the final summary. There are many models and in the following section it is explained our approach, but first a more abstract formulation of the summarization model is given.

3.4.2 Mathematical formulation of the summarization problem

There are two main point/criteria to be choose in this model:

- How to select a "relevance" score for each sentence;
- How to extract the sentences by the score and its information.

The original text T could be represented as a collection of portions of text. In this model, sentences are the smallest part of text selected for the summary: $T = \{s_1, \ldots, s_i, \ldots, s_n\}$.

The summarization algorithm, chosen the function SS, associates a score for each sentence:

$$\begin{aligned} SS: T &\to \mathbb{R} \\ \forall s_i \in T, \quad SS(s_i) &= score_i \in \mathbb{R} \end{aligned} \tag{3.6}$$

The scores are collected in a vector $SS = \{score_1, \ldots score_n\}$. It is possible to think about this first phase as a features extraction step. In fact, from unstructured data, such as a sentence, it is obtained a score, therefore a structured feature that describe original data.

The second step of the summarization algorithm starts from the vector SS and, by the extraction criteria EC, it is obtain the summary S:

$$\begin{aligned} EC: SS \cup I &\to S \\ where \quad SS, I \in \mathbb{R}^n \quad and \quad S \subset T, \quad |T| &= k. \end{aligned} \tag{3.7}$$

Where k could be choose by some heuristics and the external information are represented by I.

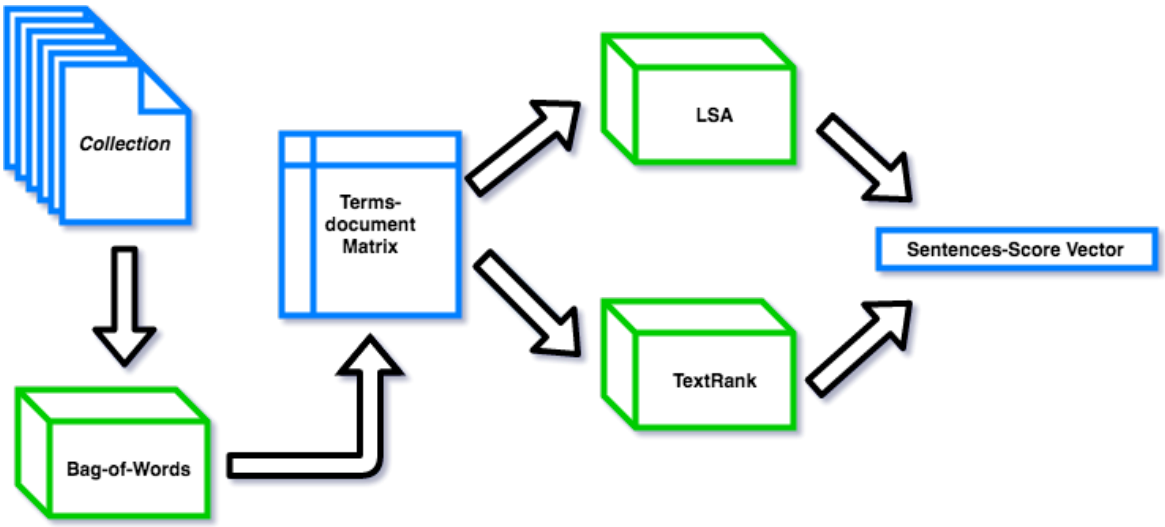

Fig. 3.2 Summary: sentences-score by LSA and TextRank.

3.4.3 Latent Semantic Analysis

As it is explained in the paragraph/section 3.4.2, the first important question to be solved in Extraction-based text summarization algorithms is to score each sentence by the "relevance" in the text. It is not possible to have a general definition for "relevance", because it depends on many factors:

- Contextual and subjects: the relevance algo/heuristic valid in Biology field could be different from economics context.
- Goal of summarization: a text corresponds to many possible summaries. In fact, it depends by what kind of information have to be collected in the summary.

In this section, it is explained one of the most used extractive-based summarization algorithm Latent Semantic Analysis (LSA) and in the next one TextRank.

LSA is used in NLP context under the assumption of Distributional Semantics Theory, a sub-field of Linguistic. The main idea is that items that have a similar distribution in the collection have similar meanings, by items is intended different pieces of the collection, as terms, sentences and paragraphs.[13] To understand what a distribution of a linguistic items is, it easy to use the Vector Space Model, section **??**, by which is possible to obtain a distribution of the words in the collection and a distribution of the Part of text desired.

The LSA Algorithm is frequently applied in the context of Information Retrieval with the name Latent Semantic Indexing (LSI). The idea is to use the matrix representation for terms and documents used in the Vector Space Model, and create a low-rank representation of the matrix with a Singular Value Decomposition. After this passage, it is possible to use the new

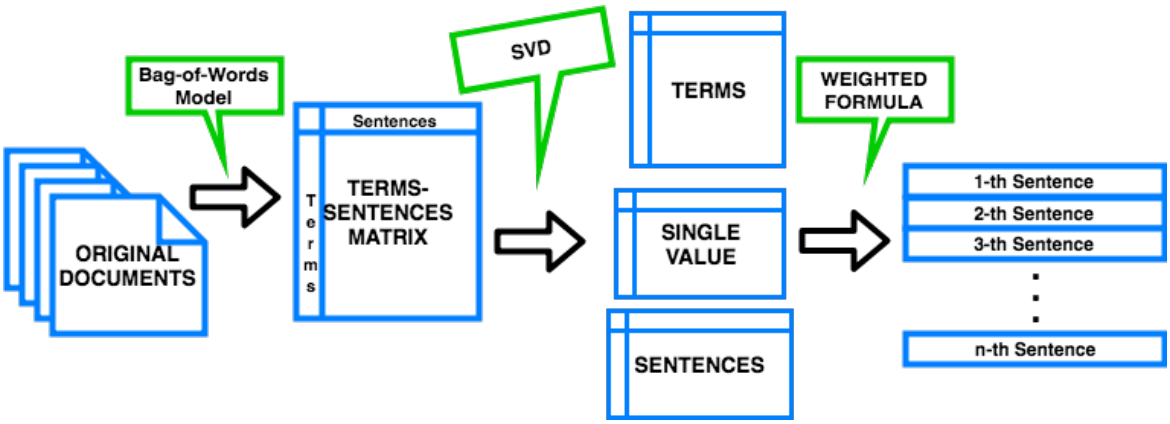

Fig. 3.3 Latent Semantic Analysis:

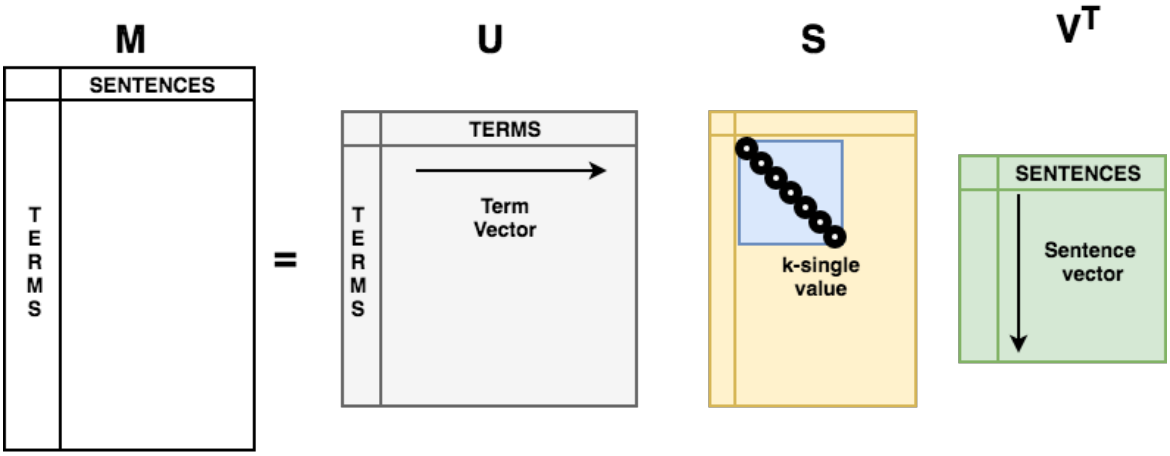

Fig. 3.4 Single Value Decomposition.

space with less dimensions to get information on the sentences or terms. Moreover, thanks to the sentences-score values, it is possible to "rank" sentences and terms by "relevance".

In the diagram **??**, LSA could be divided in two steps:

1. Low-rank representation with Single Value Decomposition (SVD;
2. Information extraction from the new space with the weighted formula 3.8.

The terms-docs matrix $M \in \mathbb{R}^{Dxn}$, produced by Bag-of-Words model, section 3.3, is the input for LSA. Without losing generality, suppose that the terms-docs matrix is scored by $tf.idf$, section 3.3.1, and the dimension choose for the subspace is p. Truncated Singular Value Decomposition is utilized to create a low rank representation for sentences and terms. To better understand how LSA works, let's summarize the SVD idea.

In the Figure 3.4, it represented the SVD of the original Terms-docs matrix $M = \{s_1, s_2, \ldots s_n\} = \{te_1, \ldots, te_D\}^T \in \mathbb{R}^{Dxn}$. In LSA approach, the columns of M are the sen-

tences $s_i \in \mathbb{R}^D$, thus the i-th column contains the information for the i-th sentence and at the j-th place M_{ij} is the $tf.idf$ score of the j-th terms in the sentence $s_i \in C$. In general, it is always the case that $D \gg n$ Furthermore, because of the number of different and new terms for each sentence, M is always sparse.

The Single Value Decomposition, explained in the appendix A, is applied to M and three matrix are obtained, as showed in figure 3.4. The matrix $S \in \mathbb{R}^{Dxn}$ is a diagonal matrix with the single value on the diagonal. The matrix $U \in \mathbb{R}^{DxD}$ contains information for the terms, meanwhile matrix $V \in \mathbb{R}^{nxn}$ has information for sentences. For the summarization purpose, only matrix $V \in$ and matrix D are considered.

"If a word combination pattern is salient and recurring in document, this pattern will be captured and represented by one of the singular vectors."[5]. This results is fundamental to be sure that all the information and pattern in the original matrix $M \in \mathbb{R}^{Dxn}$ are conserved in the low-rank space. Moreover, from a semantic point of view, "the SVD derives the latent semantic structure from the document represented by matrix M.".[27] The original set of sentences is represented in a subspace of $\mathbb{R}^{Dxn}$, such that $p \leq rk(M) = k$, with linearly-independent base vectors. For instances, in our original corpus, set of documents (or set of paragraph or set of sentences), there are 5046 different terms (4012 after text cleaning), the cardinality of the corpus (based on the Bm-level) is 46, the terms-docs matrix $M \in \mathbb{R}^{4012x46}$. If $rk(M) = 100$, it is possible to choose $p \leq 100$ such that: A_p, produced by SVD, is the best (see theorem 2 in appendix A) representation in the p-dimensional subspace of the original space generated by M. This approach is called truncated SVD.

As explained above, the patterns of the original documents are represented and conserved in the singular value dimensions. It is important that the subspace do not reconstruct exactly the original terms-docs matrix, in this way truncated SVD removes part of the noise that was in M. Because each of the p-dimensions of the new subspace are a representation of a salient pattern in the document, as in the paper [5] supposed, it is possible to hypothesize that each pattern is a representation of a salient topic in the text. So, if it is known a priori the number of topics, therefore p has to be chosen as the cardinality of topics. In the end, with the singular values $\sigma_1 \geq \cdots \geq \sigma_p$, Truncated SVD gives a measure of relevance of the topics, because the magnitude of the singular values is a measure of the importance of the pattern in the documents, see appendix A.

In the second part of LDA, it is extracted from the new low-rank space generated by SVD a score for each part of text, in our example for each sentence. As introduced in the paper "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis" by Y. Gong and X. Liu[11], singular value matrix S and right singular vector matrix V^T are used

[5] LSA

for this purpose. In the new space generated, each sentence s_i is represented by the column vector $v_i = (v_{i1}, \ldots, v_{ip}, \ldots, v_D)$ from V^T.

To create the score for each sentence based on the SVD decomposition made before, Matrix V^T and S are considered, for each sentence S_i, $i = 1, \ldots, n$ is computed a measure of "relevance" with the formula:

$$ss_i = \sum_{k=1}^{n} \sigma_k v_{ki} \tag{3.8}$$

In the end, for each sentence is available a score $R = \{ss_1, \ldots, ss_n\}$, sorting the values is obtained a rank of sentences from the most to the less relevant. If the number of sentences required for the summary are l, then the first l sentences are extracted from R in R^l. By a sort operation on R^l based on the original order in the text, it is obtained the final summary.

3.4.4 TextRank

The second summarization algorithm applied is TextRank. It is an extractive-based algorithm which gives a score of "relevance" for each sentence, therefore the final summary is a selection of the $k \in \mathbb{N}$ most relevant sentences.

As the name suggest, the core of TextRank algorithm is PageRank algorithm[24], the quite famous algorithm at the base of the search engine Google to rank website. PageRank algorithm is applied on the World Wide Web dataset to understand which are the most important website. To summarize the idea of PageRank, the score for each website is computed by the number and quality of links with other websites, therefore the assumption is that the most important websites are likely to receive link and the relevance of the other websites is significant.

The World Wide Web is represented as a directed weighted graph $G = \{V, E, f\}$, where V is the set of nodes that represents websites, $\forall\ v_i \in V$ there are some in-edges $IN_i(E)$, $e_{ji} \in E$ a generic edge from v_j to v_i, and out-edges $OUT_i(E)$, $e_{ij} \in E$ a generic edge from v_i to v_j. The function $f : E \to \mathbb{R}$ assigns a weight for each edge, such that:

$$f(e_{ij}) = a_{ij} \in \mathbb{R}, \qquad \forall e_{ij} \in E \tag{3.9}$$

The weighted scheme for the PageRank is just binary, $f(e_{ij}) = 1$ is the is a connection between node i and node j, otherwise $f(e_{ij}) = 0$. The formula of the PageRank to get the rank of each website is the following:

$$PR(v_i) = \frac{(1-\lambda)}{N} + \lambda \sum_{v_j \in IN_i(E)} \frac{PR(v_j)}{|OUT_j(E)|} \qquad \forall v_i \in V \tag{3.10}$$

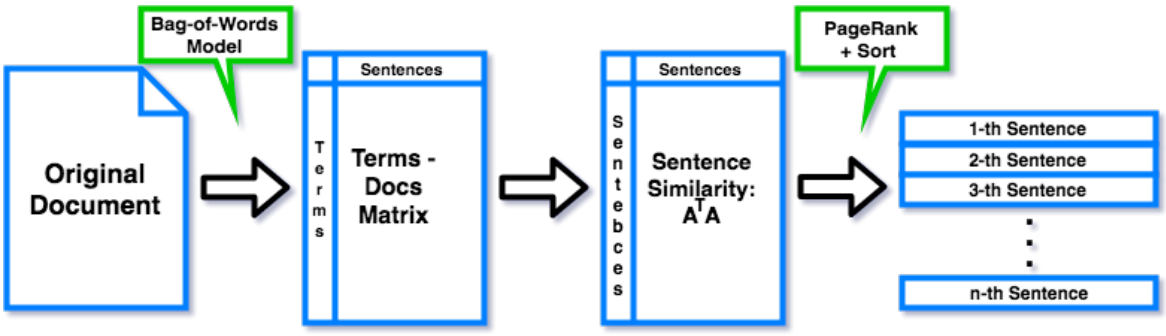

Fig. 3.5 TextRank: algorithm's flow.

Where N is the total number of websites and λ is a damping factor.

In the figure 3.5, it is reported the main step of the TextRank algorithms. Starting from the terms-docs matrix A, Bag-of-Words model (see section 3.3) is applied to create a space representation of the original document with the intention of compare with a distance the Part of Text in the document.

In TextRank algorithms is adopted a weighted Undirected graph, where nodes are Sentences (or more in general Part of text), and edges are weighted by a similarity score between the two sentences connected. In other words, the similarity between two sentences is a distance measure, usually the Cosine similarity. Furthermore, because the distance is symmetric between two sentence, the graph is Undirected.

With tis in mind, it is computed the similarity between each sentence. It is the same to say that it is calculated a product between the transpose of the terms-docs matrix M^T and M with a normalization for each element. So, the resulting matrix $B \in \mathbb{R}^{nxn}$ is a squared matrix with the dimension of the number of sentences n, so that:

$$\forall s_i, s_j \in V, \quad B_{ij} = \begin{cases} sim(s_i, s_j) = sim(s_j, s_i) \in (0,1], \quad if \quad i \neq j \quad \& \quad sim(s_i, s_j) > \varepsilon; \\ 0 \qquad if \quad i = j \quad or \quad sim(s_i, s_j) < \varepsilon. \end{cases} \tag{3.11}$$

The similarity measure used in this application is the Cosine distance:

$$w_{ij} = sim(s_i, s_j) = \frac{A_i^T \dots A_j}{||A_i^T|| ||A_j||} = \frac{\sum_{k=1}^{m} A_{ik}^T \sum_{k=1}^{m} A_{jk}}{\sqrt{\sum_{k=1}^{m} (A_{ik}^T)^2} \sqrt{\sum_{k=1}^{m} (A_{jk})^2}} \tag{3.12}$$

TextRank, as PageRank, use a graph to represent the information in between the sentences, therefore the final similarity matrix B is the weighted scheme of the graph G described above.

Notice that the Graph G is not completed-connected, in fact there are no self-loop and some connection are missing.

The last step of the algorithm, as represented in the right part of figure 3.5, is composed of three main step:

1. Modified PageRank algorithm;
2. Sort operation;
3. Selection of the first k sentences.

The formula of the Modified PageRank for the graph G is:

$$TR(s_i) = \frac{(1-\lambda)}{N} + \lambda \sum_{s_j \in IN_i(E)} \frac{w_{ji} TR(s_j)}{\left(\sum_{s_k \in OUT_j(E)} w_{ik} \right)} \qquad \forall s_i \in V. \tag{3.13}$$

There are some differences between 3.13 and 3.10. In fact, TextRank graph is weighted. The similarity between two sentences w_{ij} is used both to weight the influence of the TextRank score on another sentence $TR(s_j)$ and to normalize this influence by the total weight of the out-edges of s_j.

The output is a score for each sentence that represent the importance of the sentence in the document. Then, it is applied a sort operation on the sentences based on the weight $TR(s_i) \in (0, 1)$. In the end, the first k sentences are selected.

3.4.5 Summary Evaluation

text similarity

3.4.6 Weighted Formula with topics

3.5 Sentiment Analysis

The information overload issue described in the introduction has caused two main problem: the difficulty in understanding and the complexity of decision making.[32] It is crucial in our society to be able to collect information, but even more to be able to understand and actively use them.

Since the beginning of Web 2.0, a lot of opinionated text has recorded in digital format from social media and media communication sources, just to give some example: reviews

MINUTES	BOARD MEMBER	LSA - SIMILARITY	TEXTRANK - SIMILARITY
15-01	Martin FlodV©n	0,908	0,908
15-01	Per Jansson	0,877	0,877
15-01	Kerstin af Jochnick	0,872	0,872
15-01	Cecilia Skingsley	0,866	0,866
15-02	Stefan Ingves	0,835	0,835
15-02	Martin FlodV©n	0,865	0,866
15-02	Henry Ohlsson	0,804	0,804
15-02	Per Jansson	0,845	0,845
15-02	Kerstin af Jochnick	0,846	0,846
15-02	Cecilia Skingsley	0,887	0,887
15-07	Stefan Ingves	0,828	0,828
15-07	Martin FlodV©n	0,829	0,865
15-07	Henry Ohlsson	0,836	0,821
15-07	Per Jansson	0,857	0,857
16-07	Stefan Ingves	0,848	0,885
16-07	Martin FlodV©n	0,883	0,883
16-07	Henry Ohlsson	0,86	0,86
16-07	Per Jansson	0,853	0,853
16-07	Cecilia Skingsley	0,867	0,867
15-09	Stefan Ingves	0,933	0,933
15-09	Martin FlodV©n	0,857	0,873
15-09	Henry Ohlsson	0,86	0,86
15-09	Per Jansson	0,896	0,896
16-09	Stefan Ingves	0,879	0,883
16-09	Martin FlodV©n	0,897	0,921
16-09	Henry Ohlsson	0,925	0,931
16-09	Per Jansson	0,946	0,946
16-09	Cecilia Skingsley	0,809	0,84
16-11	Stefan Ingves	0,96	0,96
16-11	Martin FlodV©n	0,841	0,852
16-11	Henry Ohlsson	0,884	0,861
16-11	Per Jansson	0,856	0,856
17-04	Per Jansson	0,877	0,882
17-04	Martin FlodV©n	0,899	0,911
17-04	Kerstin af Jochnick	0,795	0,795

Fig. 3.6 Summary: sentences-score by LSA and TextRank.

on e-commerce websites, chats for interpersonal communications, comments on blogs and social networks.

The natural consequence of the availability of this new sources of information is the growing number of researches in both Linguistic and Machine Learning to develop techniques able to discover the opinion contained in texts. The umbrella name to call all this researches is Sentiment Analysis.[6]

In the last few years, social media analysis has increased popularity, and sentiment analysis is a core part of it, with the purpose of: "to extract from the social media content is what people talk about and what their opinions are.". Moreover, it is increasing the idea of studying the opinion holders them-self, mainly for social analysis and customer profiling.[7]

With the data available in our project, it is straightforward to apply sentiment analysis techniques to better understand the opinion of the Board Members of Riksbank. As explained in the economical introduction 1.2, one of the two goals of this project is to identify the Hawkish-Dovish position for each board member in a Minutes.

After a review of the main techniques used in sentiment analysis, two approach are applied:

- Supervised algos;
- Lexicon-based approach.

In the following paragraphs, initially it is given a more in-depth overview of opinion mining, then the two approach are explained and finally the results are evaluated.

3.5.1 Opinion and Sentiment

"Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text.".[33] The definition given by Bing Liu describes and highlights the main aspects of sentiment analysis. First of all, it is important to describe the difference between sentiment and opinion, and how they are related.

In his work, Bing Liu has explored this difference, two sentence are given as an example: *"I am concerned about the current state of the economy"* and *"I think the economy is not doing well"*. By the first sentence, the speaker expresses his/her feeling and implies a negative opinion about the economical situation, instead the second sentence reveals the concrete view

[6] Notice that the field of Sentiment Analysis is also called: opinion mining, opinion analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining.

[7] cite:economical and social eamples.

point of the author and express his/her opinion on it. The difference it is not so remarkable, however it is possible to classify the first sentence as a sentiment expression and the second as an opinion.

To summarized, "sentiment" is used "to mean the underlying positive or negative feeling implied by opinion"[33] and "opinion to mean the whole concept of sentiment, evaluation, appraisal, or attitude and associated information, such as the opinion target and the person who holds the opinion"[33].

In review mining, sentiment analysis is applied to reviews of products, the problem of opinion mining is usually divided in sub-problems. It is identified an *entity*, that is the subject/product under investigation, and the attributes/characteristics of the entity are designated as *aspects*. Bing Liu, as state-of-the-art in Aspect-Based Sentiment Analysis, gives a comprehensive definition of opinion from a theoretical point of view in therms of quintuple:

$$\text{(Entity, Aspect, Sentiment, Holder, Time)} \tag{3.14}$$

where:

- Entity: it is the target entity/object of the analysis;
- Aspect: it is the aspect/feature of the entity analyzed;
- Sentiment: it is score given to identify the sentiment of the opinion;
- Holder: it is the opinion holder;
- Time: it is period of validity for the sentiment on the opinion.

Furthermore, the sentiment analysis goal, given a document d, is to extract all the quintuples from d.

3.5.2 Our Approach

To apply this idea on our data, it is important to understand where the information, to complete each quintuple, is hidden. To summarize, in this project the goal of sentiment analysis is to uncover the polarity of the Board Members concerning monetary policy decision in the last minutes released by Riksbank.

Concerning the Aspect-Based model 3.14, it is straightforward to identify the *Time t* with the date of the minutes under analysis. Therefore, since there are 30 Minutes available: $t = 1, \ldots, 30$. The *Opinion Holder h* are the Board Members, so t = {Stefan Ingves, Kerstin af Jochnick, Martin Flodén, Per Jansson, Henry Ohlsson, Cecilia Skingsley}.

Approaches	Supervised	Lexicon-Based
Entity-Based	Supervised-Entity	Lexicon-Entity
Fixed-Aspects	Supervised-Fixed	Lexicon-Fixed
Dynamic-Aspects	Supervised-Dynamic	Lexicon-Dynamic

Table 3.1 Sentiment Analysis Models

About the *Entity* and *Aspects*, a more complex analysis has to be done. The *Entity* could be identified, for all the opinion holder h and independently from the time t, as the "Monetary policy decisions". The *aspects* that describe the entity can be various, in our work we decided to tackle the problem by different approaches:

- Entity-based: the model for the opinion description is simplified as:

  $$\text{(Entity, Sentiment, Holder, Time)} \tag{3.15}$$

  Therefore, the hypothesis is that there are no aspects that particularly describe the monetary decision of a Board Member. For each h, all the part of text dedicated to him/her in the minutes are used to analyzed the *sentiment*.

- Fixed-aspect: the aspects are independent from the time, therefore are the same for each minutes. These aspects are the Topics suggested from the Research Team of SEB, and are: a = {Inflation, ECB, FED, Krona}, see section 2.1.1.

- Dynamic-Aspects: the aspects could change over time and depends by the topics debated in the Board Members meeting. This approach uses a Topic Model to identify the Aspects, see section 4

To clarify the interpretation for the *Entity* and *Aspects*, a distinction on the different levels of analysis should be done. In general, it is possible to use as input in the analysis different part of text: the document itself or, cutting the text, paragraphs or sentences.

In the Supervised approach, described in section 3.5.3, all the sentences are used as a unique text, therefore the analysis is at a document-level. Instead, in the Lexicon-based approach, see section 3.5.4, the sentences are analyzed individually and then the information are recombine.

About the *Sentiment*, it is interpreted as a binary class to label each Board Member h (or each Topic in a, if the aspect based is used), s = {Dovish, Hawkish}. Therefore, the Sentiment analysis issue is translated in a classification problem. In both Supervised and Lexicon-based approaches, the results can be translate in a score between 0 and 1, close to 0 is Dovish and toward 1 is Hawkish. The score gives the possibility to compare the results, for instance if a

Board Member has a score of 0.1 and another one has 0.6, it could be inferred that the latter is more hawkish. Therefore, a rank can be created.

To summarize the different models created see table 3.1.

3.5.3 Supervised algorithm

There are three main step to focus on in Supervised sentiment analysis[25], highlighted in green in figure 3.7:

- Tokenization: this step is analyzed in section 3.1;
- Feature Extraction: features are extracted from the test-set, with some criteria further discussed, and used to build the Design matrix for both the test-set and the train-set.
- Classification Model: the Train-set Design matrix is used to train the classification model. In the next sections are explained the classifiers used: Naïve-Bayes, SVM, MaxEnt.

Feature extraction

Text data are very unstructured, therefore it is not straightforward to extract features from the text and it is also challenging to find a criteria to select the best features. The idea of feature extraction is to learn patterns in data to give a more organized shape to the information available. Usually the feature extraction and selection algorithms use vectors of numbers, with text data how to encode words is a second challenge to be tackle. Therefore, Feature extraction and selection are two fundamental phase of the supervised approach: the first to get the right information available to be analyzed and the latter to optimize information in the algorithms. The models used and described in this section follow the idea of the Bag-of-Words model to encode words, described in section 3.2 and are presented in order of complexity.

In table 3.1, three models are introduced for the Supervised approach. However, it is possible to focus on the supervised algorithms without this distinction. In fact, the algorithms are always applied at a document-level and the terms-docs matrix, introduced in section 3.2, is computed on the text selected. Three weighted schemes are used for the Bag-of-Words model:

- Frequency-based: is used only the frequency of a term in a document and the features are the first k most important words;

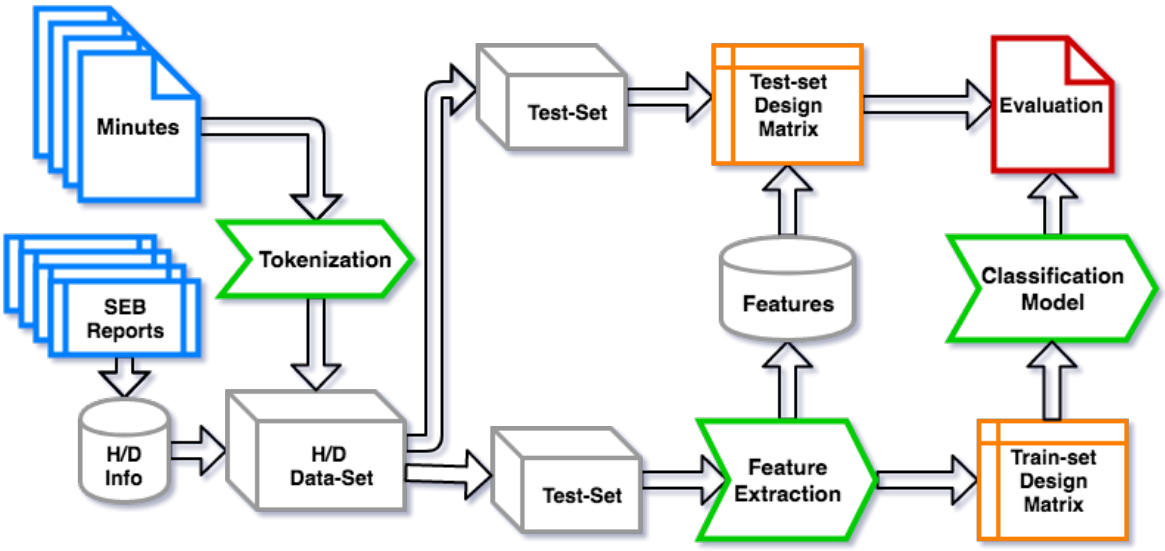

Fig. 3.7 Sentiment Analysis: Supervised algos.

- Tf.idf: the collection of document is used to dump the frequency of the terms, see section 3.3.1;
- Word2Vec: the Word2Vec model, presented in section 3.3.1, is combined with the

As showed in figure 3.7, the same techniques used to extract the features in the train-set must be used for the test-set and then in the prediction phase with the new document.

Classification Models

After the feature extraction, the design matrix is ready to be used to train the classifier. There are many types of classification algorithms used with text data, Naïve Bayes, Support Vector Machines and Maximum Entropy Models are the state-of-the-art of these techniques, in the next sections it is explained idea behind these algorithms.

Naïve Bayes

Bayesian classifiers are based on a probabilistic model, build on the so-called *naive Bayes assumption*: all the attributes of the examples are independent of each other, given the context of the class. This hypothesis is false in most of the cases but nevertheless the model perform surprisingly well.

Using the Naive Bayes assumption, different models can be created, the two more used are the Multivariate Bernoulli model and the Multinomial model. In the Bernoulli model, a document is represented by a binary vector that in each component indicates the presence or

the absence of each word. In this model, the number of times a word occurs is not considered. In the Multinomial model, the number of occurrences of the words is stored in the vector that defines the document. The latter model usually outperforms the first one with large vocabulary size, there is an average improvement of 27%.[21]

It can be shown that the number of misclassification is minimized, on average, "assigning to each observation the most likely class given its predictor values".[16] In other words the right class is the one that maximized the following probability:

$$\max_{y \in Y} P(Y = y | X = x_0) \tag{3.16}$$

Using the Bayes theorem and the naïve assumption is possible to write:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} = \frac{P(Y)P(x_1, x_2, ..., x_n|Y)}{P(x_1, x_2, ..., x_n)}. \tag{3.17}$$

$$P(x_1, x_2, ..., x_n|Y) = \prod_{i=1}^{n} P(x_i|Y). \tag{3.18}$$

and the probability becomes:

$$P(Y|x_1, x_2, ..., x_n) = \frac{P(Y)\prod_{i=1}^{n} P(x_i|Y)}{P(x_1, x_2, ..., x_n)}. \tag{3.19}$$

since the denominator is constant $\forall y \in Y$, it can be ignored. In the end, the function to maximizeto get the prediction of the class is:

$$y = argmax_{j=1,..K} P(Y_j) \prod_{i=1}^{n} P(x_i|Y_j). \tag{3.20}$$

Support vector machine

The support vector machine classifier has been developed in the 1990's and it is still widely used nowadays.[16] It is based on the intuition to divide the two classes of observations using an hyperplane.

The model has evolved, it has been generalized and more advanced functions can be used to separate the two or more classes, as polynomial functions and radial basis functions.

To present the mathematical model, it is considered the linear classifier in the binary case. The algorithm seeks for the hyperplane that maximize the distance between itself and the observations of the two classes. The closest observations define the hyperplane and they are called *support vectors*, the other observations are independent of the hyperplane position.

This total dependence, between support vectors and the classification hyperplane, would make the result too sensitive to small changes in the support vectors. For this reason, some misclassification errors are tolerated to reach a greater robustness of the algorithm. The mathematical formulation of the problem is the following:

$$\begin{aligned} &\max_{\beta_0,\beta_1,\dots,\beta_p,e_1,\dots,e_n} M \\ &\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \\ &y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} \geq M(1 - e_i), \\ &e_i > 0, \quad \sum_{i=1}^{n} e_i \leq C, \end{aligned} \tag{3.21}$$

Where M is the margin, $e_1, \dots, e_n$ are *slack variables*, that allow the observations to be on the wrong side of the hyperplane. C gives an upper bound to the number of misclassification and y_i is the binary variable that indicates the class to which the observation i belong.

Usually data are not linearly separated, furthermore the most used version of this classifier use a more complex shape to fit the data. To describe it, it is necessary to introduce the notion of *kernel* function, it could be seen as a generalization of the inner product. The most used are:

$$\begin{aligned} \text{Linear kernel:} &K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}; \\ \text{Polynomial kernel:} &K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d; \\ \text{Radial kernel:} &K(x_i, x_{i'}) = \exp^{(-\gamma \sum_{j=1}^{p} (x_{ij} x_{i'j})^2)}, \quad \gamma > 0. \end{aligned} \tag{3.22}$$

It can be shown that the solution of the classification algorithm depends just on the inner products of the observations. The boundary function has the form:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i). \tag{3.23}$$

An example of the result is showed in the table 3.2.

SVM	BoW	Tf-idf	Avg-Word2vec	Avg-Tf-idf Word2vec
Accuracy	0.92	0.92	0.54	0.54
F1	0.92	0.92	0.44	0.38
Precision	0.93	0.93	0.77	0.29
Recall	0.92	0.92	0.54	0.54

Table 3.2 Cofusion Matrix for Support Vector Machine Algorithm in Supervised-Entity Model.

Maximum Entropy Model

Evaluation

3.5.4 Lexicon-based

As previously discussed, the main problem of text data is to extract a structure from the information available to use it. In the supervised approach 3.5.3, it is discussed the Bag-of-Words model. With this idea, it is created a vector space to represent documents. In this section, it is analyzed another approach to extract features with an external source of information, lexicons or dictionaries.

Lexicons[8] are a collection of words with some additional information, the criteria to select the words can be based on:

- Subject: the words in the dictionary are from a specific field or category, such as Economical terms.
- Information: the words are selected because of their effect in the meaning of the sentence, for instance positive or negative words.

The information provided for each terms in the lexicon can be also very different, however the most used are:

- positive/negative score:
- Subjectivity/objectivity score:
- Mood sore
- modality score:

[8] Also colled Dictionaries

Usually, lexicon-based algorithms for sentiment analysis are used at a sentence-level. In fact, the scores presented above give a lot of additional information for each sentence in the documents, even if the scores can not be used directly on the documents, it is possible to transfer these informations at a document-level by weighted formulas.

For instance, in many papers [33, 29] it is used as an assumption that only sentences with an high score of subjectivity are classifiable in a sentiment way.

An additional issue of lexicon-based approach, it is that Part of Speech Tagging is fundamental to be sure of assign the right term in the document with the right meaning in the lexicons. The sequence of letters that compose a words is not a perfect identifier of the meaning, in fact the meaning of most of the words in every languages depends on the context in which the word is. Part of Speech Tagging is discussed in section **??**.

Our Approach

Different Lexicons have been applied after using some Python Libraries for Part-of-Speech Tagging. Two different approaches has been followed with the lexicon-based algorithms:

- Weighted-formula: the most subjective sentences are selected and classified as opinionated text. A weighted formula is used to take into account the scores and give a H/D final score.
- Supervised-approach: scores can be seen as features and the same classification models from section 3.5.3 can be used.

Lexicons

3.5.5 Evaluation

Chapter 4

Topic Model

4.1 Introduction

Topic Modeling is a useful model to extract "abstract" topic in a collection of documents. The main idea behind the model is to create cluster of "similar" words/ngram and use them to describe the topics of the collection of document. In particular, on one hand the process categorizes in different clusters the words/ngrams in the collection of documents and on the other hand extracts the most representative documents for each document from each cluster.

The main hypothesis behind this model is that: texts are generated according to a specific model. LDA (Latent Dirichlet Allocation) is a typical model used as assumption, so the hypothesis in with this model is that: topics are generated from a Dirichlet distribution. The parameters of the model are assumed by Bayesian estimation.

4.2 Bayesian Network

> "The subject of probability is over two hundred years old and for the whole period of its existence there has been dispute about its meaning. [...] When a question has proved to be difficult to answer, one possibility may be that the question itself was wrongly posed and, consequently, unanswerable. This is de Finetti's way out of the impasse. Probability does not exist."
>
> D.V. Lindley, Foreward of: "Theory of Probability A Critical Introductory Treatment", Bruno De Finetti"

A first basic distinction that has to be done concern the difference between classical probability and the Bayesian approach. In this context, it is important to notice the passage from the classical notion of probability and the idea introduced with Bayesian probabilities of the degree of belief. Beliefs on the "object" under examination are subjective to someone opinion , for this reason, the probabilities in Bayesian Probability theory are conditional to some prior knowledge or assumptions. To avoid, philosophical discussion regarding the meaning of probability, as specified in the introduction of the paragraph, it is given a standard definition of Bayesian Probability Theory.

All start from the Bayes' rule regarding the manipulation of conditional probabilities. First, the Product rule is a key point. The joint probability of two events A and B can be expressed as:

$$P(H,D) = P(H|D)P(D) = P(D|H)P(H) \tag{4.1}$$

In the Bayesian interpretation of probability, the two events could be distinguished for example in Hypothesis and Observed data. Let assume that the goal is to infer the validity of the Hypothesis from the observed data. With the relation 4.1, it is possible to have an explicit formula to infer from data:

$$P(H|D)P(D) = P(D|H)P(H) \Longrightarrow P(H|D) = \frac{P(D|H)P(H)}{P(D)} \tag{4.2}$$

The above formula is also known as Bays Theorem. A specific notation is used in Bayesian theory to indicate each term of the formula:

- $P(D|H)$ is known as likelihood and indicates the probability of the observed data given certain hypothesis;
- $P(H)$ is known as prior and contains the a priori knowledge about the hypothesis before the observation of the data.
- $P(H|D)$ is the posterior of the model and collects the final knowledge after the observations summed to the ex-ante information.

The term $P(D)$ is not known or observed. However, thanks to the Marginalization principle it is possible to explicitly calculate it. In fact, the Marginalization principle states that, if the possible classes of H are exhaustive and mutually exclusive, then:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} = \frac{P(D|H)P(H)}{\sum_{h\in H} P(h)P(D|h)} \tag{4.3}$$

In the end, the Bayes Rule and the Marginalization Principle allow to compute the posterior $P(H|D)$ by only the estimate of $P(h)$ and $P(D|h)$

It is important to understand that the experimenter usually knows the likelihoods values, as the observation registered under a particular hypothesis. Furthermore, the most subjective part of this approach is held in the a priori knowledge contained in the prior. If on one hand, the subjectivity could be thought as a weakens of the model, because some assumptions have to be done with some hypothesis; on the other hand, it allows to update the previous knowledge. In fact, $P(D|H)$ and $P(D)$, can be seen as the weights to update the prior knowledge and transform it in the ex-post knowledge with the posterior. This process is called Bayesian Updating.

4.2.1 Graphical Models

A graphical model is a tool used to illustrate in a visual way and work with conditional independence between variables in a given problem [INT]. The conditional independence notion between two variables could be explained by the absence of a direct impact on each other's value. With the probabilistic notation, taken three random variables (X,Y,Z), the conditional independence of X from Z given Y is true if $P(X|Z,Y) = P(X|,Y)$. A shared notation for conditional independence is $X \perp\!\!\!\perp Z|Y$.

In Graph Theory, it is usual to indicate a particular graph G with a set of nodes V and a set of edges E, so $G = \{V,E\}$. For a good introduction to Graph Theory see [1]. In Graphical Models theory, nodes are random variables and edges represents causal relationships between

[1] cite a good book

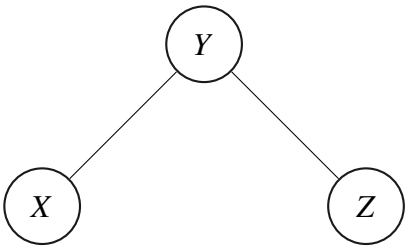

Fig. 4.1 Graphical model with three nodes and conditional independence between nodes X and Z given node Y.

variables, if the edge is directional the direction is from the cause variable to the effect variable, otherwise, if there is only a correlation between two variables, the edge is undirected[2].

Just to give a simple example, let's consider again the above definition of conditional independence with the three random variables (X,Y,Z), but from a Graphical Model perspective. A graph $G = \{V,E\}$, with $V = \{X,Y,Z\}$ and $E = \{(X,Y),(Y,Z)\}$, is the right representation of the conditional independence between X and Z given Y.

In picture 4.1, there is a connection between X and Y, and Y and Z, but there is no connection between X and Z, to indicate the conditional independence. It is not the same as independence between variables, $X \perp\!\!\!\perp Z$. In fact, it only states that the random variable Y does not encode any information from X that is relevant to the random variable Z, and vice-versa.[INT]

The graphical representation is not sufficient to completely represent the whole model, indeed probability distribution functions are needed to describe the random variables in the model. In this work are considered only discrete random variables. The graph is helpful also in this context, because edges represent which values influence a certain distribution. For instance, considering the graph 4.1, the values of X and Z influence the distribution of Y and only the values of Y influence the distribution of X and Z, but no value of X can directly influence the distribution of Z and vice-versa.

4.2.2 DAG

Bayesian networks are a particular type of Graphical Models, the difference concerns the type of graph involved to represent the conditional independence between variables. Therefore, Bayesian Networks are represented by Directed Acyclic Graphs (DAG).

[2] COWEL

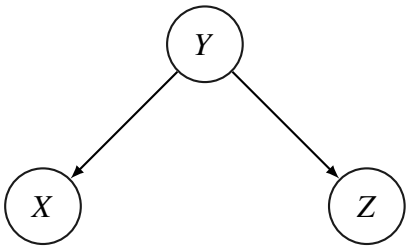

Fig. 4.2 DAG, three random variables with conditional independence between the variables X and Z the variable Y.

DAGs are graph with directed edges, connections between variables could be only in one direction, and without cycles. [3]

The representation with DAGs is an easy way to understand how to factorize the distributions of the variables in the model. In fact, the edges in the Bayesian network encode a particular factorization of the joint distribution. [INT] The Chain Rule (or Product rule) is used combined with the Conditional independence to create a simplified form of the joint distribution. The general formula could be summarized for a given set of random variables $X = \{X_1, ..., X_n\}$ as:

$$P(X) = \prod_{i=1}^{n} P(X_i | par(X_i)) \tag{4.4}$$

where $par(X_i)$ indicates the nodes that are parents of nodes X_i.

Picture 4.2 shows the DAG representation of the previous example with the three random variables. The chain rule [4] is useful to obtain a more convenient form of the joint distribution:

$$P(X,Y,Z) = P(Y)P(X|Y)P(Z|Y,X) \tag{4.5}$$

Then with the conditional independence knowledge regarding the variables, the joint distribution is further simplified as:

$$P(X,Y,Z) = P(Y)P(X|Y)P(Z|Y) \tag{4.6}$$

A convenient form to visualize and use the probabilities that define the random variables is Conditional Probability tables (CPT). In fact, if the set of random variables used to describe the problem are discrete and mutually dependent as described above, with CPT is possible to obtain for each variable the probability conditioned on the other variable that can influence

[3] A chain in a graph is a series of nodes where each successive nodes in the series has a connection (regardless of the direction) with the previous one. A path is a chain with the direction of the edge corresponding with the direction of the chain. A cycle is a path that starts and ends in the same node

[4]

	y_1	y_2	y_3
x_1	$P(X = x_1 \mid Y = y_1) = p_{x_1\mid y_1}$	$p_{x_1\mid y_2}$	$p_{x_1\mid y_3}$
x_2	$p_{x_2\mid y_1}$	$p_{x_2\mid y_2}$	$p_{x_2\mid y_3}$
x_3	$p_{x_3\mid y_1}$	$p_{x_3\mid y_2}$	$p_{x_3\mid y_3}$
x_4	$p_{x_4\mid y_1}$	$p_{x_4\mid y_2}$	$p_{x_4\mid y_3}$

Table 4.1 Conditional Probability Table (CPT) for the random variable X given the conditional independence between X and Z given Y.

	Known Structure	**Unknown Structure**
Fully Observability		
Partial Observability		

Table 4.2 Learning a Bayesian Network

the value. In table 4.1 there is an example of a CPT for the random variable X, for each value of $X \in \{x_1, x_2, x_3\}$ and for each possible value of the conditional variable $Y = \{y_1, y_2, y_3\}$ is showed the probability. Notice that the variable Z is not present in the table due to the conditional independence between X and Z given Y.[10]

4.2.3 Learning probabilities in a Bayesian Network

The topological structure of the DAG and the probability distribution of the variables completely describe the problem under examination. However, both the structure and the distributions have to be learned from the reality. In this context, it is possible to distinguish four different situations, summarized in table 4.2.

For this reason, it is usual to speak about learning in Bayesian Network. In this work, we are in the second situation, where the topological structure of the graph is assumed fixed under certain hypothesis about the conditional independence between variables, the priors are the collection of the partial knowledge ex-ante the updating process. Furthermore, only some of the variables are observed during the learning process.

The data observed is the new source of knowledge to add in the model to the a priori knowledge collected in the priors. There are a set of techniques used to improve to the prior information with the new one . The main idea of the learning probabilities from a fixed structure of the network is exposed in this section.

Assume that the knowledge about the causalities between variables is encoded in the structure of the network G. In the following formula, G represents the hypothesis made to have a certain graph structure and so a particular factorization of the joint distribution. By the formula 4.4, it is possible to write the probability joint distribution of the variables of the

model $X = (X_1, ..., X_n)$, as follow:

$$P(X|\theta, G) = \prod_{i=1}^{n} P(X_i|par(X_i), \theta_i, G). \tag{4.7}$$

Where θ is the set of parameters used to describe the distributions of the random variables, thus θ_i is the set of parameters to describe the distribution of $P(X_i|par(X_i), \theta_i, G)$. The uncertainty of the model is encoded in the distribution of the random variables used to represent the variable. In fact, the prior knowledge is encoded in the parameters θ of the distributions of the random variable. $P(\theta|G)$ is the prior probability density function assumed for the distribution of joint distribution.

Furthermore, it is considered a set of random sample $D = \{x_1, ..., x_m\}$ from the joint probability distribution of X. This is the new knowledge that it is used to learn the probabilities of the model, summed to the prior one.

Finally, the problem of learning the probabilities in a Bayesian Network can be summarized as: given a random sample D, compute the posterior distribution $P(\theta|D, G)$. [14]

4.2.4 Distribution

To complete this short introduction to DAGs and before finishing the explanation about learning probabilities, the distributions of the random variables in the models are presented. The random variables in DAGs represent some events of the problem under examination. To analyze the Topic Model, it is possible to focus the attention to just categorical events. It means that the distribution involved have finite space of states.

In particular, binary events are analyzed. In fact, the topic models, described in section 4.3, analyze the presence or absence of words. Thus, a node w represents an event and his indicator ${I_w}^5$ is used to check the presence or absence. If the word is observed the indicator will be true with the value $I_w = 1$, otherwise $I_w = 0$ to represent the absence.[14]

The indicator I_w is a binary variable $I_w \in \{0, 1\}$, called Bernoulli. For a more accurate description of the Bernoulli Distribution see the Appendix. The Bernoulli distribution is characterized by only one parameter $p \in [0, 1]$, that is the probability of I_w to be equal to 1, $P(I_w = 1) = p$ and the opposite event is described by $P(I_w = 0) = 1 - p$.

However, the events could occur several times and the idea to describe these events is to check the value of the event every time. In the topic models, the events, as said above, are the presence or absence of a particular word w. The Binomial distribution describes this type of events, one more parameter is needed to describe a Binomial Distribution. Thus, aside from

[5] indicator definition

p also n is used and represents the number of occurrences of the event. The shared notation for a Binomial Distribution is $Bin(p,n)$. Furthermore, the variable described is no more the indicator I_w but the sum of the indicators for different texts $w = \sum_{t \in T} I_w^t$, with the hypothesis that the events $t \in T$ are independent and identically distributed (i.i.d.).

The Multinomial distribution is a generalization of the Binomial distribution. In fact, it describes an event that can have multiple outcomes, $E \in \{o_1, ..., o_k\}$ means that the event E can assume k-different values. Moreover, the Multinomial distribution as the Binomial models describes events that occur several times, thus one of the two parameters of the distribution is n. The second parameter used to describe a Multinomial distribution is a vector with the probabilities for the k-different categories: $p = (p_1, ..., p_k)$. Since the probabilities have the sum equal to 1, $p_1 + ... + p_k = 1$, one parameter is redundant, because it could be obtained from the others:

$$p = (p_1, ..., p_{k-1}) \in R^{k-1}, \qquad p_k = 1 - \sum_{i=1}^{k-1} p_i \tag{4.8}$$

The Multinomial distribution is denoted with $Mult(p,n)$. It is a discrete distribution in a K-dimensional space where the elements of the vector are integer $x \in Z_+^K$, where $\sum_{i=1}^{k} x_i = n$. p is an element of the simplex space S_k, that is a space where all the vectors have the sum equal to 1 of (k-1)-dimension.[6] The probability density mass function is:

$$f(x_1, ..., x_k; p_1, ..., p_k; n) = \frac{\Gamma(n+1)}{\prod_{i=1}^{k} \Gamma(x_i + 1)} \prod_{i=1}^{k} x_i^{\alpha_i - 1} = \frac{1}{Beta(\alpha)} \prod_{i=1}^{k} p_i^{x_i}$$

The interpretation of the Multinomial distribution can be viewed as drawing n i.i.d values from a Multinoulli distribution with the probability of probability mass function $f(X = 1) = p_i$. Thus, for each element x_i of the vector X counts the number of occurrences that the i-th value appear.

Another important distribution, frequently used with Bayesian Models is the Multinoulli distribution (or Categorical distribution, or Generalized Bernoulli distribution). It is a special case of the Multinomial distribution with $n = 1$.

In both the Multinoulli and Multinomial distributions, the outcome of the distribution can be described with one number that indicates the number of the class selected. In the context of Natural Language Processing, it is usually adopted another notation, a k-dimensional vector with all the elements equal to 0 and only one element equal to one in the position of the class of the outcome.

[6] for the same reason of **??**

In the field of Natural Language Processing, the events described are the presence or absence of a particular word. Thus, the possible classes are a dictionary of words, the dimension of the dictionary and the type of words contained depend on the application. In this first example, it is possible to consider the dictionary *DIC* as the set of all the possible words in the English language.

Each word can be simply characterized with a k-dimensional vector w^j, where k is the dimension of the dictionary DIR. As described above, the vector w^j has all zero elements except for one element equal to one in the j-th position that corresponds to the position of the word considered in the dictionary:

$$w^j = (0, ..., 0, 1, 0, ..., 0) \in R^k.$$

If it is thought in a vector space, the set of all the vectors of the words $\{w^1, ..., w^j, ..., w^k\}$ is an Orthonormal Base of the k-dimensional space created by the dictionary *DIC*.

The Dirichlet distribution is a probability distribution over a probability simplex S_k. It is denoted with $Dir(\alpha)$, where $\alpha = (alpha_1, ..., \alpha_k) \in R^k$, $\alpha_i > 0 \forall i$ and $k \geq 2$. The probability density function of $X = \{x_1, .., x_k\}$ is equal to:

$$f(x_1, ..., x_k; \alpha_1, ..., \alpha_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1} = \frac{1}{Beta(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

where $\{x\}_{i=1}^{k} \in S$ means that $\sum_{i=1}^{k} x_i = 1$ and $x_i > 0 \forall i = 1, ..., k$. If all the elements of α have the same value $\alpha_1, ..., \alpha_k = \hat{\alpha}$ it is called a symmetric Dirichlet distribution.[Tu]

The idea of the Dirichlet distribution is that can model the probabilities of an event that has k possible states. In fact, the numbers in the vectors of a random variables distributed as a Dirichlet sum to 1, thus it is frequently used to deal with categorical distribution. The role of the parameters α is fundamental to understand the meaning of a Dirichlet distribution: they can be interpreted as the frequency registered for a particular state. Thus, if the Dirichlet distribution is used to model the prior knowledge and the posterior is again distributed as a Dirichlet, the update can be seen as the sum to the ex-ante parameters with the frequency registered in the data observed.

4.2.5 Cojugate Priors

The learning problem in Bayesian Network, described above, wants to use a given random sample D, to compute the posterior distribution $P(\theta|D, G)$, where G records the information of the conditional independence between the variables. [14]

The prior and the posterior are called conjugate distribution, if the posterior distribution $P(\theta|D,G)$ has the same probability distribution family as the prior probability distribution $P(\theta)$, named conjugate prior of the likelihood function.

The Dirichlet distribution is a conjugate prior with two conjugate likelihood functions: the Multinoulli distribution and the Multinomial distribution. Thus, if the randomly sampled observations $x \in D$ has a categorical or a multinomial distribution, moreover if the prior distribution of the parameters $P(\theta)$ is distributed as a Dirichlet, then the posterior distribution $P(\theta|D,G)$ is also a distributed as a Dirichlet. In this case, it is possible to start from the a priori knowledge contained in the Dirichlet distribution of the prior and then update the knowledge with a modification of the parameters of the prior Dirichlet forming a new Dirichlet that describes the posterior knowledge, the updated regards only the hyper-parameters of the Dirichlet. Finally, the advantage of this method is that there is a closed formula to describe the update without the necessity of numerical integration. The downside of this approach is that the model is constrained to only probability distribution that are conjugate to each other [7].

In a probabilistic notation, what explained above can be represented as:

$$\begin{aligned} \alpha &= (\alpha_1, ..., \alpha_k) \\ \theta|\alpha &= (\theta_1, ..., \theta_k) \sim Dir(k, \alpha) \\ X|\theta &= (x_1, ..., x_k) \sim Mult(k, p) \end{aligned} \tag{4.9}$$

then the posteriors have the form:

$$\begin{aligned} N &= (N_1, ..., N_k) \\ \theta|X, \alpha &\sim Dir(k, \alpha + N) \end{aligned} \tag{4.10}$$

Where N counts the occurrence of the knowledge acquired for each possible state: N_i represents the number of occurrence of the i-th state in the random sample D. The posterior distribution $P(\theta|D,G)$, also indicated as $\theta|X,\alpha$, is a Dirichlet with the hyperparameter α updated to $\alpha + N$ that means:

$$\theta|X, \alpha \sim Dir(k, \alpha + N) = Dir(k, (\alpha_1 + N_1, ..., \alpha_k + N_k)) \tag{4.11}$$

Ones the posterior distribution of the joint distribution is obtained, it could be useful to retrieve the marginal distribution. In fact, to model the problem under analysis, as the

[7] cite a table with all the conjugate distributions

case of the topic model, only a few key distributions are relevant. The advantage of the conjugate priors is that the distribution of interest is obtained analytically. The most relevant distribution usually used are:

- Posterior predictive: if it is considered the set of random sample D, the distribution of a new observation $x \in D$, under i.i.d. assumptions, is:

$$P(x|D) = E_{P(\theta|D,G)}(P(x,\theta|D)) = \int P(x,\theta|D)d\theta = \int fPx|P(\theta|D)d\theta \tag{4.12}$$

  where the posterior distribution can be down into:

$$P(\theta|D) = \frac{P(\theta,D)}{P(D)} \propto P(\theta,D) = P(\theta|\alpha)\prod_{x\in D} P(x|\theta) \tag{4.13}$$

- Marginal distribution of the data: it is denoted with P(D) and is obtained with integration over the parameter:

$$P(D) = \int P(D,\theta)d\theta = \int P(\theta|\alpha)\prod_{x_i\in D} P(y_i|\theta)d\theta \tag{4.14}$$

4.2.6 How: Gibbs Sampling

4.3 Probabilistic Topic Model

As explained in section 3.4.3 about Latent Semantic Analysis, there are different models that try to apply statistical methods on collections of texts to extract patterns useful to many applications, such as Information Retrieval issues, Sentiment analysis, Summarization, and Topics extraction. The latter is at the center of the analysis in this chapter.

First, it is important to define what a topic is. In fact, it is not trivial to say which are the keywords that better describe a text. For instance, the general topics discussed in an article on a newspaper are easily understandable by a person. Completely different is the discussion if the task has to be completed automatically. In this context, it is possible to use the hypothesis that a topic is represented not by a single keyword, but by a cluster of words found in the text analyzed.

The purpose of the program developed in SEB Bank was to create a summary of the opinion of the Board Member from the Riksbank from the minutes of the discussion in the board members meeting, the explanation of the dataset is in section 2. In the program developed during the internship, the topics used to create the summary of the opinion for

each Board Member were static. In other words, the keywords used in the summarization algorithm are decided by a human.

In this section are analyzed some probabilistic methods created to extract topics from collections of texts without knowledge added by a human. In this way, it is possible to create a model that automatically extract the topics from each text. In the next sections, first is introduced the Probabilistic Latent Semantic Analysis as the passage from matrix factorization to probabilistic model (pLSA). Then, it is explained the Latent Dirichlet Allocation Model as an extension of the pLSA.

The models exposed can be seen as a generative model, that is to say, that a probabilistic procedure is defined by the model to generate text. Thus, to create a document, first, it is chosen a distribution over topics, then for each term in that document, a random topic is sorted by the distribution chosen, and finally a word is drowned from the topic distribution. If this process is inverted by statistical techniques, then it is inferred the set of topics used in the generative process for the documents.[28]

4.3.1 From matrix factorization to probability: Probabilistic Topic Model

The first model introduced in this direction was proposed by Thomas Hoffmann and it is called Probabilistic Latent Semantic Analysis.[15] Like the name, it is a probabilistic model used to extend the Latent Semantic Indexing Model. To summarized the idea of LSA, already introduced in section 3.4.3, three points are given:[28]

- the terms-documents matrix $M \in \mathbb{R}^{Dxn}$, introduced with the Vector Space Model, is useful to derive semantic information of the collection and the documents;
- the sparsity of M can be seen as the noise of the data, consequently the dimensionality reduction is essential to obtain knowledge form M;
- thanks to the Vector Space Model documents and terms are represented as points in Euclidean space.

In the probabilistic models, the first two claims above are consistent, the third one is substituted by the idea of topics. In fact, in the models discussed in the next sections, the semantic properties of terms and documents are contained in probabilistic topics. The latter are a distribution over words, meanwhile, documents are seen as mixtures of topics.

The mixture models and the distributional topics over words take the place of the Linear Algebra Used in LSA to leave the place to Bayesian Statistic.

Mixture of Unigrams

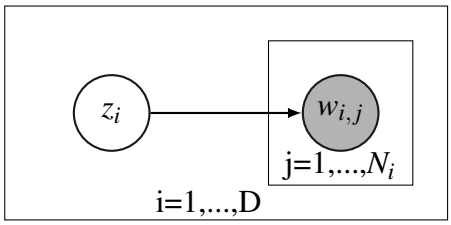

Fig. 4.3 Mixture of Unigrams

Probabilistic Semantic Analysis

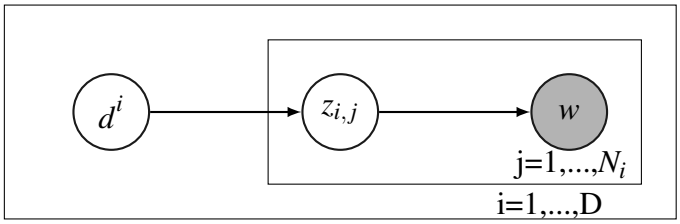

Fig. 4.4 Probabilistic Latent Semantic Analysis

4.3.2 Probabilistic Latent Semantic Analysis (pLSA)

The starting point for PLSA is a statistical model colled Mixture of Unigrams, introduced by Nigam. [23] In picture 4.4 is showed Bayesian network associated with the latter model. There are two key assumptions in this model:

- the terms-document data, viewed as pair $(d_i, w_{i,j})$, are produced by a mixture model;
- there is a 1-to-1 correspondence between topics and documents, in other words, each document d_i must have only one topic.

In this way, each document in the collection $d_i \in C$ is generated from a probability distribution. In particular, it is used a mixture of components $z_j \in Z = \{z_1, ..., z_k\}$ parametrized by some parameters θ. At this point, a document d_i is generated by first selecting $P(z_k|\theta)$, a topic prior probability, then ha

4.3.3 Latent Dirichlet Allocation

One of the improvement of the Probabilistic Latent Semantic Analysis is called Latent Dirichlet Allocation (LDA). The main difference is that there is an additional hypothesis on the distribution of the priors. However, before explaining the LDA as a probabilistic model,

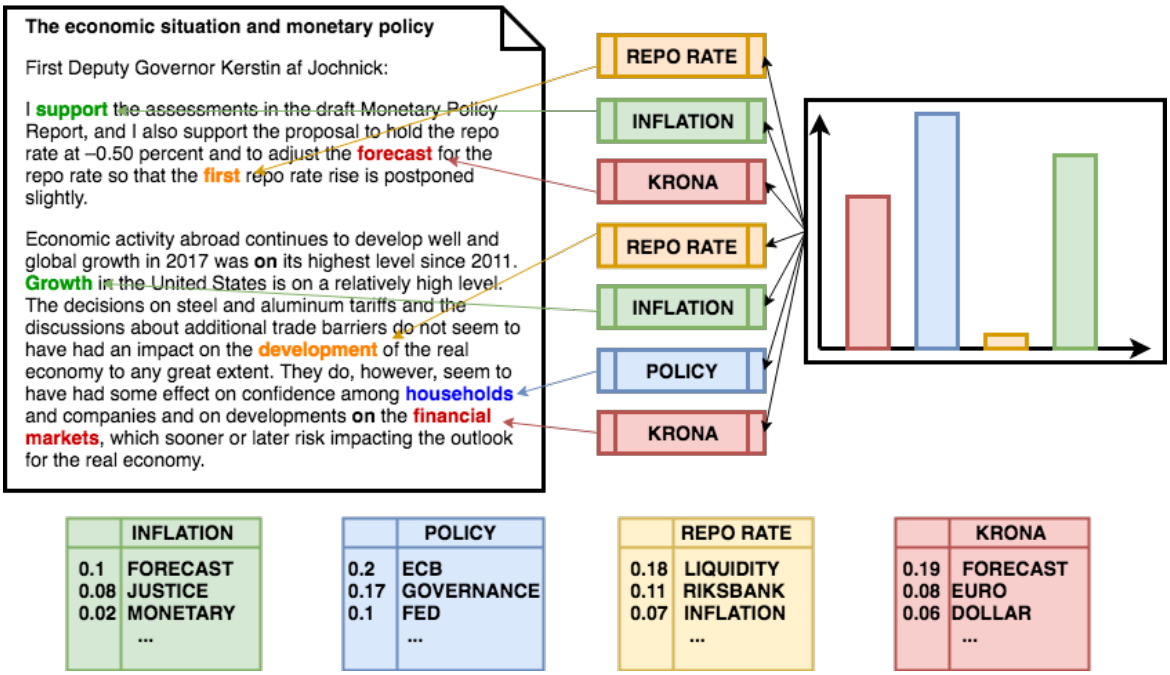

Fig. 4.5 LDA: visualization of the distribution involved.

it is useful to catch the intuition behind, therefore in the picture 4.5 are shown the main distribution in a graphical way.[6]

In the left side of picture 4.5, there is a excerpt of the Monetary Policy minutes of Riksbank from the meeting of April 2018. The board members of the Riksbank are used to speak about different themes during the meeting, the subjects of the meeting can change by time and are influenced by several factor not analyzed in this work. [8]

Some of the main themes discussed in this period according to the Research team of SEB Bank are: inflation, repo rate, krona, and ECB (European Central Bank). In the text showed in picture 4.5, some of the words are highlighted with different colors. For instance, "first" and "development" that are words related with topic "Repo rate" are highlighted in red. The idea in LDA is to link, in this visual representation by highlighting, all the words in a document based of the topic most related with them. Topics are distributions over the dictionary of the collection. Usually common words are stripped by the text before the analysis, as explained in section 3.1.

Ones the highlighting work is done much more information are available to classify and analyze the document evaluated. In fact, for each document a distribution over topics is

[8]For a good overview of the main economical backgrounds in the Monetary Policy meetings of the Riksbank, the Central Bank of Sweden provides some good papers at https://www.riksbank.se/en-gb/press-and-published/riksbanken-play/2013/what-is-monetary-policy/.

available just by counting the number of words related with each topic, as visualized in the right part of picture 4.5. Furthermore, the same work is done for all the documents in the collection analyzed. Thus, some information are retrieved also for the distribution of topics over words, as explained with the example in the down part of picture 4.5.

The latter is just an example of how ideally LDA works. However, the real results are not always so clear. In fact, the topics are not decided a priori by the user, only the number of topics is chosen. The distribution of the topics over words, the distribution of the topics for each document and the distribution of the topics over the entire collection of documents are inferred by the probabilistic model behind LDA.

There are two main ways to see LDA[28]:

- Generative process: a word is generated by a topic distribution over words and this distribution is taken by a distribution over topics for each document in the collection;
- Statistical inference: starting from a document in the collection, documents are a set of words observed, which topic model is the most likely to have generated the sequence of words observed?

To better explain how this process work, a good way to understated it is to start with the generative process, that it is to say how the model assume a documents, or better the sequence of words, is created. To do so, we have to assume that the topics are already created, however, as just said, this is not really true, but represent the knowledge conserved in the prior at each iteration.

The generative process wants to create documents as a sequence of words, hence, for each document $d_i \in D$ and for each word $w_{i,j}$ from the dictionary, the generative process to draw the word has two stages:

1. randomly choose a distribution over topics;
2. for each word $w_{i,j}$ in the document d_i:
    - randomly choose a topic from the distribution in step one;
    - randomly choose a word from the corresponding distribution of the topic over words.

The hypotheses that a document could show multiple topics, that the topic is chosen by a per-document distribution, and that each word in the document corresponds to only one topic is encoded in the process. Finally, the topics are shared by all the documents, but each document has its own distribution on this set of themes.[6]

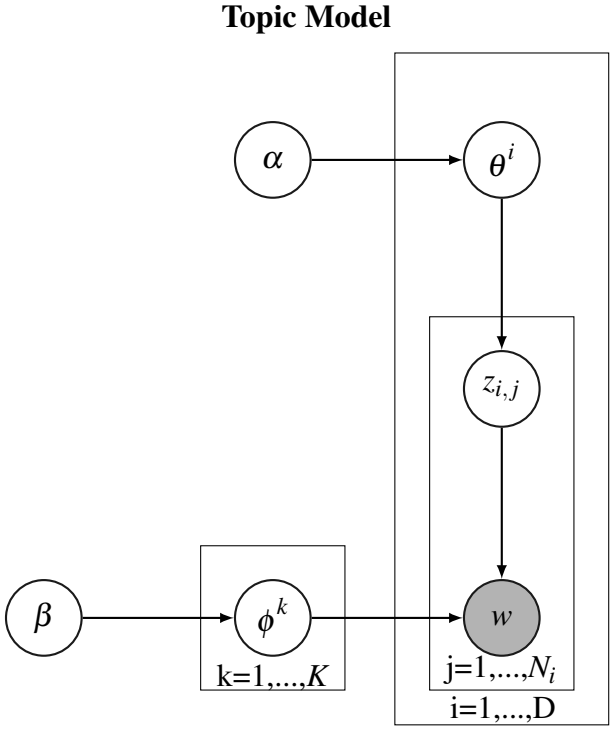

Fig. 4.6 Topic Model

In the statistical inference view, the hidden structure of the topic has to be learn from the observed variables that are the words. The hidden structure of the topics is represented in LDA by the distribution of topics per document, and the distribution per word-topic in each document. This view better reflect the reality of the model, in fact there are no information passed to the algorithm about topics, as remarked by one of the pioneer of LDA D. Blei: "The interpretable topic distributions arise by computing the hidden structure that likely generated the observed collection of documents."[6]

The graphical way to show LDA with a DAG is reflected in picture 4.7. There are three may levels:

- the collection level with the hyperparameters α and β,
- the topic level with ϕ^k, with $k \in K$, and the document level θ^i, with $d_i \in C$ and $|C| = D$;
- $z_{i,j}$ and $w_{i,j}$ that are at a word level in the dictionary DIC of terms considered, with $|DIC| = V$.

The random variable and parameters have different distributions that represents the previous assumptions:

- $\alpha \in \mathbb{R}^K$ is the vector of priors for the topics, $\alpha_K \in \mathbb{R}$ is the prior weight for topic k.
- $\beta \in \mathbb{R}^V$ is the vector of prior for the words, $\beta_j \in \mathbb{R}$ is the prior weight for word $w_j \in DIC$ for the topics distribution.
- $\phi_k \sim Dir(\beta)$ is the V-dimensional Dirichlet distribution of words for topic k, $\phi_{k,j}$ is the probability of word $w_j \in DIC$ occurring in topic k.
- $\theta_i \sim Dir(\alpha)$ is the k-dimensional distribution of topics for document $d_i \in D$, $\theta_{i,k}$ is the probability of topic k to be in document $d_i \in C$.
- $z_{i,j} \sim Multinoulli(\theta_i)$ is a Multinoulli distribution, represented with a k-dimensional that identify the topic for word $w_{i,j}$ in document $d_i \in C$.
- $w_{i,j} \sim Multinoulli(\phi_{z_{i,j}}$ is the Multinoulli distribution for the word j-th in document $d_i \in C$.

Notice that $z_{i,j} \in \mathbb{R}^K$ and $w_{i,j}\mathbb{R}^V$ are two vectors of the say type described in section 4.2.4, such that: they have all zero entry except for the entrance of the class that they are representing. Furthermore, it is important to remember the conjucate prior between Dirichlet distribution and the Multinoulli as shown in the formulas: 4.9 and 4.10.

The inference part is based on the update of the priors. In fact the generative process defines the joint probability distribution for the variables in the model, however, only $w_{i,j} \forall\, j \in N_i, \forall d_i \in C$. Therefore, the joint distribution is used to compute the conditional distribution of the hidden variables given the observed variables, to obtain the posterior distribution.[8]

The joint distribution of the hidden and observed variables is:

$$P(\Phi, \theta, Z, W) = \prod_{k=1}^{K} P(\phi_k) \prod_{i=1}^{N} P(\theta_j) \prod_{j=1}^{N_i} P(z_{i,j}|\theta_j) P(w_{i,j}|\phi_k, z_{i,j}) \tag{4.15}$$

With this notation, there are three ways of define the dependences encode in the LDA: with the statistical assumptions behind the generative process, with the joint distribution, and with the DAG model.

The posterior inference issue is faced in different ways in literatures. The main problem is the computation of the conditional distribution, the posterior, given the observed variables.

The formula to indicate this quantity is:

$$P(\Phi, \theta, Z|W) = \frac{P(\Phi, \theta, Z, W)}{P(W)} \tag{4.16}$$

The denominator quantity $P(W)$ is the marginal probability and it could be theoretically computed by summing the joint distribution over every possible value of the hidden structure.

However, the number of possible configurations makes exponentially intractable the computation of this value. This is a common problem in Bayesian Probability, many efficient method have been developed to face it. Two main categories exist in this context:

- Sampling-based algorithms: Gibbs Sampling is the most famous algorithm of this category, see section **??**.
- Variational Inference method: instead of samples, some parameters are optimized from a family of distributions over the hidden structure of topics, the optimum parameters are the closest with the posterior

4.3.4 Why LDA?

The model described in the previous section is a powerful tool to discover hidden patterns in the collection of documents analyzed. The three main distribution obtained from LDA are used to different purpose.

The distribution of topics over words is used to create clusters of words, picture **??** shows in the lower part these distributions. Furthermore, if for each cluster of words are extracted the most frequent, by the context of the words is possible to manually label these cluster that are the final topics of the document.

Ones topics are created, it is possible to use the distribution of the topics per document to understand which are the most relevant topics for each document. Usually, it is used a for both the priors hyperparameters α and β of the prior Dirichlet with a value < 1, it is called sparse Dirichlet prior and try to model the intuition that only few topics are relevant for each document and only few words are relevant for each topics.[8]

These cluster of words created and labeled are used in many approach such as: Information Retrieval [20] and to discover big amount of documents.

Since the release of the first Topic Model, between 1999 with the pLSA by Hofmann[15] and 2003 with LDA by D. Blei[8], many other models have been proposed. Author-Topic Model (ATM) was proposed in 2004, to model the relation between topics and authors of the documents another a Multinomial distribution is added. The knowledge contained in main dictionary and lexicons available on line could be used to improve how the words are related

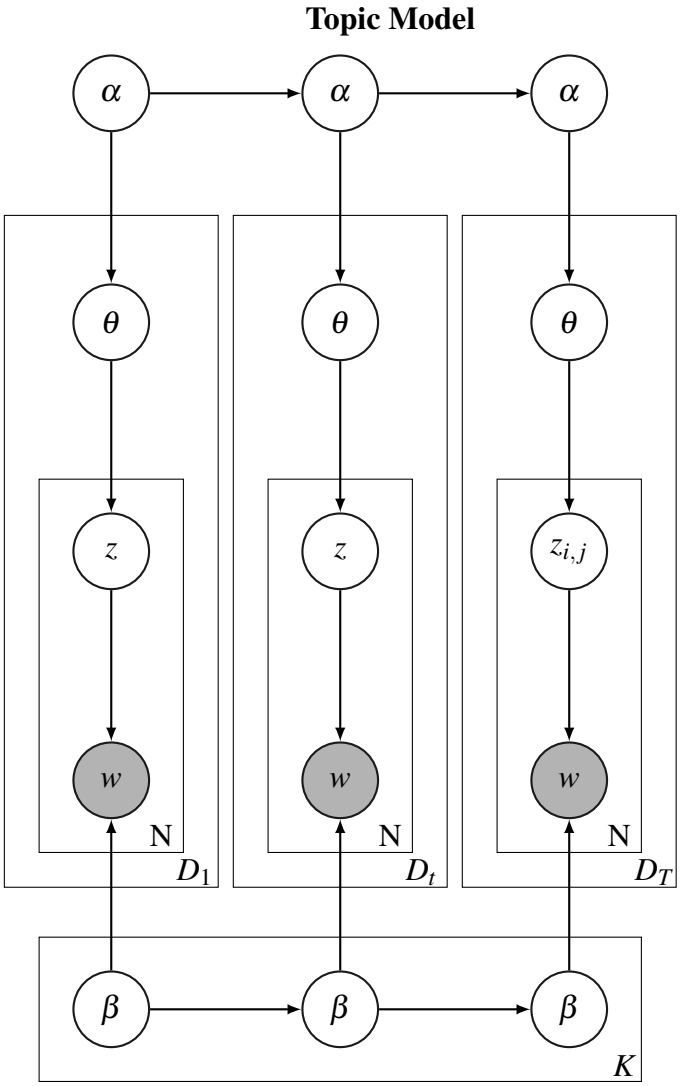

Fig. 4.7 Topic Model

in the model, this type of model are known as Knowledge-based Topic models.[19][4] Time is another important variable that can be used to model the evolution of the topics over time, models with this purpose are named Dynamic Topic Models (DTM)

4.3.5 Dynamic Topic Model

The main idea in DTM is to divide the corpus of docs in slices (years for instance), then to assume that each slice's docs is exchangeable and so draw from a LDA model, finally, let the topic distribution evolve from slice to slice. [7] The documents are divided in slices of times $t \in T$, such that: $D = D_1, ..., D_T$ is a partition of D.

Mixture of Unigrams

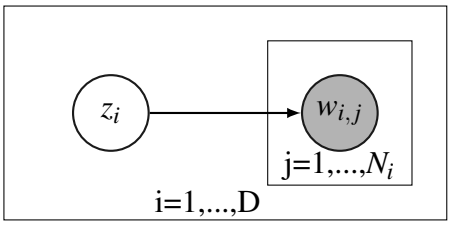

Fig. 4.8 Mixture of Unigrams

The evolution over time of topics is modeled by a logistic normal distribution evolving over time[2] and a state-space model on the natural parameter of the topic multinomial[31]. The final result is an additional distribution over the hyperparameters that chains the parameter for each slice of time $t \in T$ with a gaussian noise, as showed in the graphical model and in the following equations:

$$\begin{aligned} \beta_{t,k}|\beta_{t-1,k} &\sim N(\beta_{t-1,k}, \sigma^2 I) \\ \alpha_{t,k}|\alpha_{t-1,k} &\sim N(\alpha_{t-1,k}, \eta^2 I) \end{aligned} \tag{4.17}$$

4.3.6 Coding with LDA

Some of the most used Libraries to use LDA and others topic models are listed below:

- LDA in R with cran.r-project.org
- LDA in Python:
  - scikit-learn;
  - GenSim;
  - Mallet.

Two tutorial have been used for the analysis of the Topics in the dataset of the Minutes:

- machinelearningplus.com: it is a good tutorial to implement in LDA in Python with Scikit Learn.
- analyticsvidhya.com: there are some implementation of topic models in Python with the library GenSim.

Unigrams

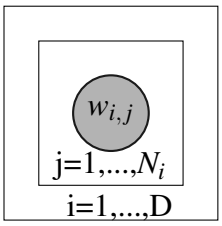

Fig. 4.9 Unigrams

Chapter 5

Results

5.1 Summarization

5.2 Sentiment Analysis

5.3 Topic Model

The dataset analyzed with the Topic model is the collection of Minutes from 2012 to 2018 described in section 2.1. This dataset could be viewed at two different level: board member level and minutes level. With the first method, it is considered for each minutes the part of text related to each board member, therefore for each minutes there are 6 parts of text one for each board member.

By a less zoomed view, it is possible to consider the minutes as a single unit of text. There are 38 minutes in total, every year the Riksbank has an average of six meeting with all the board members to discuss Monetary Policy decisions. Hence, since at the BM-level are investigated 6 part of text for each minutes, there are 228 unit of text in total.

The two level of analysis are justified by the fact that the topics analysis by the LDA gives informations for each document analyzed over the topics and for the set of topics over the words in the collection. In this respect, the Minutes-level analysis could be thought as a topic analysis for each minutes and so for each time period. On the other end, the BM-level analysis has the intuition of which topics are more relevant for each Board Member.

There are 4148 unique words in the dictionary *DIC* of the collection *C* od documents.

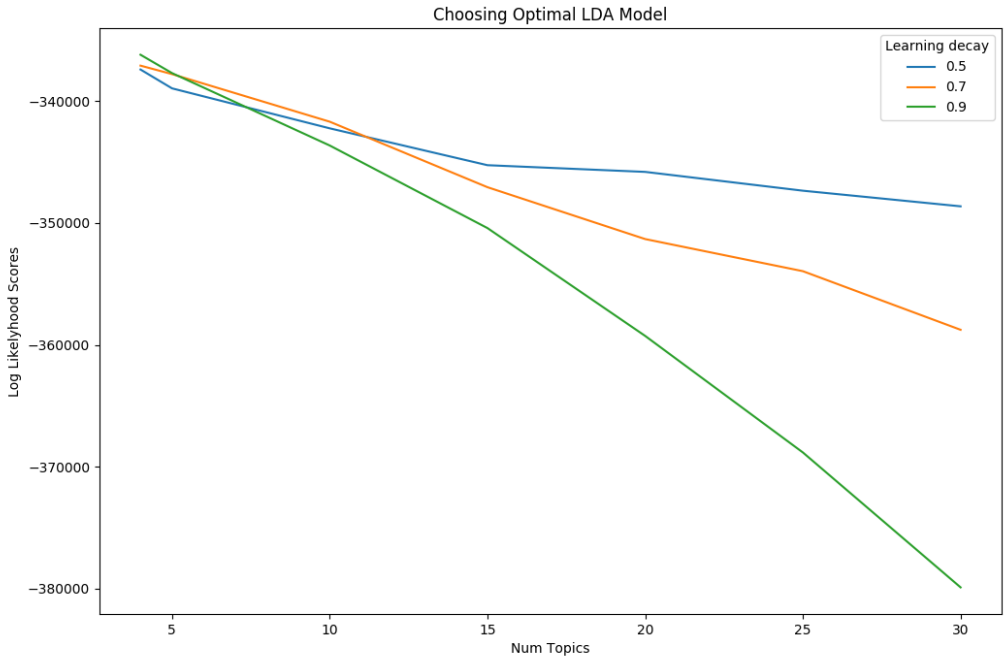

Fig. 5.1 LDA: loglikelihood and learning rate.

5.3.1 LDA at BM-level

- Stop-words cleaning and threshold of 10 for each word;
- Matrix: 1237x226 sparse matrix;
- Sparsity: 23.13% is the percentage of non-zero elements in the matrix;

5.3.2 LDA at Minutes-level

- Stop-words cleaning and threshold of 10 for each word;
- Matrix: 1237x38 sparse matrix;
- Sparsity: 68.27% is the percentage of non-zero elements in the matrix;

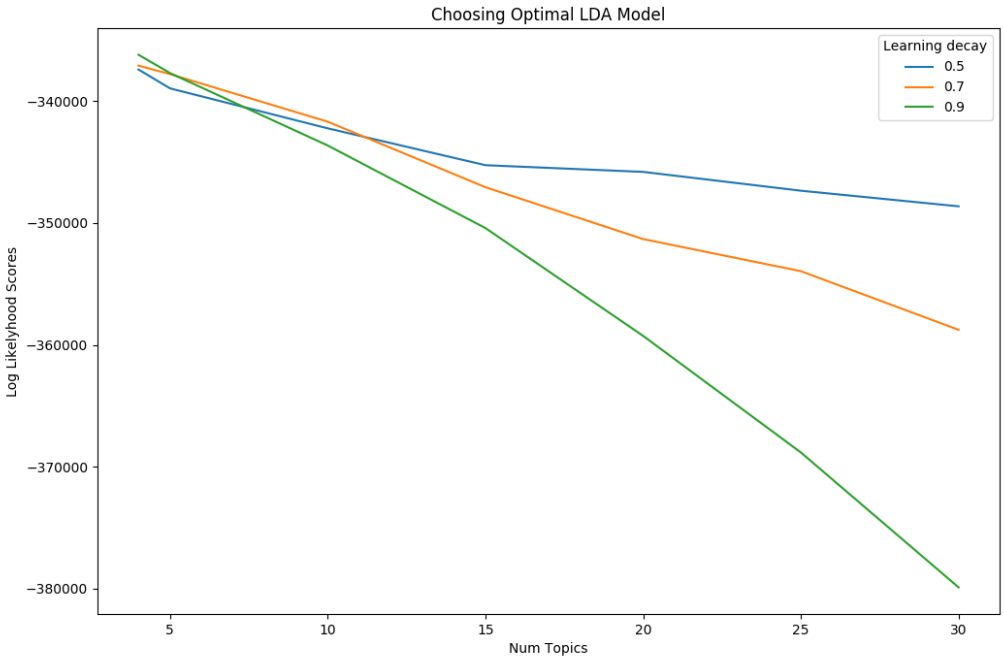

Fig. 5.2 LDA: loglikelihood and learning rate.

5.4 Conclusions

Latent Dirichlet allocation was applied on the Minutes released from Riksbank, the minutes were divided in two dataset: BM-level and Minutes-level, as explained in the previous section. The LDA provides two main distributions: Topics over text and topics over words.

References

[INT] Intro bn.

[2] Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177.

[3] Allahyari, M., Pouriyeh, S., Asse, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). Text summarization techniques: A brief survey. *In Proceedings of arXiv*, (1):9 pages.

[4] Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. pages 25–32.

[5] BERRY, M. W., DUMAIS, S. T., and O'BRIEN, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*.

[6] Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

[7] Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. pages 113–120.

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[9] CVE-2008-1368 (2008). Monetary policy. [online] https://www.investopedia.com/terms/m/monetarypolicy.asp.

[10] Das, B. (2004). Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem. *CoRR*, cs.AI/0411034.

[11] Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. pages 19–25.

[12] Gurusamy, V. and Kannan, S. (2014). Preprocessing techniques for text mining.

[13] Harris and Zellig (1954). Distributional structure. 10:146–162.

[14] Heckerman, D. (2008). *A Tutorial on Learning with Bayesian Networks*, pages 33–82. Springer Berlin Heidelberg, Berlin, Heidelberg.

[15] Hofmann, T. (1999). Probabilistic latent semantic indexing. pages 50–57.

[16] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). An introduction to statistical learning—with applications in r. 82.

[17] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

[18] Kahveci, E. and Odabaş, A. (2016). Central banks' communication strategy and content analysis of monetary policy statements: The case of fed, ecb and cbrt. *Procedia - Social and Behavioral Sciences*, 235:618 – 629. 12th International Strategic Management Conference, ISMC 2016, 28-30 October 2016, Antalya, Turkey.

[19] Li, F., He, T., Tu, X., and Hu, X. (2012). Incorporating word correlation into tag-topic model for semantic knowledge acquisition. pages 1622–1626.

[20] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[21] McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification.

[22] Monegato, G. (1998). Fondamenti di calcolo numerico.

[23] Nigam, K., Mccallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134.

[24] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. (1999-66). Previous number = SIDL-WP-1999-0120.

[25] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. *CoRR*, cs.CL/0205070.

[26] Sarkar, D. (2016). *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*. Springer Science+Business Media New York, Bangalore, Karnataka India, 1st edition.

[27] Steinberger, J. and Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proceedings of ISIM'04*, pages 93–100.

[28] Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models in handbook of latent semantic analysis.

[29] Tromp, E. (2011). Multilingual sentiment analysis on social media.

[Tu] Tu, S. The dirichlet-multinomial and dirichlet-categorical models for bayesian inference.

[31] West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models (2Nd Ed.)*. Springer-Verlag, Berlin, Heidelberg.

[32] Yang, C. C., Chen, H., and Hong, K. (2003). Visualization of large category map for internet browsing. *Decision Support Systems*, 35(1):89 – 102. Web Retrieval and Mining.

[33] Zhao, J., Liu, K., and Xu, L. (2016). Sentiment analysis: Mining opinions, sentiments, and emotions bing liu (university of illinois at chicago) cambridge university press, 2015 isbn 9781107017894. 42:1–4.

Appendix A

Singular Value Decomposition[22]

Theorem 1. *Given* $A \in \mathbb{R}^{mxn}$ *exist two orthogonal matrix* $U = \{u_1, \dots u_m\} \in \mathbb{R}^{mxm}$ *and* $V = \{v_1, \dots v_n\} \in \mathbb{R}^{nxn}$*, such that:*

$$U^T AV = S, \qquad A = USV^T$$

where $S \in \mathbb{R}^{mxn}$ *is diagonal, so that:*

$$S_{i,j} = \begin{cases} 0, & i \neq j \\ \sigma_i, & i = j \end{cases}$$

with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0,\ p = min\{m, n\}$.

There are many different algorithms to compute U and V, they are no unique. The matrix S has the singular value of A, that are the square root of the eigenvalues of $A^T A$. The columns of U and V are respectively the left singular vector and the right singular vector, so that $\forall i = 1, \dots p$:

$$Av_i = \sigma_i u_i$$

$$Au_i = \sigma_i v_i. \quad \text{(A.1)}$$

The geometrical meaning of the singular values of A is related with the hyper-ellipsoid and represents

$$E = \{y : y = Ax, ||x||_2 = 1\}$$

To better understand the importance of SVD as a low rank approximation, or mapping in a lower dimensional space of the original space, the following theorem is stated:

Theorem 2. *If $\exists r \in \mathbb{N}$, such that:*

$$\sigma_1 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} > \cdots > \sigma_p 0, \qquad p < r$$

then:

1. $rk(A) = r$;
2. $\{u_1, \ldots, u_r\}$ *is a base for* $R(A)$;
3. $\{v_{r+1}, \ldots, v_n\}$ *is a base for* $N(A)$;
4. $$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U_r S_r V_r^T;$$
5. $||A||_2 = \sigma_1$

where:

$$U_r = (u_1, \ldots, u_r), \quad V_r = (v_i, \ldots, v_r);$$

and $N(A)$ *is the core of the transformation and* $R(A)$ *the image space.*

The next theorem it is useful to understand the relation between the space associated with the original matrix and the space of the low-rank approximation.

Theorem 3. *Given a Singular Value Decomposition of* $A \in \mathbb{R}^{mxn}$*, with* $rk(A) = r$*. If, it is fixed* $k < r$*, it is defined:*

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

and

$$\beta = \{B \in \mathbb{R}^{mxn} : rk(B) = k\}$$

then the following results are true:

$$min_{B \in \beta} ||A - B||_2 = ||A - A_k||_2 = \sigma_{k+1}$$

The last theorem gives a measure for the distance in 2-norm between the original matrix A and β the set of matrix with rank k. Furthermore, A_k is the best approximation for A with rank k.

Appendix B

PageRank