Exploratory Data Analysis of Videos Shared on Social Media Platforms

POLITECNICO
MILANO 1863

Michele Lunetti

Student Id: 898819

Advisor: Prof. Marco Brambilla

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

This thesis is submitted for the degree of
Master of Science in Computer Science and Engineering

June 2021

To Klaudia

Ringraziamenti

Innanzitutto vorrei ringraziare il Professor Marco Brambilla, del Dipartimento di Elettronica Informazione e Bioingegneria del Politecnico di Milano, per il supporto nella preparazione e nella ricerca finalizzata alla stesura della tesi.

Ci tengo a esprimere la mia gratitudine ai miei genitori e ai miei fratelli che mi hanno supportato e sopportato con fiducia in questi anni di studio, sia a distanza che in lockdown a casa, in qualsiasi condizione.

Un grazie enorme anche alla mia ragazza per il suo fondamentale sostegno e affetto.

Infine vorrei ringraziare i miei fantastici compagni di studio: Lori, Fab, Ale, Harry, Konso, Dani, Ivo, Bat, Ceru, Smalt, Abt, Carlo, Serj, Simo, Mirko, Phil, Lore e Buschi. Avete reso questi anni i più belli della mia vita.

Questa esperienza ha segnato il mio vissuto e la porterò sempre nel profondo del cuore.

Abstract

Social Media nowadays plays a more relevant than ever role in our lives. Its presence has such a widespread reach that it has assumed a key relevance; not only for the single end-users, but also for businesses and institutions, which desire to uncap the potential of the data generated from these communication platforms. This great amount of data, while almost useless if not processed, can be turned into valuable information. This information, not only about the users behaviour, but also about the context in which they live, gives us a quick picture of the ever changing social network.

Over the years many techniques have been proposed and developed in order to extract such information, and the fields of application for such techniques are growing steadily.

The purpose of this thesis is to extract, through the use of exploratory data analysis and natural language processing techniques, the main topics that can describe the trending videos on the Social Media platform YouTube. This platform is not only one of the most used worldwide, but it has also an open nature with well documented libraries and fully-featured API support.

In this scenario we analyse a span of six months of trending videos: using the metadata of such videos from nine distinct countries, in order to describe the different use of the Social Network across different nations. We will then delve into the analysis of the closed captions of the videos from three English speaking countries, in order to extract the trending topics on YouTube, and analyze their relationship with the popularity of their videos.

Abstract

I Social Media in questi giorni hanno raggiunto un ruolo quanto mai rilevante nelle nostre vite. La loro presenza è ormai così capillare da aver assunto un'importanza chiave; non solo per i singoli utenti finali, ma anche per le imprese e le istituzioni, che desiderano sbloccare il potenziale dei dati generati da queste piattaforme di comunicazione. Questo grande ammontare di dati, sebbene pressochè inutile se non processato, può essere trasformato in preziose informazioni, non solo riguardo il comportamento degli utenti coinvolti ma anche riguardo il contesto in cui vivono, producendo un'istantanea dei social networks che cambiano ogni giorno.

Nel corso degli anni sono state proposte e raffinate molte tecniche al fine di estrarre queste informazioni e i campi applicativi per queste tecniche sono in costante crescita.

Lo scopo di questa tesi è di estrarre, tramite l'utilizzo di tecniche di analisi esplorativa dei dati e di elaborazione del linguaggio naturale, gli argomenti principali che possono descrivere i video di tendenza sulla piattaforma Social Media di YouTube. Questa piattaforma non solo è una delle più usate al mondo ma presenta anche una natura aperta allo studio con librerie ben documentate e un completo supporto API.

In questo scenario analizziamo un arco temporale di sei mesi di video di tendenza: usando i metadati di questi video da nove paesi distinti, in modo da descrivere il diverso utilizzo del Social Network tra le varie nazioni; quindi approfondiremo l'analisi dei sottotitoli dei video da tre nazioni anglofone al fine di estrarre gli argomenti di tendenza su YouTube e la loro relazione con la popolarità dei video.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and problem statement

Social Media Platforms are already deeply embedded in our daily lives, currently the average amount of time that internet users aged 16 to 64 spend using social media each day is, in hours and minutes, 2:25 worldwide, 1:52 in Italy.[1] People rely on them for very different needs, ranging from daily news and updates on critical events to entertainment, connecting with family and friends, reviews and recommendations on products/services and places, fulfilment of emotional needs, workplace management, to name just a few.

These Digital Media are socio-technical systems that produce and enable inscriptions of individual and collective actions, the enormous amount of information generated by their users are providing the social sciences with quantities of information that are comparable to those collected in natural science laboratories, but the quality of such traces is radically different.

Academics and practitioners have explored and examined the many sides of social media over the past years, it is clear that this data can be leveraged and the researches on the user-generated content is highly influential in a myriad of settings, from purchasing/selling behaviours, entrepreneurship, political issues, to venture capitalism [7].

One of the most used and famous social media platform is YouTube. Since its launch in 2005, YouTube has grown from a repository of amateur videos into the biggest online video platform worldwide. Featuring a wide variety of corporate and user-generated content that ranges from music and gaming videos to DIY's and educational clips, the video giant is now a leading online destination for millions of users from around the world.

[1] Social flagship report Q3 2020 - GlobalWebIndex

Currently, the video sharing platform is the second most popular social media platform with over 1.9 billion users worldwide; in the United States, YouTube saw a market reach of around 90 percent in 2018, and its mobile versions are enjoying similar success globally. [2]

With the many use cases arising in this web science context, the two constraints of information quantity and information quality render the process of knowledge extraction and data analysis quite challenging.

In this thesis we will focus on the implementation of exploratory data analysis, an approach to analyzing data sets to summarize their main characteristics, and dive deeper by implementing Topic Modelling algorithms for discovering the abstract "topics" that occur in a collection of documents.

Moreover we will focus on the analysis of YouTube trending videos, a daily category of videos proposed by a not fully disclosed YouTube algorithm that takes in considerations factors such as: number of views, likes, comments, where the video is coming from and age of the video; although this ranking algorithm is not fully disclosed, we can consider this collection a good sample of the videos with most user interactions.

1.2 Proposed solution

In this thesis we will analyse most trending content on the social media platform YouTube, starting from exploratory analysis on the metadata of the videos from both a spatial and a temporal point of view: in this phase we will analyse the categories, length, views, likes, number of comments, titles, and descriptions in order to uncover statistical correlation and insights between these parameters.

In the second part of the thesis we will analyse a specific dimension of the media content in order to deepen our exploratory analysis: we will extract the closed captions of the trending videos from three specific countries (United States of America, Great Britain and Canada) which all share the common English language, and apply Natural Language Processing techniques in order to uncover the abstract 'topic' which are dominant on this social network.

The statistical model used in this process is know in literature as Topic Modelling, we will focus on the implementation and comparison of Latent Dirichlet Allocation (LDA) and LDA with Fast Collapsed Gibbs Sampling (MALLET), and see if the results obtained are useful in our exploratory context.

Our approach collects the closed captions of each whole video, in order to generate a single document for each of them. These documents are processed with both classic text

[2] U.S. user reach of leading video platforms 2018 Published by H. Tankovska, Jan 26, 2021

mining operations and further procedures specific for this data set. The corpus of documents is then fed to the LDA algorithm, which will classify them in different topics.

This new representation allows us to compare the videos based on the different topics they are grouped by the LDA, and to get useful insight on the main topic relevant on the whole social platform , allowing us to monitor what is happening and gain valuable insights on the whole social network.

Once we obtain these topic vectors for each video, we test the implementation of a variety of regression models in order to predict the number of views of each video starting from the topics extracted in the previous steps.

The main concern for this dimension is the quality of the closed captions extracted, the progress in speech to text recognition in the last few years has been staggering, but for what concerns the accuracy of YouTube for the automatic closed captions in their videos it is still around a value of 0.7 , as reported in [12]. Another concern is the granularity of the categories of videos investigated; as we will see, certain categories of videos such as "Nonprofits & Activism" may present richer content per document, but the scarcity of trending videos for such categories will preclude our analysis on these sets. A less strict criteria of selection for the videos to analyse, supported by the pipeline necessary to extract such data, could easily be used in order to extend the research on more in depth use cases, such as for example in humanitarian, journalistic or activist contexts.

1.3 Structure of the thesis

The structure of the thesis is as follows:

- Chapter 2 defines and explains the background knowledge and concepts that are related to the work that has been performed for this thesis.
- Chapter 3 presents an overview on the past works that are related to this thesis, the problem they try to answer and the solution they propose.
- Chapter 4 contains a high level description of the employed methods that are used in this thesis.
- Chapter 5 describes the source codes and implementations of the used methods.
- Chapter 6 presents the results of the experiments and discusses these outcomes.
- Chapter 7 concludes this report by summarizing the work, doing a critical discussion and proposing the possible future related work.

Chapter 2

Background

This chapter presents the theoretical background of the thesis, it contains the references to concepts, models and techniques on which the writing of this work is based.

2.1 Web Data Extraction and Information Retrieval

Web Data Extraction is an important problem that has been studied by means of different scientific tools and in a broad range of applications. Many approaches to extracting data from the Web have been designed to solve specific problems and operate in ad-hoc domains. In our specific case we collected data from the YouTube social media platform through the use of the following two specific technologies.[5]

2.1.1 YouTube Data API

An Application Programming Interface (API), is a computing interface that defines interactions between multiple software applications or mixed hardware-software intermediaries, it is used by a steadily growing number of companies for exposing resources and services online. YouTube proposes different APIs for different kind services, in our framework we used the YouTube Data API v3, which allows developers to access video statistics and YouTube channels data via two types of calls, REST and XML-RPC.

The YouTube Data API allows to incorporate functions normally executed on the YouTube website into the custom built framework. Between the different types of resources that can be retrieved using the API there are the video id, title, description, statistics, and many others. The API also supports methods to insert, update, or delete many of these resources[1].

[1] https://developers.google.com/youtube/v3/docs

Although the YouTube API is freely accessible, and most of the data we are going to use is retrievable through the methods exposed by this interface, it is subject to limitations in its usage. The collection of data is therefore limited, especially in the query phase of the list of trending videos, this concerns leads us to the next technology.

2.1.2 Web Data Extraction

An alternative to the platform API is to use a python script in order to automatically extract from the web pages the information we want to ingest in our data analysis pipeline. This process is commonly known as Web Scraping[16]. Between the common processes need to perform this activity, we can identify the following fundamental ones:

- Query a web server
- Request the data, usually in the form of HTML or other files that compose the web page
- Parse the data in order to extract the necessary information

Between the most common libraries used for this purpose the most widespread adopted are BeautifulSoup[2], Scrapy[3], and Selenium[4]; our choice has fallen on this last one, mainly due to the ease of use and complete documentation.

[2]https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[3]https://docs.scrapy.org/en/latest/
[4]https://www.selenium.dev/

2.2 Data Mining

Data mining refers to the process of searching hidden information from a large number of data through algorithms. Due to the focus of our thesis on exploratory data analysis on closed captions, this section will mostly introduce Natural Language Processing algorithms and Text mining techniques.

2.2.1 Bag of Words

A common text data representation method in Natural Language Processing is Bag-of-Words (BoW)[13]. A BoW is a simple text vectorization model where a collection of documents (a.k.a. corpus) are tokenized and reduced to just a bunch of words (i.e. a bag of words), in which neither context, nor order placement nor grammar are taken in consideration, but only the count of occurrences of each word in each document.

It is a common good practice to apply this transformation only after a pre-processing pipeline of the raw data which usually involve activities[11] such as :

- Tokenization: breaking down a stream of text into words, phrases, symbols, or other meaningful elements called tokens
- Filtering: removal of high frequency words which are not relevant for current analysis context (a.k.a. stop words)
- Lemmization: reduction of words to their normalized form[21]

The value associated to the BoW may be different from the simple term frequency: for example, it can be binary (1 if the word is present, 0 otherwise), it can be a weight as the frequency with which each word appears in a document out of all the words in the document and many others.

2.2.2 Vector space model

Vector space model is an algebraic model used to represent text documents into a multidimensional space, in the form of vectors. Each of the dimensions is a term present in the documents and each document is a vector oriented in this space, according to the words from which it is formed. These vectors can be called Feature Vectors, in other words, vectors whose elements represent numeric or symbolic characteristics, called features. Each dimension corresponds to a term (see Figure 2.1). If a term occurs in the document, its value in the vector is set according to the chosen weight measure [2.2.1].

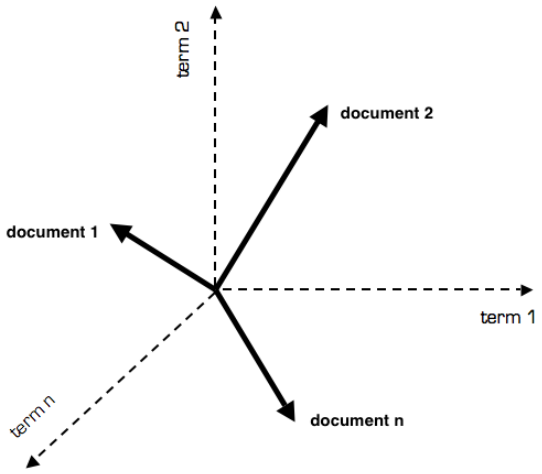

Figure 2.1 Vector space model

2.2.3 Differences between supervised and unsupervised tasks

An important characteristic to be highlighted in data mining is the difference between supervised or unsupervised tasks. In supervised ones, both the input and the expected output are known in advance: a typical example is classification, in which we know the data but also the label and we want to build a model that is able to assign the correct label to as many instances as possible. On the other hand, in unsupervised tasks, only the input is known, so the goal is to find natural patterns and structures among the data: clustering algorithms discover latent features that are used to group the data, but the belonging to a certain cluster is not known beforehand.

2.3 Topic modeling

Among the text mining tasks, topic modeling is one of the most popular, the main idea of topic modeling[8] is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words.

In simple terms, the algorithm scans the documents in the corpus, examines the word frequency co-occurrence and automatically learns groups of words that best characterize those documents, defined as topics. The main implications of topic modeling relies on the assumption that words which frequently appear together in a text, belong to the same topic.

This is the intuition which allows a document to be seen as a mixture of different topics and a topic as a collection of words [17].

Topic modeling is an unsupervised classification method, which means that we don't know in advance what we're looking for, instead is the algorithm that discovers and reports groups of terms for us. This implies an intrinsic difficulty in the evaluation of topic models since there is no "labeled" data about the topics discovered.

2.3.1 Latent Dirichlet Allocation(LDA)

One of the most common topic modeling methods is Latent Dirichlet Allocation (LDA). It is a generative probabilistic model for collections of textual documents.

LDA is a three-level hierarchical Bayesian model, in which each item of a collection, i.e. a document, is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In this way, the topic probabilities provide an explicit representation of a document.[2]

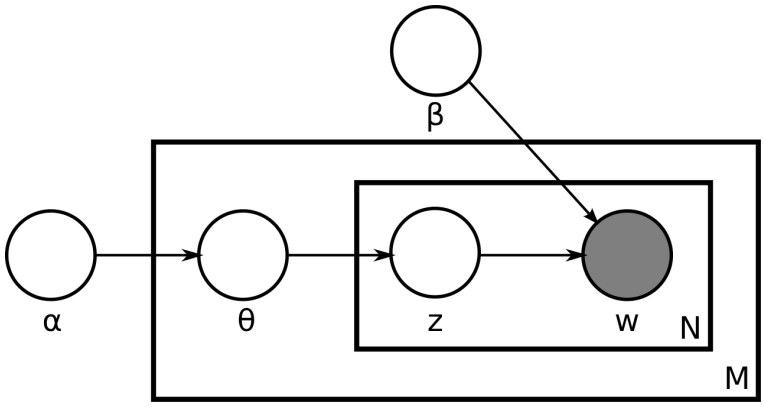

Figure 2.2 Plate representation of the LDA model

The graphical representation of LDA can be observed in Figure 2.2. The outer plate represents the M documents, while the inner one represents the repeated choice of topics and words within a document, with N as the number of words in a document.

- α is the parameter of the Dirichlet prior on the per-document topic distributions,
- β is the parameter of the Dirichlet prior on the per-topic word distribution,
- θ_m is the topic distribution for document m,
- φ_k is the word distribution for topic k,
- z_{mn} is the topic for the n-th word in document m

- w_{mn} is the specific word. It is grayed out, because it is the only observable variable in the system while the others are latent.

LDA is a parametric model, which means that the number of topics has to be provided as a parameter. For this reason, one of the most challenging tasks is to find the best number of topics when creating the LDA model.

There are many different variations of LDA, and the better performance of one model over the other is seldom due to the different use cases and context analysed[27]. Apart from the model proposed by Blei et Al.[2], Gensim standard LDA is based on the Online Learningfor Latent Dirichlet Allocation algorithm by Hoffman, Blei et al.[9], and Latent DirichletAllocation with Fast Collapsed Gibbs Sampling (also known as Mallet[14][28]) are between the most adopted. Mallet (MAchineLearning for LanguagE Toolkit) is a a Java-based console applicaction for language processing, document classification, clustering, topic modeling, etc. It uses Collapsed Gibbs Sampling, Pachinko Allocation and Hierarchical Latent Dirichlet Allocation for topic modelling[20].

2.3.2 Perplexity

One of the most common way to evaluate topic modeling is computing the perplexity of a held-out test set. This is achieved by splitting the available data into two parts: a training and a test set. For LDA, a test set is a collection of unseen documents w_d, and the model is described by the topic matrix Φ and the hyperparameter α for topic-distribution of documents.

The log-likelihood is computed in this way:

$$\mathcal{L}(w) = \log p(w|\Phi,\alpha) = \sum_d \log p(w_d|\Phi,\alpha)$$

of a set of unseen documents w_d given the topics Φ and the hyperparameter α for the topic-distribution θ_d of the documents. Likelihood of unseen documents can be used to compare models: higher likelihood implies a better model.

Then, the perplexity is

$$perplexity(test\ set\ w) = \exp\left\{-\frac{\mathcal{L}(w)}{count\ of\ tokens}\right\}$$

which is a decreasing function of the log-likelihood $\mathcal{L}(w)$ of the unseen documents w_d; therefore, the lower the perplexity, the better the model.

However, it has been proven with an experiment that perplexity and human comprehension of the found topics are often not correlated and that sometimes perplexity could even be

misleading [3]. Therefore, we are now going to introduce other measures to evaluate the topic models.

2.3.3 Topic coherence

Among the other measures that have been introduced, in addition to perplexity, to try to fill the gap between perplexity scores and human comprehension, there is topic coherence. It is defined as the average or median of pairwise word similarities, formed by top words of a given topic [23].

These methods are divided into two categories: intrinsic ones, that do not use any external source or task from the dataset, and extrinsic ones, which use the discovered topics for external tasks or external statistics to evaluate topics.

Two of the most known topic coherence measures are the intrinsic measure UMass and the extrinsic measure UCI, both computed as the sum of pairwise scores on the words $w_1, ..., w_n$ used to describe the topic, usually the top n words by frequency $p(w|k)$.

$$Coherence = \sum_{i<j} score(w_i, w_j)$$

- UCI measure [18]: it uses the Pointwise Mutual Information (PMI) as pairwise score function, defined as follows

  $$score_{UCI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

  where:

  - $p(w_i)$ represents the probability of seeing the word w_i in a random document,
  - $p(w_i, w_j)$ the probability of seeing both word w_i and w_j in the same random document

  It is extrinsic because the frequencies of when words co-occur are computed over an external corpus; so, for example, if we chose Wikipedia as external source, we would have:

  $$p(w_i) = \frac{D_{Wikipedia}(w_i)}{D_{Wikipedia}}$$

  where $D_{Wikipedia}(w_i)$ is the count of documents of the Wikipedia corpus containing the word w_i.

- UMass measure [15]: it uses a non-symmetric pairwise score function

$$score_{UMass}(w_i, w_j) = \log \frac{D(w_i, w_j) + 1}{D(w_i)}$$

  where $D(w_i, w_j)$ is the count of documents containing both word w_i and w_j.

  It is intrinsic because it compares a word with only the previous and following words, i.e. words within the corpus.

In addition to these two, another coherence measure has been introduced: CV [22], that is a new combination, which mixes the indirect cosine measure with the NPMI (normalized pointwise mutual information) and the boolean sliding window. It is generally the easiest to interpret and therefore chosen to evaluate LDA models.

2.4 Supervised Learning

Supervised Learning is a well known machine learning task whose objective is to learn the function that maps an input to an output given some labeled training data. In this work, different Supervised Learning methods have been used in order to resolve a regression problem. Below we introduce the main Regression models evaluated during this work:

- **LinearRegression**: LinearRegression fits a linear model with coefficients w = (w1, . . . , wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.
- **Lasso**: Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean.
- **Ridge**: A regressor which resolves the ridge equation by the method of normal equations.
- **PolynomialFeatures**: Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the specified degree.
- **KNeighborsRegressor**: Regression based on k-nearest neighbors. The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.
- **DecisionTreeRegressor**: A 1D regression with decision tree. The decision trees is used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve.

- **RandomForestRegressor**: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

The following models are implemented using the Scikit-Learn library, for further details about the code implemented please refer to their well documented library[5].

[5]https://scikit-learn.org/stable/supervised*learning.html*

Chapter 3

Related Work

In this chapter we discuss the existing works in the literature that have been the basis or have influenced the making of this thesis. A lot of studying has been done in topic modeling and social media analysis, we mention some of the articles found during the research phase in both the directions.

3.1 Topic modeling

Some of the most influential articles, which leaded us to the implementation of Topic modelling using Latent Dirichlet Allocation, have already been cited in chapter 2; one of the most significant papers is the Latent Dirichlet Allocation [2], that lays the foundations of all the related works and available variations, his works on the subject improved with the years and in 2010 published a paper describing Online Learning for Latent Dirichlet Allocation algorithm[9] by Hoffman, Blei et al. This algorithm became the standard on which Gensim implemented LDA and provided the most important article foundamental for this work.

Similarly with the Mallet (MAchine Learning for LanguagE Toolkit), a Java LDA implementation from UMASS Amherst by McCallum[14] and its further developments with Fast Collapsed Gibbs Sampling[28] provided a valid alternative to compare with the Online LDA from Blei et Al.

3.2 Topic Modelling for knowledge extraction

In their article [4], Matthias Eickhof and Nicole Neuss propose an analysis whose objective is to provide insight into the available methods for topic mining, and how these methods are applied both in Managerial information systems and other managerial disciplines. This

research exposes yet a lack of adoption of such model types in software intended for the use by social scientists. It also validates that in MIS text mining analysis are solid tool and widely used for content analysis.

3.3 Topic modeling applications on YouTube

As already highlighted, social media, while they continue to rise, have drawn the attention of many researchers. Many aspects, both quantitative and qualitative, have been studied and several approaches have been tried in order to describe this multifaceted reality.

In the study "Social Networks and the Diffusion of User-Generated Content: Evidence from YouTube"[26], Susarla et al. aknowledge the enormous success of social media platforms, and starting from video information and user information collected from YouTube social, analyse how interactions between users are influential not only in determining which videos become successful but also on the magnitude of that impact. In another paper by Kaiser et Al.[10] topic modelling is used on YouTube titles and tags in order to analyse misinformation in Brazil regarding the Zika virus in the video recommendations. In a publication on Springer by Obadimu et Al.[19], using topic modelling, the authors examined five recurring forms of toxicity among the comments posted on pro- and anti-NATO channels on YouTube. By identifying and examining the toxic behaviors of commenters on YouTube, their analysis helps understanding toxicity on online social networks. Topic modelling has been used also for community detection, Gargi et Al.[6] study the YouTube video graph to generate named clusters of videos with coherent content starting from videos with a high amount of views and than apply a scalable greedy algorithm. In their proposed solution they systematically add the next "most viewed video" that is not connected to any of the seed, so that coverage of the cluster divided by the size of the cluster is maximized. At that point they calculate, with topic modelling, text features that are used to compute the text coherence of each cluster. In this approach Gargi et al.[6] extract the text features again only from titles and descriptions.

These articles not only provided valuable information about the application of topic modeling on social media, but also provided interesting insight about the direction of this study.

Chapter 4

Methodology

This chapter describes the core idea of this thesis, focusing on the main steps and the decisions which will affect the rest of the work. More in detail it will deal with the main idea behind this project and the related research questions that have been introduced in the previous chapters. Also, the reader is supplied a glossary with some definitions that can be useful for the reading. In the last part is shown the proposed solution at high level, with a focus on the different macro-phases, their inputs, and outputs through the use of graphical tables. To conclude this chapter a high level description for the proposed solution is given, with a focus on

4.1 Core idea

The core idea is to show how exploratory data analysis can be used on social media platforms in order to obtain knowledge and valuable information on the most recent trends and events.

These platforms are a rich source of value, not for a monitoring role in contexts such as journalism and activism, but also for the collective intelligence[24] which can be leveraged through the use of automated web science technologies.

Thanks to the nature of YouTube platform, there are a multitude of dimensions that can be taken into consideration, from video and sound analysis to image processing, from regressions to text mining; In this thesis we will focus on the latter and the metadata of the trending videos in order to perform high level content discovery on a particular category of videos with high user engagement.

Beforehand, a first component will analyse the metadata related to the trending videos of different nations with the intention of highlighting the different usage of the platform and characterise the different user-bases.

4.2 Objectives and research goals

The research starts with the purpose of answering the following questions:

- Does each country use Youtube differently?
- Is topic modeling a valid method for exploratory analysis in this context?
- Does the topics of a video influence its success? Can we use the topics to predict the number of views of a YT video?

The main goal of this thesis is to test if topic modelling is a valid tool on mostly automatic generated video closed captions in order to perform content discovery on the social media platform of YouTube.

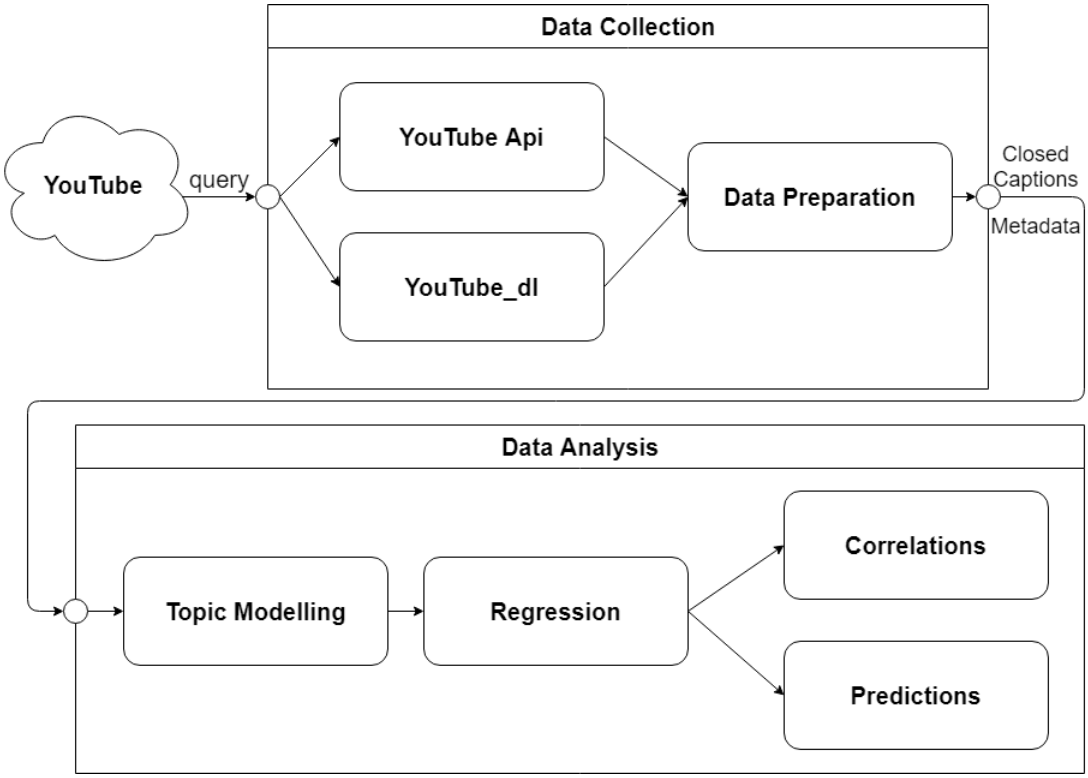

Figure 4.1 Overview of the proposed solution

4.3 Data Collection

The first job required in this thesis is the data collection. We query the platform's API in order to collect the data required for any further investigation. As previously stated, this project focuses on those videos which are classified daily by the platform as trending, due to their high user engagement.

4.3.1 Youtube Data API and Youtube_dl

In this phase we collect data from two different tools.

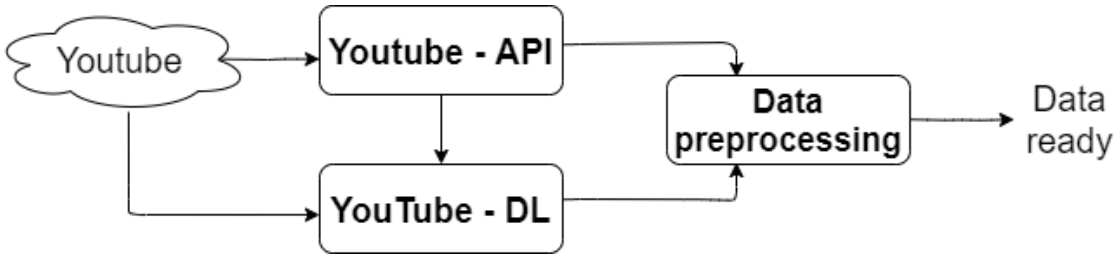

Figure 4.2 Data collection pipeline

In order to retrieve the metadata of the user shared content and the closed captions of the videos. For the metadata of the trending videos, such as video ID, title, views, and so on we use the YouTube data v3 API, this interface is freely available to every developer but up to a very restrictive daily cap. Another constrain to the YT API is the lack of access to the automatically generated closed captions of the videos without the consent of the up-loader of the video, a requirement that inevitably excludes this tool for a scalable and completely automated solution. One first alternative candidate was Selenium WebDriver, a python framework used for automated testing of websites, although successful in its retrieval task, it has been excluded due to low performance. The final choice is to use a third party module called Youtube_dl, a well maintained and faster solution.

4.4 Data Preprocessing

Once the data has been collected a preprocessing is required: the large collection of data retrieved from the YT API presents dirty data, coded attributes, unreadable characters and data from 11 different countries.

In this phase, not only we clean the data collected and preprocess it for the next phases of this work, but we also inspect the data collected and focus on answering to the research

question: "Does each country use YouTube differently?". In doing so we compare the categories, views, likes and dislikes from a geographical point of view.

A first step of preprocessing is required already in the data collection, as a matter of fact when extracting the closed captions of video we already set a threshold in order to get only the countries with a significant number of English videos. This because the text extracted will be used as documents in input for topic modelling later and a cross language analysis is beyond the scope of this work.

4.4.1 Document preprocessing for LDA

When the closed captions are extracted, they undergo another preprocessing step in order to be ready as a corpus of documents for the LDA topic model.

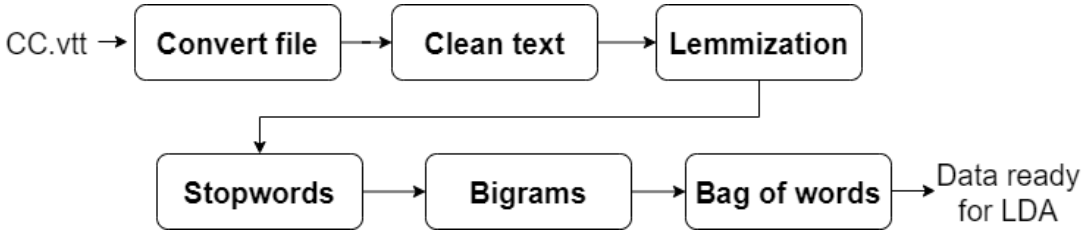

Figure 4.3 Preprocessing pipeline

This pipeline first converts the WebVTT files to texts, then it cleans the text from useless parts such as emails and described sound. After that, three important steps are performed: Lemmatization, which converts each word to its root word (machines to machine, walking to walk, mice to mouse, and so on), stopwords removal, and than bigrams are added. In the last step the documents are transformed in a Bag of Words.

At this point the dataset is in the right shape for the Latent Dirichlet Allocation (LDA) model, the probabilistic topic model which has been implemented in this work. A document-term matrix is in fact the type of input which the model requires.

4.5 Data Analysis

The Data Analysis pipeline receives as input the metadata collected from the videos and a cleaned corpus of closed captions for trending videos of the following countries: United States, Canada, and United Kingdom. The main steps of this phase are the Latent Dirichlet Allocation for topic modelling and the regression model for video views from the topics.

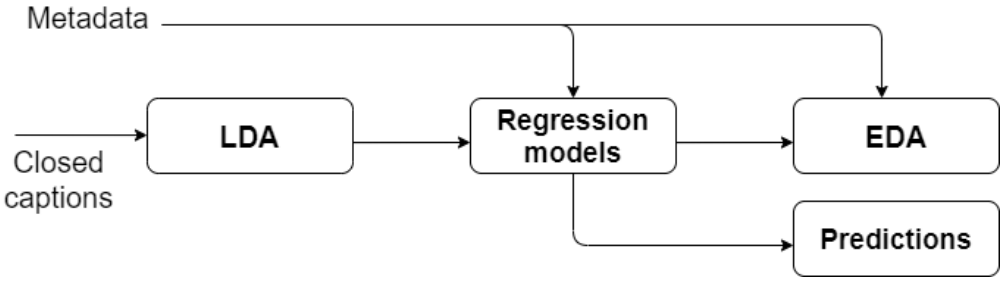

Figure 4.4 Data analysis pipeline

4.5.1 LDA topic modelling

In this step we focus on answering the question: "Is topic modeling a valid method for exploratory analysis in this context?". The first step is to extract the abstract topics of YT trending videos. Topic models provide a powerful tool for analyzing large text collections by representing high dimensional data in a low dimensional subspace.

Each document, corresponding to a trending video, is going to be represented as weighted combinations of topics and topics as weighted combinations of words, as shown in Figure 4.5.

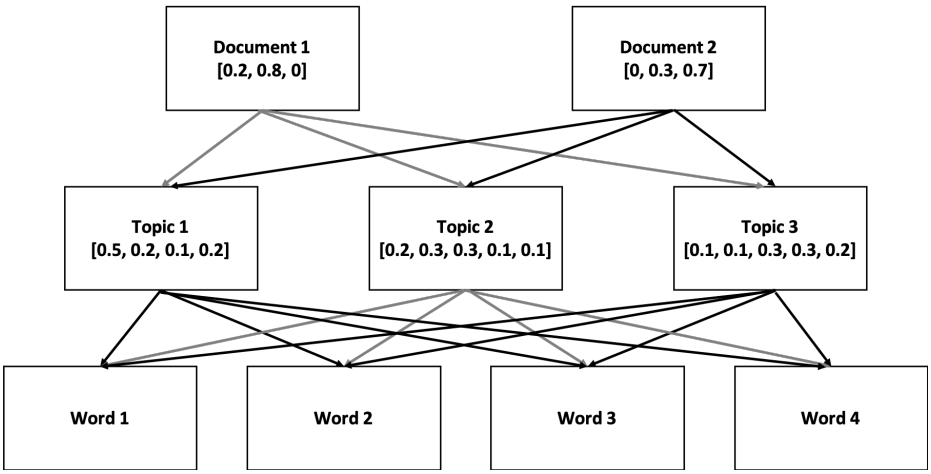

Figure 4.5 Feature vectors

At the end of this phase we inspect the resulting topics, in order to check from a qualitative point of view the knowledge extracted from months of closed captions of trending videos.

We compare two different models: Gensim standard LDA based on the Online Learning for Latent Dirichlet Allocation algorithm by Hoffman, Blei et al.[9], and Latent Dirichlet Allocation with Fast Collapsed Gibbs Sampling (a.k.a. Mallet[14][28]). Mallet (MAchine Learning for LanguagE Toolkit) is a Java-based package put out by UMASS Amherst. The difference between Mallet and Gensim's standard LDA is that Gensim uses a Variational Bayes sampling method which is faster but less precise that Mallet's Gibbs Sampling.

In this process we use the same full corpus composed of all the CC extracted, and after tuning hyper parameters of both models we compare the results using the coherence score as evaluation measure.

4.5.2 Topic vectors

In this stage, illustrated in Figure 4.6, the best LDA model (with respect to the results of the previous phase) and all the CC documents are taken as input. Every document is fed to the LDA model, that evaluates it and express it as a weighted combination of topics. In this way, for every document, we create a feature vector made of the scores of the topics. The topic vector representation of the video is preferable over the one made of tokenized documents, as it is better suited for mathematical manipulation.

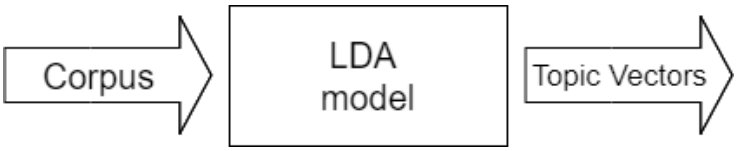

Figure 4.6 Topic vectors

4.5.3 Scalar Regression

In this step of data analysis we want to answer the question: "Does the topics of a video influence its success? Can we use the topics to predict the number of views of a YT video?" With the resulting topic vectors as input we compare the results of some of the most common regression models, in order to find the best method to describe the relationship between the observed and target variables.

- LinearRegression
- RidgeCV

- LassoCV
- PolynomialFeatures with max degree of 3
- KNeighborsRegressor with number of Neighbors from 1 to 10
- DecisionTreeRegressor
- RandomForestRegressor

Since the error calculations on the training data might not be a good estimate for how the model will perform on some unknown data, one common practise in evaluating different machine learning algorithms is to split the data into train and test set. But since the error rate can be highly variable, depending on which observations are included in the training set and which are included in the test set instead, we decide to implement a test and train split in a 2 to 8 ratio, and then apply K-fold cross-validation to evaluate the different models. After testing the different models, we picked the best one using the coefficient of determination as first evaluation metric. Random Forest Regressor is the model that performed better, we proceeded in a two step hyperparameter tuning, but this time on the Root Mean Squared Error, since it produces a more talkative result. We first performed a *RandomizedSearchCV*, and then on the local optimum a *GridSearchCV* in order to refine our tuning. The best parameters are finally used on the test set left away in order to compute a fair RMSE.

Chapter 5

Implementation

In this chapter, we illustrate in detail the entire development and implementation, going through all the stages:

1. Data Collection with YouTube API
2. Data Collection with YouTubedl
3. Data Preprocessing
4. Topic Modelling
5. Regression models
6. Further data analysis

The project is developed on Google Colab and is structured using different Jupiter notebooks for each different section, the code is written in Python language. Data is stored on a private Google Drive repository. In the end, we also shortly present the libraries used in the implementation.

5.1 Data Collection with YouTube API

The starting point for the data collection from YT is to identify the trending videos and collect their respective metadata. We choose YouTube as social media because of its open source nature, the multiple APIs made accessible, and the different data available so that we were able to extract all the information and data needed to conduct our research.

We decide to use the YouTube Data API v3 [1], this API is free for developers up to a daily quota and can be accessed via a Python library called YouTubeDataAPI. To access this resource it's required to register an account and get an API access token.

So, we invoke an API method in order to get the list of trending videos and their metadata. In order to gather the trending videos from specific countries we need to specify some parameters in the request method to the YouTubeDataAPI:

- Part: specify which part of the data to retrieve. Some examples are: id, snippet, contentDetails, fileDetails, liveStreamingDetails, localizations, player, processingDetails, recordingDetails, statistics, status. In our case we are interested only in the Id and statistics.
- Chart: identify the chart from which we gather the data: in our case "mostPopular"
- regionCode: describes the country of origin of the content, it's a string parameter following the ISO 3166-1 alpha-2 country code. We gathered data for eleven countries: USA, Great Britain, Germany, Canada, France, Russia, India, Brazil, Mexico, South Korea, and, Japan.
- Key: the Google project API key
- maxResults: number of results to return for each query, default 5 and max 50.

The *YouTubeDataAPI.videos().list()* method is used to get the results in a paginated structure which returns the "maxResults" number of videos for each page, we set the maxResults equal to 50 and get the first 4 pages returned in order to collect the first 200 trending video each day. The data returned is in JSON format (JavaScript Object Notation), which uses key-value pairs to describe the properties of an object. In this phase we retrieve the information about the trending videos, such as video ID, video title, country, description, tags, category, views, like, dislikes, date of publication and date in which became trending.

In the end we collected a total of 405499 video details, from 11 countries, for 187 days between August 12th 2020 and February 18th 2021. It's worth to note that for 4 days we were not able to collect videos, and for some countries such as India and South Korea for most days we did not get up to the max 200 videos but slightly less results from the API.

[1] https://developers.google.com/youtube/v3/docs

5.2 Data Collection with Youtube_dl

The second step of data collection consist in the download and formatting of video closed captions. In our solution we use youtubedl which is an open-source command-line tool to download videos from YouTube.com. It requires the Python interpreter, version 2.6, 2.7, or 3.2+, and it is not platform specific. For our scenario we are interested only in the closed captions, other data points are beyond the scope of our research.

Although the number of collected trending videos so far is mostly the same across every country, after a quick inspection we can affirm that the number of videos with English CC vary considerably. We decide to gather the CC of videos only from 3 English speaking countries, not only because they present more videos with auto-generated CC, but also because the introduction of videos in foreign languages introduce the risk that erroneous captions for English language from other languages, introducing noise and errors for the next phase.

This restriction brings the number of trending videos down to 112079, almost 600 per-day. Of this list of videos we need only the attribute "videoID", which has the function of unique key on YouTube, in order to retrieve the closed caption. We remove the duplicate IDs of those videos which get to trending for more than one day and we feed the remaining list to a Python script that uses youtubedl in order to get the subtitles.

For our implementation we set the following configuration for the downloader in order to speed up the process and store only the required information:

```
youtube_dl_options = {
      'writeautomaticsub': True,
      'skip_download': True,
      'outtmpl': '/content/drive/My Drive/storage_folder',
      'subtitlesformat': 'best',
      'cookies': '/content/drive/My Drive/youtube.com_cookies.txt',
  }
```

Listing 1 Youtube-dl options configuration

During this process most of the videos did not return any captions, either because the video was not spoken or because of limitations of the YT automatic CC feature. Another problem we encountered in this step was a 429 temporary ban due to excess requests, which required the implementation of a 4 seconds timeout between each video. Also, we decided to implement a checkpoint system in order not only to be able to stop and restart the download

from the last checkpoint in case of unexpected errors, but also for checkup of specific download sessions. The resulting files are stored in .vtt format on a google drive.

5.3 Data Preprocessing

Apart from some data manipulation necessary for the previous data collection steps, most part of the data collected so far still needs cleaning and some transformation steps before it can be used as input in our analysis. The main objective of this phase is to go from the dirty metadata table to a clear dataset containing all the relevant information for each country, and to format and clean the closed captions into text documents ready for the LDA topic model.

Beside dropping the null values present in the metadata, checking the data types in order to avoid potential data loss, and extracting extra month-year columns for aggregated analysis; there is a video category field populated for most videos extracted, this field is a numeric code corresponding to a specific category, for each country there may be a different code but the categories are the same.

Quite more complex is the text processing required for the preparation of the corpus, as already shown in figure4.3 in section 4.4.1. We are now going to explain more in details the steps performed in this pipeline:

- **Clean text**: this first step consist in removing specific parts of text which do not take part in the topic analysis, such as emails or words between square brackets used for sound effect description(for example "[music]"). This task is performed using regex expressions on the documents.

- **Preprocess text**: in this step we performed a tokenization that consist in splitting the text into sentences and the sentences into words. Then lowercase the words and remove punctuation.

- **Lemmization**: Like lemmatization, stemming reduces a word to its root form but the result is not necessarily a word itself since this technique is heuristic-based. It has been proven that in our context, lemmization performs better than stemming, as proven by Balakrishnan et Al.[1] in 2014.

- **Stopwords**: in this step we remove all those words that are most common in the English language that don't carry much semantical meaning. These words are usually referred as "stop words" in NLP, in our case we import a list of them from the library Spacy.

Form	Suffix	Stem	Lemma
studies	-es	studi	study
studying	-ing	study	study

Table 5.1 Example of the difference between stemming and lemmatization

- **Bigram**: at this point we identify word pairs within data, called bigrams. Bigrams are entity generated on the conditional probability of a token given the preceding token, we use the *gensim.models.Phrases* from the gensim library and keep a base *min_count* = 5 and a *threshold* = 50.
- **Bag of Words**: The last step is to generate a Bag-of-Words representation of our data, in order to know the occurrence of each term in each document. We use the methods from the Gensim library in order to do so. First we build the dictionary of all our documents; a dictionary is a document-term matrix that "encapsulates the mapping between normalized words and their integer ids"[2]. Than, the dictionary is used to create the Corpus, by converting it into the BoW representation, through the *doc2bow()* function.

At the and of this workflow we obtained the corpus, a list of vectors whose length is equal to the number of videos with closed captions we collected in the previous phase. We can now use this collection as input for the topic modelling, as described in the next section.

5.4 Topic Modelling

This section deals with selection and training of topics models able to represent videos as a set of topics. Latent Dirichlet Allocation (LDA) has been selected for this task. As discussed in chapter 2.3, LDA is a semi-unsupervised method for topic modeling since it requires specifying in advance the number of topics to be extracted. The main concerns is to find the best number of k topics to describe the videos present in our dataset.

We use the Gensim library and compare two different algorithms: the first is the standard LDA of Gensim based on the Online Learning for Latent Dirichlet Allocation algorithm by Hoffman, Blei et al.[9], the second one is a Java implementation from UMASS Amherst called Mallet which uses collapsed Gibbs sampling in order to speed up computation, this second model is widely adopted in the linguistic and humanistic field more than in computer

[2]https://radimrehurek.com/gensim/corpora/dictionary.html

science. Mallet is accessible on Gensim as well thanks to a Python wrapper method, but it still requires to download the Java package[3] in a directory mounted and accessible on Colab.

5.4.1 Model selection

As stated in the previous chapter, the goal of this step of topic model implementation is to extract the best number of topic to describe our videos. The lack of labels on an unsupervised learning model's training data such as LDA makes evaluation problematic because there is nothing to which the model's results can be meaningfully compared. One good measure to evaluate LDA is Topic Coherence. Topic Coherence measure is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity. We choose to evaluate, for an increasing number of k topics, the coherence score of the models, keeping track of the computational time as well for our corpus of 14058 documents. In order to have reproducible results we set the *random_state* parameter to 100, the alpha parameter is left to the *'auto'* value, the number of passes equal to 10, and chuncksize equal to 100. We set an arbitrary limit for the number of topics to 100; due to the nature of the dataset it is plausible to assume a high number of different topics may be present in the data, but still a number higher than 100 would probably lead to too scattered and less meaningful results.

5.4.2 Feature vectors

At this point we use our best LDA model to extract the topics of our videos. All the documents are transformed in their topic vector representation and added to the respective video metadata. So, if the final LDA model has k topics, the topic vector of a document n will be:

$$topic\ vector_{k,n} = (score_{1,n}, score_{2,n}, ..., score_{k,n})$$

with:
$score_{i,n}$ is the score of the topic i for the document n, with i from 1 to k

In the figure 2 below an example of topic vector for a document.

[3] http://mallet.cs.umass.edu/dist/mallet-2.0.8.zip

```
[
(4, 0.42523986), (36, 0.16016124), (23, 0.08013795),
(5, 0.067439966), (21, 0.056753136), (20, 0.048525784),
(8, 0.018027443), (28, 0.017915955), (19, 0.016125029),
(27, 0.015038591), (30, 0.013560475), (39, 0.01109642),
(40, 0.010313854), (37, 0.010162509)
]
```

Listing 2 example of topic vector for a random video

5.5 Regression models

In the last step we use the library scikit-learn to compare some of the most common regression models against our data and evaluate with which precision we can predict the number of views from our topic vectors.

For this evaluation we use k-folding cross-validation with a number of folds *cv*=10 using the *cross_val_score()* method from sci-kit learn library. In detail we compare the coefficient of determination (R^2) score for the following models:

1. LinearRegression
2. RidgeCV
3. LassoCV
4. PolynomialFeatures with max degree of 3
5. KNeighborsRegressor with number of Neighbors from 1 to 10
6. DecisionTreeRegressor
7. RandomForestRegressor

In this first round of analysis all of the aforementioned ML models are implemented with the default parameters from the scikit learn library[4]. In a second round of analysis we take the best regression model selected in the previous task and perform some fine tuning. For this process, instead of the R^2 score, we consider the root mean squared error (RMSE) as evaluation parameter for our prediction, which in our opinion it gives a more intuitive performance metric at this stage of the research. In order to fine tune the hyper parameters we first split the dataset in a train set and a test set with a size of 0.8 and 0.2 of the whole

[4]https://scikit-learn.org/stable/supervised$_l$*earning.html#supervised* − *learning*

set. Then we use the *RandomizedSearchCV* on the train test, which given an initial space of hyper parameters to explore, uses again k-folds *cv*=10 to test randomly an arbitrary set of 500 random samples within the hyper parameters space. From this random set we then perform a *GridSearch* on 72 combinations of hyperparameters candidates in order to further investigate the results. The best parameters are later used on the test set left away in order to compute a fair RMSE.

5.6 Further Data Analysis

Over the development of this thesis further analysis where performed on the data collected in order to: first, fulfil the purpose of this thesis to propose a valid insight on the tool and technologies used for exploratory data analysis on video shared on social platforms, and second to answer the research question reported in the objectives and research goals section.

Between the other tools we used for this analysis there are:

- Numpy, Pickle, and Pandas: for data wrangling, data storage, and quick inspection.
- Matplotlib, Plotly, MS Power BI, and Seaborn: for data visualization and interactive graphs
- PyLDAVis, t-SNE, and WordClouds: for topic visualization and inspection

As stated in the introduction of this chapter, the various steps are encapsulated in different Jupiter Notebooks, which allow us to load the data from a shared drive, run the calculations, and visualize the results obtained directly every time we go back to the file. Most of the data (such as metadata, closed captions and models) are stored in csv, json and vtt files at the end of each step on the Google drive.

5.7 Libraries

5.7.1 Gensim

Gensim [5] is an open-source library for topic modeling and natural language processing, that is implemented in Python and Cython. In this case, it offers support to perform LDA and calculate the coherence and perplexity.

[5] https://radimrehurek.com/gensim/

5.7.2 Spacy

SpaCy [6]is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. In this case, is used for removing stop words and lemmization over our Corpus.

5.7.3 PyLDAvis

PyLDAvis [7] is a Python library for interactive topic model visualization. It is a port of the widely used R package by Carson Sievert and Kenny Shirley. In our case we used it in order to interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization.

5.7.4 Scikit-learn

Scikit-learn [8] is an open-source library that provides useful tools for data mining and machine learning tasks. We used it for the supervised machine learning models in the regression problem.

5.7.5 Scipy

Scipy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. Among its core packages we used:

- NumPy [9], the fundamental package for scientific computing with Python.
- Matplotlib [10], a Python 2D plotting library.
- Pandas [11], which provides easy-to-use data structures, the pandas dataframe, that are widely used.

[6]https://spacy.io/
[7]https://pyldavis.readthedocs.io/en/latest/readme.html
[8]https://scikit-learn.org/stable/
[9]http://www.numpy.org
[10]https://matplotlib.org/index.html
[11]http://pandas.pydata.org/index.html

Chapter 6

Experiments and results

This chapter describe the results of previous phases of data collection and data processing, focusing on how the methodology and the implementation, have been put to the test in different experiments. First, the collected dataset is analyzed and a preliminary exploratory analysis is presented. Then, we focus on the result of the topic modelling phase, where several combinations of methods are used in order to visualize and evaluate the topic extracted for the video closed captions. In the last part, we study if the topic vectors extracted are good enough to predict the views of a youtube video, and we present the results in order to understand which model is the best solution to achieve our objective.

6.1 Dataset

As already mentioned in section 4 and 5, our dataset is composed from files collected from different data source, such as: video details and metadata from the YouTube Data API, and video closed captions collected with Youtube-dl. In the next paragraphs we are going to illustrate how a single video is recorded and the overall structure of the whole dataset.

6.1.1 Video details and metadata

The smallest entity of our dataset is the single trending video on YouTube. Each video is extracted as a JSON object from the API, the primary key attribute of this object is the *videoID*, then more attributes associated to a video ID are retrievable, based on the *part* key declared in the API request. In our case the two parts we are interested in are *snippet* and *statistics*, as shown in the example in fig 3.

```
{
   "kind":"youtube#video",
   "etag":"etag",
   "id":"string",
   "snippet":{
      "publishedAt":"datetime",
      "channelId":"string",
      "title":"string",
      "description":"string",
      "thumbnails":{
         "(key)":{
            "url":"string",
            "width":"unsigned integer",
            "height":"unsigned integer"
         }
      },
      "channelTitle":"string",
      "tags":[
         "string"
      ],
      "categoryId":"string",
      "statistics":{
         "viewCount":"unsigned long",
         "likeCount":"unsigned long",
         "dislikeCount":"unsigned long",
         "favoriteCount":"unsigned long",
         "commentCount":"unsigned long"
      }
   }
}
```

Listing 3 An example of YouTube video as JSON object

Apart from these fields, we add to each video the following external attributes necessary for temporal and geographical investigation:

1. Country categoryID
2. Trending date
3. Category Name

At the end of this phase we collected a total of 14058 video metadata, divide over 7 months as shown in the table6.1:

	United States	United Kingdom	Canada	Combined
August	484	545	548	1577
September	782	891	914	2586
October	739	815	788	2336
November	540	665	632	1835
December	598	714	669	1981
January	589	763	686	2038
February	485	657	563	1705
Tot CCs	4217	5050	4800	14058

Table 6.1 Count of closed captions collected

6.1.2 Metedata composition

Overall the video collected sum to a total of 405499, for 11 countries, for 187 days between August 12th 2020 and February 18th 2021. Firstly we check the distribution of views, likes, dislikes, and comments of trending videos as shown in 6.1 below.

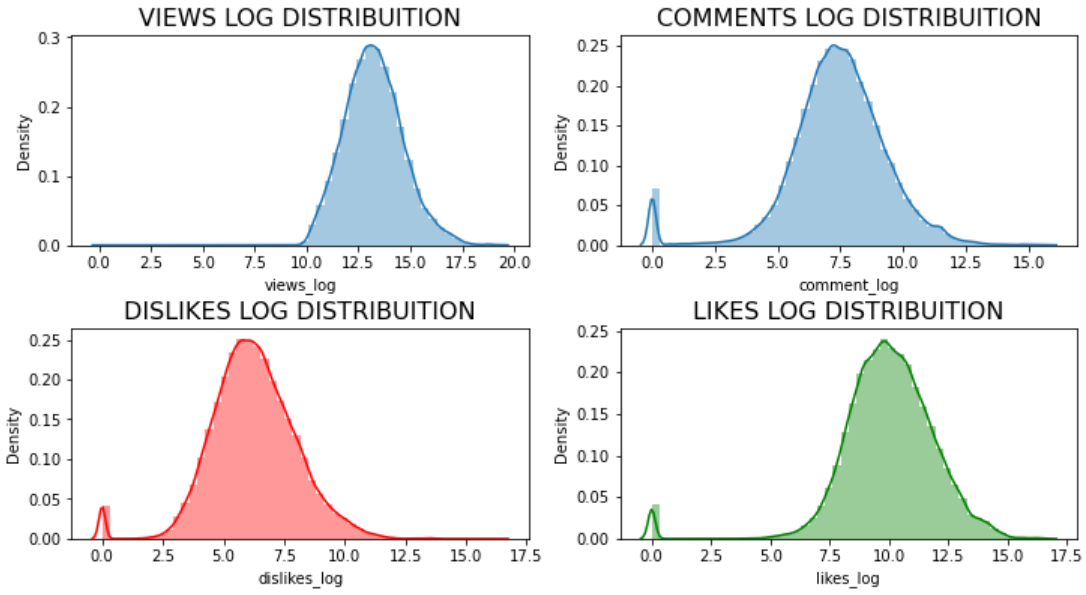

Figure 6.1 Trending video distributions of views, likes, dislikes, and comments.

Since our research has an exploratory nature, with a focus on the topics of the videos, we thought that some initial analysis on the categories of the video could be a valid starting point for further investigations. As shown in figure 6.2, we check the count of videos for each category.

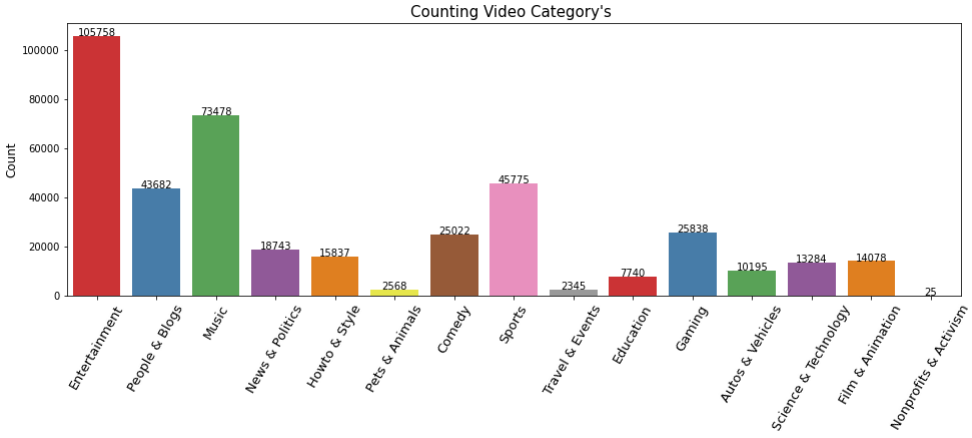

Figure 6.2 Count of trending videos for each category.

Then we compare, using box plots in fig 6.3 , the distribution of views, likes, dislikes, and comments across different categories.

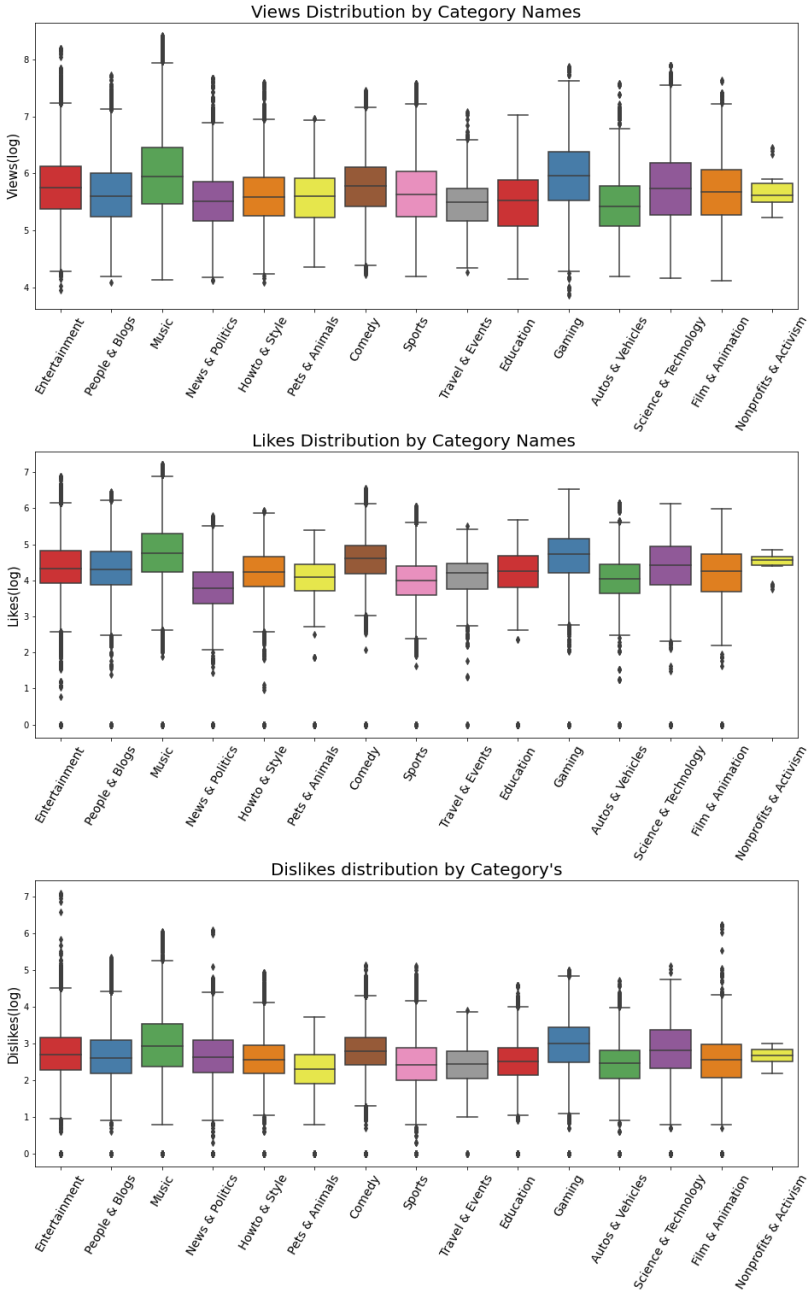

Figure 6.3 Box plots describing distribution of views, likes, and dislikes over categories.

6.1.3 Geographical analysis

In our metadata the only geographical reference we have is the country where the video is published. At this coarse-grained level of data, the analysis is quite similar to the one already performed for the categories. Yet the results are quite interesting, since they allow us to see how the platform usage changes significantly between different countries.

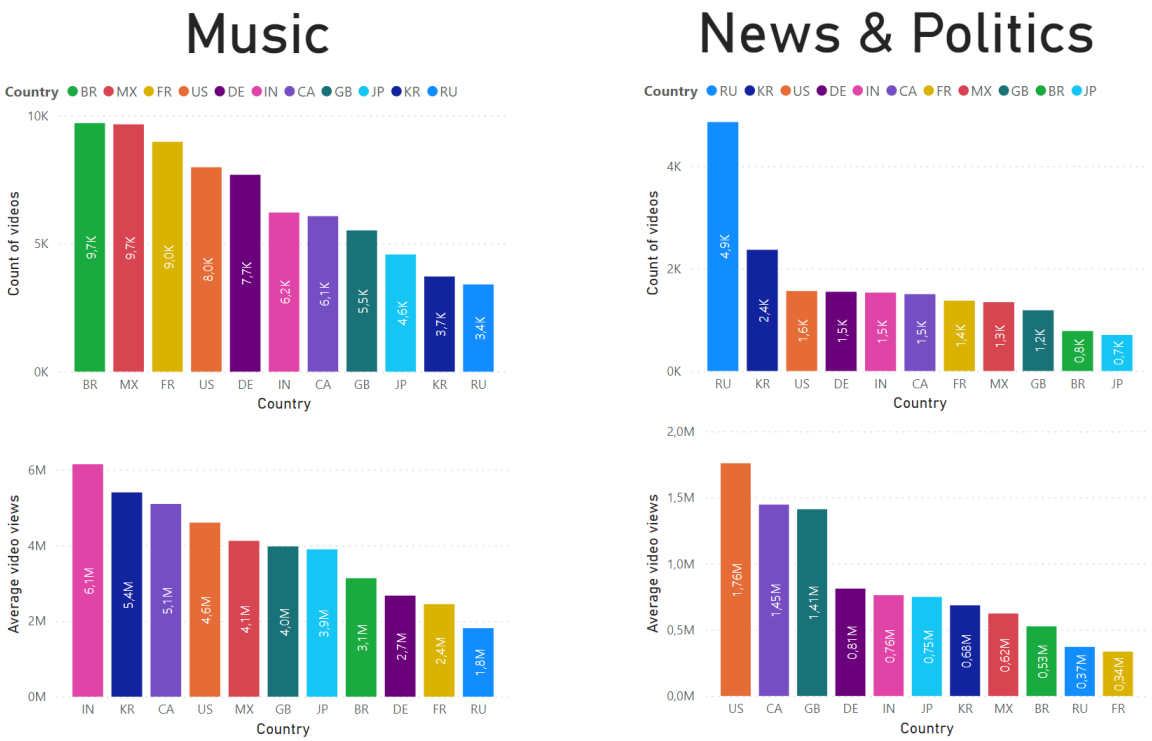

Figure 6.4 Country comparison for "Music" and "News and Politics" categories.

In the figure 6.2 we can see an example for two specific categories: ”*Music*” and ”*News and Politics*”. With a glance we can see how in Russia the YouTube platform has a larger user engagement for the ”*News and Politics*” whereas in the rest of the world there is greater usage for entertainment and music videos.

6.2 Topic Modelling

In this section the results obtained from the Latent Dirichlet Allocation are presented. More in detail we will investigate the topic extracted using PyLDAvis, a Python wrapper for the popular R package LDAvis[25] from Sievert and Shirley. We will then generate the topic vectors used as features in the regression models in the last phase of analysis.

6.2.1 Document Preprocessing

The second major chunk of our dataset is composed of the closed captions of the youtube videos. This data are downloaded as separate WebVTT4 (Web Video Text Tracks) files.

```
WEBVTT Kind: captions; Language: en
00:11.000 --> 00:13.000
<v Roger Bingham>We are in New York City
00:13.000 --> 00:16.000
<v Roger Bingham>We're actually at the Lucern Hotel, just down the street
00:16.000 --> 00:18.000
<v Roger Bingham>from the American Museum of Natural History
00:18.000 --> 00:20.000
<v Roger Bingham>And with me is Neil deGrasse Tyson
```

Listing 4 Example of WebVTT file

These files are cleaned and transformed in plain text using the WebVTT Python module. The texts are stored in a Pandas dataframe containing the corresponding *videoID*s of the videos extracted. From this dataframe a Dictionary is obtained and is fed to the preprocessing pipeline described in the section 5.3. We obtained a clean corpus of 14058 documents from the aforementioned pipeline, from which a dictionary is produced. A dictionary is document-term matrix that "encapsulates the mapping between normalized words and their integer ids"[1], and in our case it consists of 64926 different words and bigrams.

[1] https://radimrehurek.com/gensim/corpora/dictionary.html

6.2.2 LDA evaluation

The results of the comparison already described in section 5.4.1, between LDA and MALLET, are reported in the figure6.5 below. We used the topic coherence as evaluation metric of the model: this metric does not tell us with certainty the quality of the model, but it gives a good hint on the comprehensibility of our topics. A good level of coherence is usually considered 0.60, whereas a lower acceptance score is around 0.40.

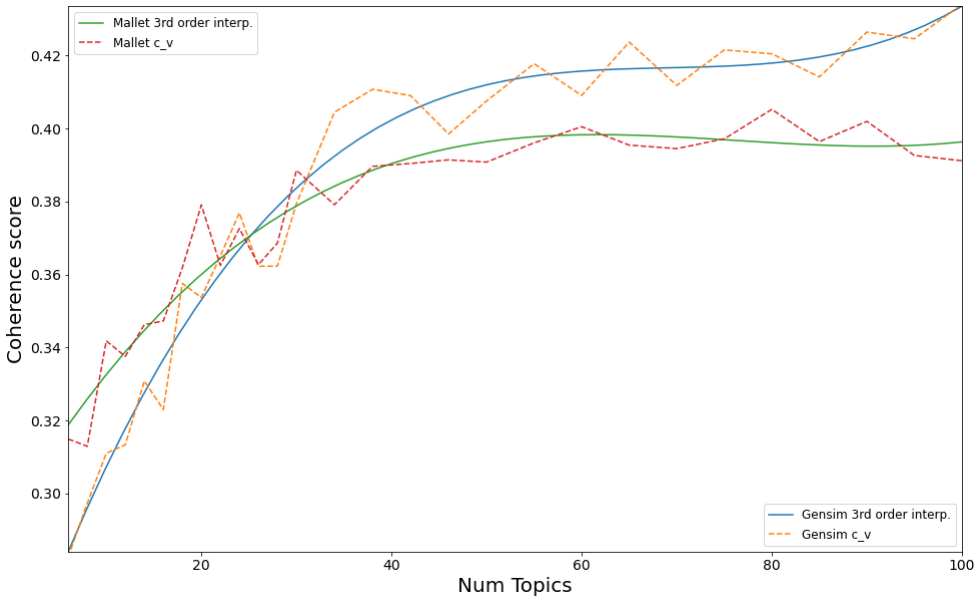

Figure 6.5 Coherence score comparison LDA and Mallet

As we can see from the figure6.5, we get better results with the MALLET package for small number of topics, but a clearly lower topic coherence score for higher k values compared to the online variant of LDA. Since the computational time increased linearly between 10 and 30 minutes for increasing n. of topics and ram consumption was not a problem for both the tested algorithms, we accept the standard LDA as model better fitting our data.

As already stated previously in section 5, a higher amount of topics hinders the analysis due to the fragmentation of our dataset in small subsets of scarce significance. But we also want a number of topic k high enough to be able to capture a good number of arguments talked over the six months, since we already discovered that there are many different categories of videos in the trending dataset. For this reasons we decide that 42 as number of topics is a good trade-off between too many and too few topics. The coherence score is still above the acceptance score, in what appears to be a plateau of local optimum, as show in the figure6.6, and the highest coherence score obtained with more topic is a small and not particularly significant increase.

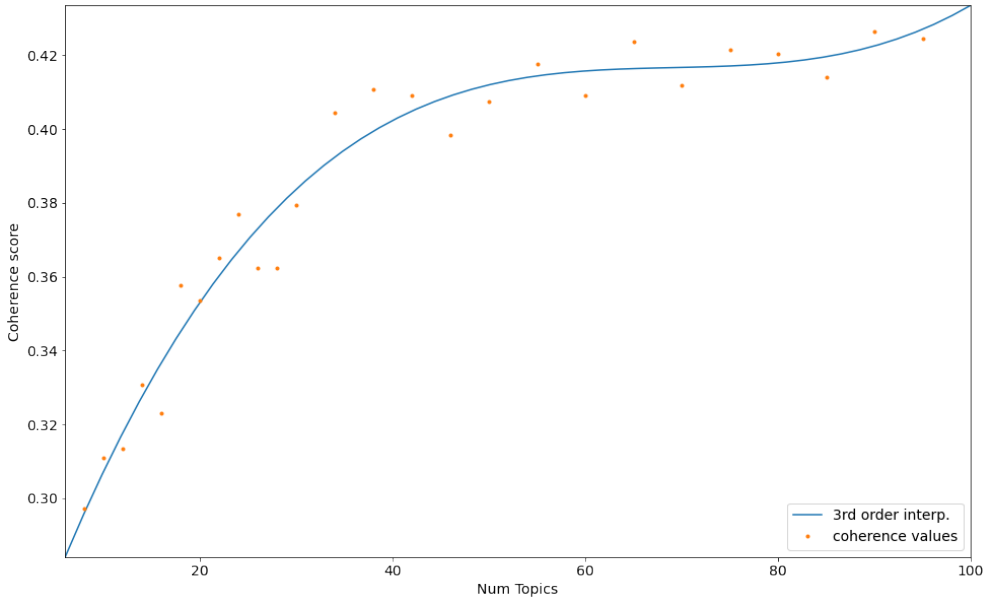

Figure 6.6 LDA coherence evaluation

6.2.3 Topic inspection

A good way to access the quality of the LDA model is to manually inspect the topics extracted and check if it simply clusters words of similar meaning or more semantically different words in the same topic. In this task we use the library PyLDAvis, which uses Jensen-Shannon Divergence Principal Coordinate Analysis(JS_PCoA) for multidimensional scaling[2] to show the topics in a 2D plane and the individual terms that are most useful for interpreting the various topics. The figure 6.7 below illustrates on the left panel the different topics, larger topics have a larger presence in the corpus, yet they often are less interpretable. It's also important to remember that topics close to each other are usually more related, but due to the multidimensionality reduction of JS_PCoA we can't assume nor in which way nor how much correlated from their distance in the visualization.

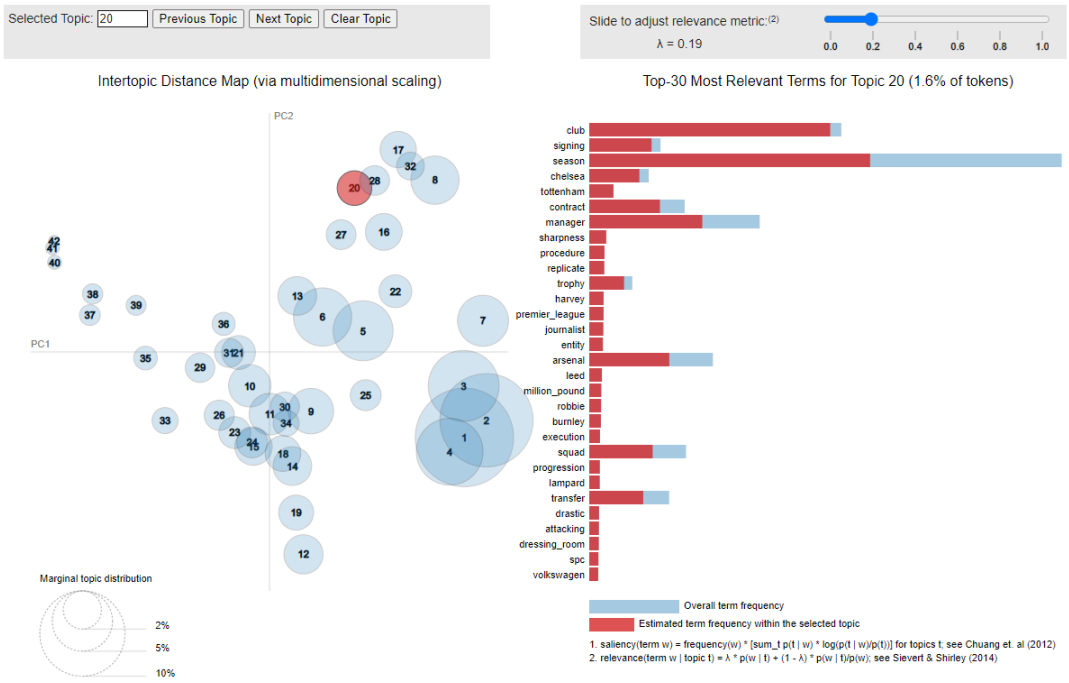

Figure 6.7 LDA topics visualised with PyLDAvis, with the global topic view on the left, and the term barcharts (with Topic 20 selected) on the right. Linked selections allow users to reveal aspects of the topic-term relationships compactly.

In the next figure 6.8 we inspect the most relevant terms of some topics. The relevance is adjusted based on a λ relevance which is parameter with a value from 0 to 1. Setting $\lambda = 1$ results in the familiar ranking of terms in decreasing order of their topic-specific probability, and setting $\lambda = 0$ ranks terms solely by their lift[25]. In our experiments a λ

[2]https://pyldavis.readthedocs.io/en/latest/modules/API.html#pyLDAvis.js_PCoA

between 0.18 and 0.40 gave the most significant terms. Looking at the figure 6.8 we can note the use of bigrams such as Premier_League and million_pound. We can see from the top-left topic example (T10) already that we have a good model, since the word Perseverance which presents different meanings, in this case has been correctly assigned to a topic relative to the landing of the NASA Mars rover called "Perseverance" happened in the time period considered by our experiments.

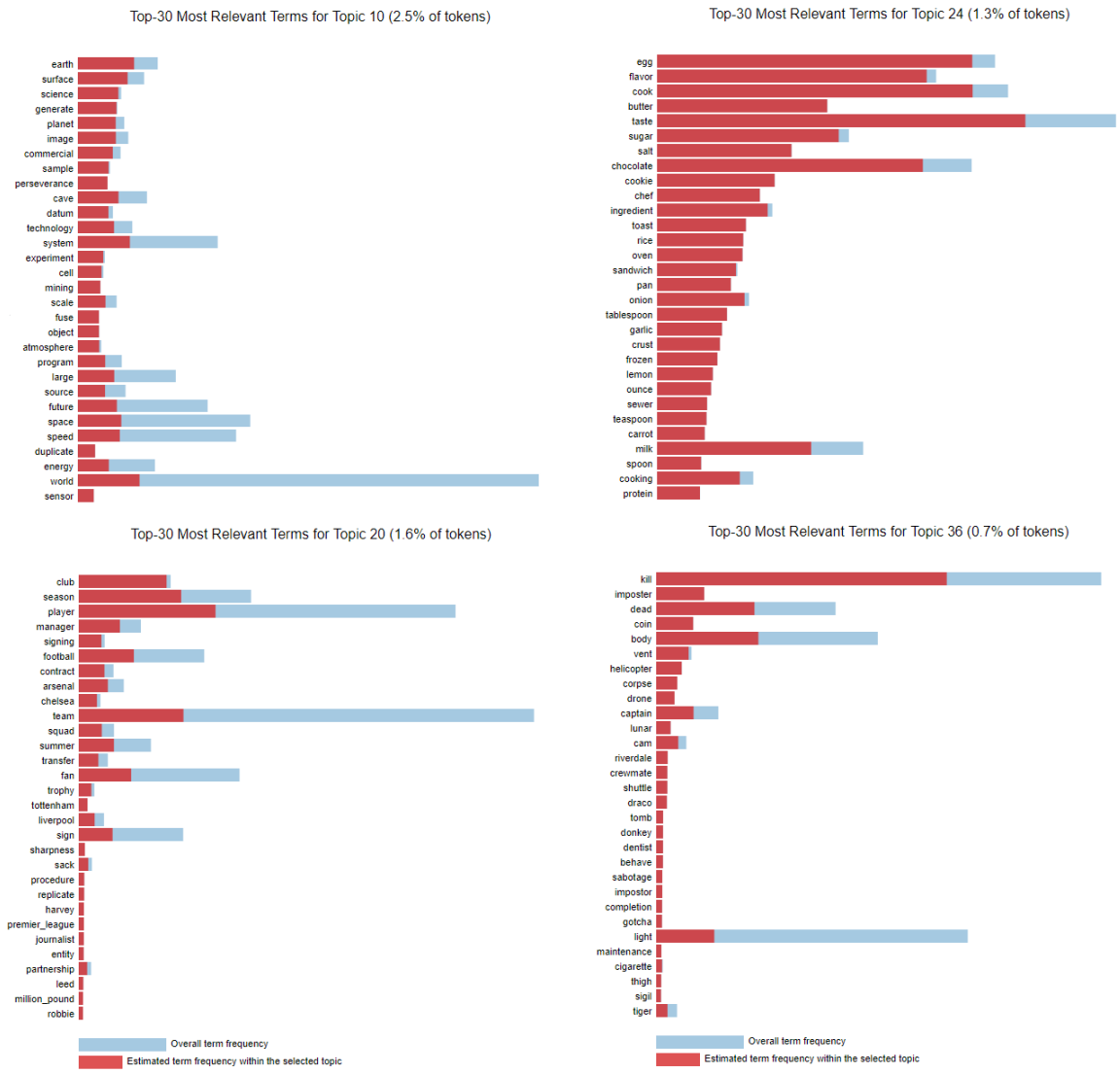

Figure 6.8 Four examples of relevant terms for topics extracted with LDA, in the figures we can quite easily guess the topic from the words.

A valid point of the topic analysis is that not only it allow us to extract knowledge from a qualitative point of analysis with the inspection of the extracted topics, but also to get quantitative information about which topics have been most discussed in the corpus.

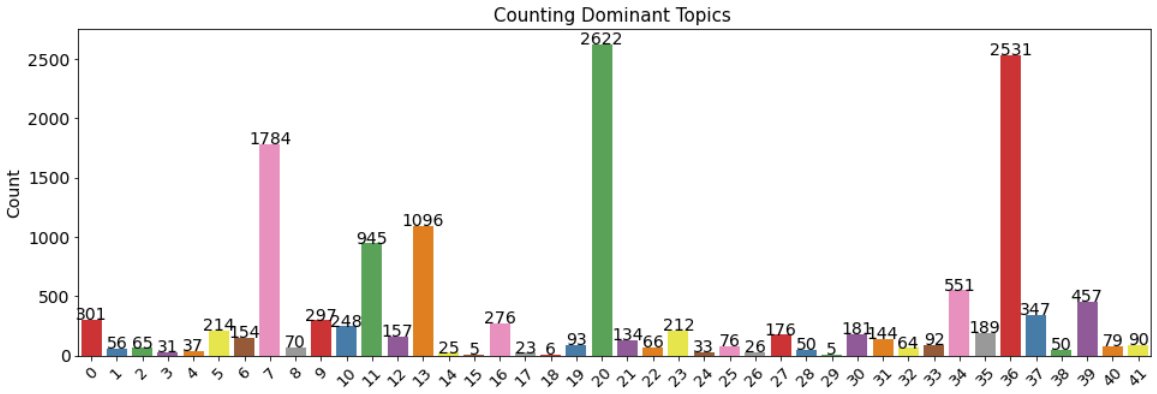

Figure 6.9 Count of videos by dominant topic.

One information that can be valuable is the count of videos per "dominant topic". As explained in section 5.4, each document is a combination of more topics that sum up to 1; we can define as dominant topic the topic with the greatest value in the document. In the figure 6.9 we show how most videos fall in the top 5-7 topics out of the 42 available, interestingly the dominant topics are not the ones with a largest presence in the topic. These topics (figure 6.12) in fact are those more related to the colloquial part of the videos, which are the majority in this dataset due to the specific dimension considered in this work.

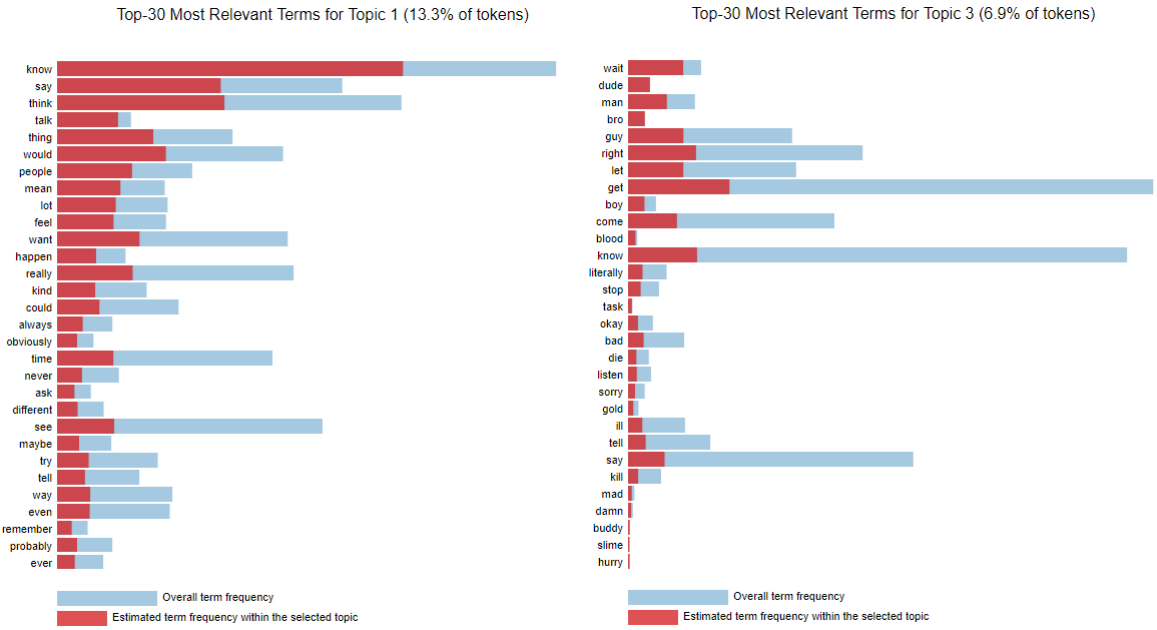

Figure 6.10 Example of topic with a large presence in the document, as we can see from the relevant terms this is a topic related to the colloquial part of the videos

6.3 Views prediction

As last step in our analysis we merge the topic vectors obtained as output from the LDA with the metadata on the respective *videoID*s. In this section we check multiple models using a k-folding cross-validation with a number of folds *cv*=10 using the *cross_val_score()* method from the sci-kit learn library. In details we collect the coefficient of determination (R^2) score for seven different models, the results are reported in the next table6.2.

	R^2	Notes
Linear Regression	0.035	
RidgeCV	0.045	
LassoCV	0.046	
Polynomial Features	-0.04	max_ord=2
KNeighborsRegressor1	0.482	k=1
KNeighborsRegressor2	0.328	k=2
KNeighborsRegressor3	0.23	k=3
DecisionTreeRegressor	0.39	
RandomForestRegressor	0.586	

Table 6.2 Coefficient of determination (R^2) score comparison between different regression models.

Due to the high number of features present in the dataset, the time required for calculation with the polynomial features was two order of magnitude grater than for the rest of the models, which for such small amount of data never took more than 10 minutes to compute. Due to resource contraints (both time and memory allocation on Colab) we were not able to compute Polynomial Features with a max order greater than 2. From the results obtained in this step we decide to further investigate and fine tune the Random Forest Regressor, in order to get a sub-optimal solution to the regression problem.

6.3.1 Random Forest Regressor and RMSE

For this last process, we selected the Random Forest as regressor model, but instead of the R^2 score, we are going to compute the Root Mean Squared Error(RMSE) as evaluation parameter for our prediction, which in our opinion gives a more informative performance metric at this stage of the research. We use, on the train test and with k-folds *CV*=10, the *RandomizedSearchCV* first on a random grid described as follow 5, from which we sample 500 results.

```
{'bootstrap': [True, False],
 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}
```

Listing 5 Random grid for hyperparameter optimization in the Random Forest Regressor, the number of possible distinct combinations is 4320.

We then apply *GridSearchCV*, using again k-folds *CV*=10, on a smaller grid of 72 candidates close to the local optimum, in order to further delve our optimization. The best parameters resulting from this 2-step search are later used on the test set left away in order to compute a fair RMSE. We test the best regressor models obtained with 10 different values of k-folds *CV*, from 2 to 20 folds. The result of the fine tuning are reported in the table below.

CV	4	8	12	16	20
RandomForest	5330016	4617426	3142082	3107566	3042540
Best RF after Random Search	5168458	4228941	2521529	2439914	2407748
Best RF after Grid Search	5186300	4231411	2539758	2422173	2400508
Improvement	3,13%	9,18%	24.61%	28.29%	26.75%

Table 6.3 Root Mean Squared Error(RMSE) score comparison between the default and the fine tuned regression model, The improvement(%) is wrt the best result obtained in the two phases.

For better understanding of the result obtained, we report in table 6.4 some useful descriptors for the number of views of the videos.

Descriptors for video number of views	
Mean	2258285
STD	5910368
min	21963
25%	414382
50%	912607
75%	1985079
max	184778248

Table 6.4 Mean, standard deviation, Q_1, Q_2, Q_3, min, and max values for video number of views

6.4 Extra tables

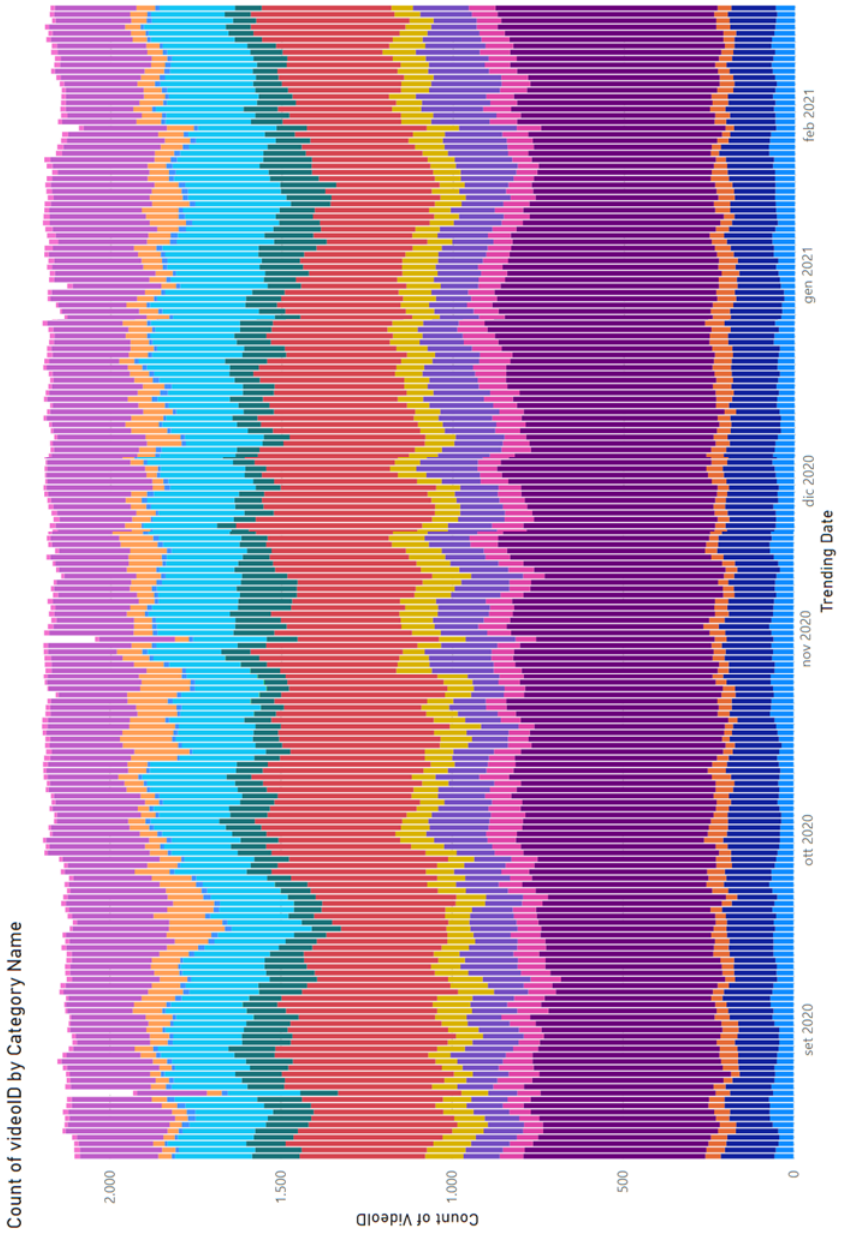

Figure 6.11 Temporal video count by category

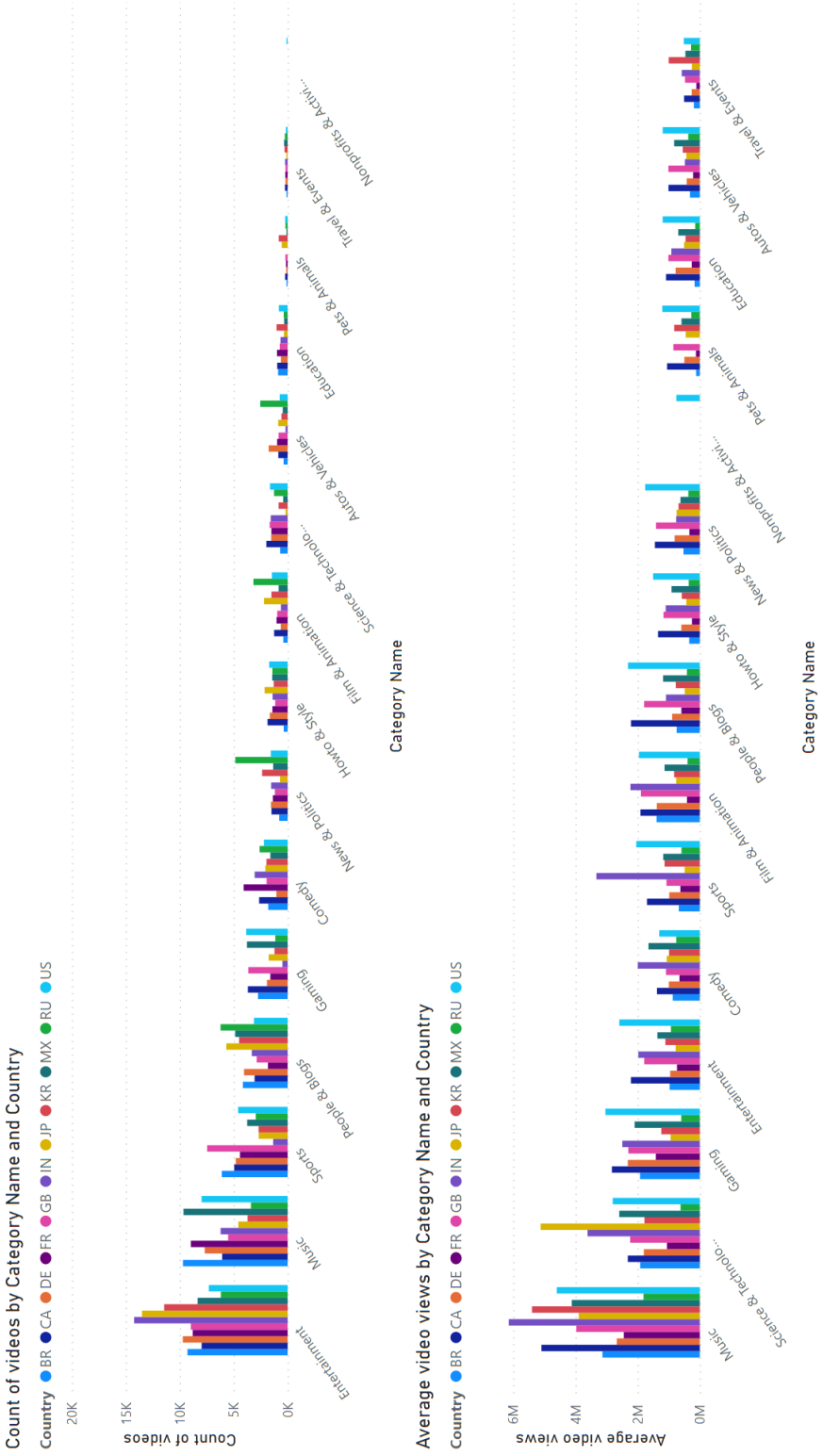

Figure 6.12 Video count and average view by category and country

Chapter 7

Conclusion

The main objective of this thesis was to provide useful insight on the most recent trends and events starting from the videos uploaded to the social media platform YouTube. We explored the social networks population from both a temporal and geographical point of view including in our study multiple dimensions including the spoken script of the videos. A fully-automatic data collection pipeline has been designed to extract a significant amount of raw data on the trending videos. Later we transformed this information into a series of vectorial values useful to represent the different abstract topics treated in the resources, giving us a topic model able to describe those features and their aggregate values for each video entity. Finally we verified the expressive power of the newly calculated dimensions through some clustering and regression experiments.

7.1 Contribution

In this study we proposed a method of Knowledge extraction in a scope of exploratory analysis that applies to the rich domain of YouTube, which is potentially generalizable to any social media. Using minimal prior knowledge about the domain, the workflow proved to be able to extract valuable human-readable unseen knowledge as output. Within the frame of our analysis we built a model able to automatically gather the metadata of trending videos, integrate it with the spoken transcriptions, and process them to generate vectors of topics pertinent to the user content. Each dimension considered was analysed both independently and in relation to the others. We demonstrated the effectiveness of Topic Modeling methods transforming videos closed captions into numerical vectors interpretable by machines. Although the topic vectors have proven poorly effective in predicting the number of views of the videos, it still has shown encouraging results on a regression problem without having to resort to other information than the transcript of spoken videos using a

random forest regressor. We can also affirm that the topic model presented good results and is per se a valid way to describe in a human readable and understandable way the abstract categories that are treated in the videos on social media platforms such as YouTube.

7.2 Future work

The result of this thesis represents a starting point for further study on the prospects of using user generated videos for knowledge extraction from a social network. The potential to study social networks on rich data inputs such as videos uploaded both from professional studios and from random dudes opens the road to many interesting fields of research, from sociology to marketing.

The possibility to leverage the collective intelligence on social networks such as YouTube could lead, in the context of natural language processing, from a simple semantic analysis, such as the one performed in this thesis, to more complex pragmatic analysis. Apart from NLP, many other dimensions could be integrated to the workflow of this thesis, such as: digital image processing, sound processing, relationship between content creators, influence propagation, sentiment analysis, and many others. Furthermore, the modeling of topics has been treated as a static problem and it would be interesting to integrate it with dynamic techniques to evaluate its temporal evolution.

Bibliography

[1] Balakrishnan, V. and Lloyd-Yemoh, E. (2014). Stemming and lemmatization: A comparison of retrieval performances.

[2] Blei, D. M. and Lafferty, J. D. (2009). Text mining: Classification, clustering, and applications. *chapter Topic Models, Chapman & Hall/CRC*.

[3] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

[4] Eickhoff, M. and Neuss, N. (2017). Topic modelling methodology: its use in information systems and other managerial disciplines.

[5] Ferrara, E., De Meo, P., Fiumara, G., and Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70:301–323.

[6] Gargi, U., Lu, W., Mirrokni, V., and Yoon, S. (2011). Large-scale community detection on youtube for topic discovery and exploration. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5.

[7] Greenwood, B. N. and Gopal, A. (2015). Research note—tigerblood: Newspapers, blogs, and the founding of information technology firms. *Information Systems Research*, 26(4):812–828.

[8] Griffiths, T. L. and Steyvers, M. (2002). A probabilistic approach to semantic representation. In *Proceedings of the annual meeting of the cognitive science society*, volume 24.

[9] Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864. Citeseer.

[10] Kaiser, J., Rauchfleisch, A., and Córdova, Y. (2021). Comparative approaches to mis/-disinformation| fighting zika with honey: An analysis of youtube's video recommendations on brazilian youtube. *International Journal of Communication*, 15:19.

[11] Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., and Nithya, M. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1):7–16.

[12] Liao, H., McDermott, E., and Senior, A. (2013). Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 368–373. IEEE.

[13] McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. *http://www. cs. cmu. edu/mccallum/bow/*.

[14] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet.

[15] Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, pages 262–272. Association for Computational Linguistics.

[16] Mitchell, R. (2018). *Web scraping with Python: Collecting more data from the modern web*. " O'Reilly Media, Inc.".

[17] Mohr, J. W. and Bogdanov, P. (2013). Introduction—topic models: What they are and why they matter. *Poetics*, 41(6):545–569. Topic Models and the Cultural Sciences.

[18] Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics.

[19] Obadimu, A., Mead, E., Hussain, M. N., and Agarwal, N. (2019). Identifying toxicity within youtube video comment. In Thomson, R., Bisgin, H., Dancy, C., and Hyder, A., editors, *Social, Cultural, and Behavioral Modeling*, pages 214–223, Cham. Springer International Publishing.

[20] Papanikolaou, Y., Foulds, J. R., Rubin, T. N., and Tsoumakas, G. (2017). Dense distributions from sparse samples: improved gibbs sampling parameter estimators for lda. *The Journal of Machine Learning Research*, 18(1):2058–2115.

[21] Plisson, J., Lavrac, N., Mladenic, D., et al. (2004). A rule based approach to word lemmatization. In *Proceedings of IS*, volume 3, pages 83–86.

[22] Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

[23] Rosner, F., Hinneburg, A., Röder, M., Nettling, M., and Both, A. (2013). Evaluating topic coherence measures.

[24] Schoder, D., Gloor, P. A., and Metaxas, P. T. (2013). Social media and collective intelligence—ongoing and future research streams. *KI-Künstliche Intelligenz*, 27(1):9–15.

[25] Sievert, C. and Shirley, K. (2014). Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70.

[26] Susarla, A., Oh, J.-H., and Tan, Y. (2012). Social networks and the diffusion of user-generated content: Evidence from youtube. *Information Systems Research*, 23(1):23–41.

[27] Vayansky, I. and Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, 94:101582.

[28] Yao, L., Mimno, D., and McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. KDD '09, page 937–946, New York, NY, USA. Association for Computing Machinery.