POLYTECHNIC OF TURIN

Master of Computer Engineering

Master thesis Project

Machine learning-based methodology for online user behavior understanding

Coordinator:
Paolo Garza

Candidate:
Gianfranco Fantappié

July 2018

Acknowledgment

I would like to acknowledge the contributions made by my supervisor from my hosting institution, Prof Than Nguyen, and the help provided by Professor.ess Pinar Øzturk.

Moreover I want to thank the important help and support provided by my local supervisor Prof. Paolo Garza.

finally I would like to thank all the person that gave me personal support over the course of this work, as my family and my friends.

G.F

Contents

List of Figures

Abbreviations

GDPR **G**eneral **D**ata rotection **R**egulation

SP **S**ervice **P**rovider

NFM **N**on Negative **M**atrix **F**actor

LDA **L**atent **D**irechlet **A**llocation

TF **T**erm **F**requency

TFIDF **T**erm **F**requency **I**nverse **D**ocument Term **F**requency

NLP **N**atural **L**anguage **P**rocessing

HBE **H**uman Judgment **B**ase **E**valuation

GLC **G**enerated **L**abel **C**ategories Evaluation

Chapter 1

Introduction

The new release European Privacy Policy:*"General Data Protection Regulation"* known with the acronym of **’GDPR’**, brought a revolution in the area of personal information and data ownership. The new regulation wants to provide to the user a higher degrees of control over their personal information, stored online. Now they are able to download any piece of data, coming from different Service Providers and use them freely.

It is natural to see that this leads to the generation of a high pool of heterogeneous data, related to a wide range of interest, user driven, bringing a change in the area of data analysis. In fact, until this point, the tendency was more company driven, with a high amount of information related to a singular field of interest.

Consequently this new agglomerate of information will generate unique derived insight, owned by the user, providing him more power, not only over those information, but also in a data-market context.

At the same time this insights, by the fact of being related to the online user behavior, may be caring information related to a wide area of interest for the user, as Private Concern [3], or wealth-fare context.

Right now, in literature no documentation is present, related to the process for searching and deriving, from a user perspective, this mentioned insights. This project wants to explore the problematic related to this new approach from a technical point of view, while designing an initial tool for the user able to search and generate some relevant insights.

1.1 Research Question

Regarding the presented problem, it shown a high degrees of complexity, not only because of the high amount of data store online, but also to the high amount of possible different analysis.
For this reason it was mandatory the need of narrowing it down to a sub-problem, limiting the input information and, aiming to a fewer number of potential analysis.
For the following project we selected a limited number of Service Providers from which obtaining the initial pool of data, while aiming to generate the most valuable derive information out of those data.
By working according to the privacy concern of the user, it is mandatory to build an application that doesn't relying to any third part service as support. This prerequisite puts a high constrain on the following work, making mandatory the design of an application that can run in any local calculator, taking under consideration their computational power limitation.
Consequently this project wants to achieve the following Research Questions:

- **Research Question 1: Data Structure Analysis**
  The raw information obtained from the service providers are organized, each of them, as it will be shown in Section 3.2.2, with non-standard data type and structure, making it hard to perform even simple analysis over them. For this reason it was mandatory converting those information into a more suitable and standardized format.

- **Research Question 2: Insights search and generation**
  Since we didn't know what type of data to expect from different service providers, and we had even less idea about how to link them together or what could be derived from their combination, the analysis process has become a rather exploitative enterprise.

- **Research Question 3: Tools Search**
  Concerning the Research Question 2, working in a local user environment, another problem raises according to the search of the tools, or process to use in order to derive the mentioned insights.

Data Consideration

At the same time by working according to the privacy policy regulation, it is not possible to generate a large and well prepared corpora for supervised Machine learning algorithms,forcing us to work uniquely with a single user personal data information as the training and test corpora, putting another high constrain on the accuracy and performances. For the following work, regarding the motivation expressed above, will be used as sample data, **my personal information**.

1.2 Proposed Method

Our method is characterized by a combination of two mayor sides, related to the above Search Questions. The first part wants to explore the data-structure and initial context of the pool of information obtained from the SP, and applying a filter, for selecting the ones carrying the most valuable information.
Secondly, by using this selected data, we applied an Enterprise search in order to see what type of insights may be generated, while evaluating their results, using a set of metrics.
By working with a majority of text-based information, this work will be strongly related to the **Natural Language Processing** field.

1.3 Contributions

Regarding the fact that this field of search was in the initial state, and no previous relevant work was present, this project wants to be a reference study case, for future works. Contemporary, a important contribution obtained from this work is linked to the type of analysis we decided to consider.
This needed the design of an automated process able to cluster and label a pool of wide information potentially unrelated, in the most efficient way.
We will present a new model with new metrics for its evaluation, for Automatically Cluster and Labels, in an Unsupervised way, while presenting the obtained results.

1.3.1 Application Cycle

The steps performed in the following project could be summarized as follow in Figure 1.1

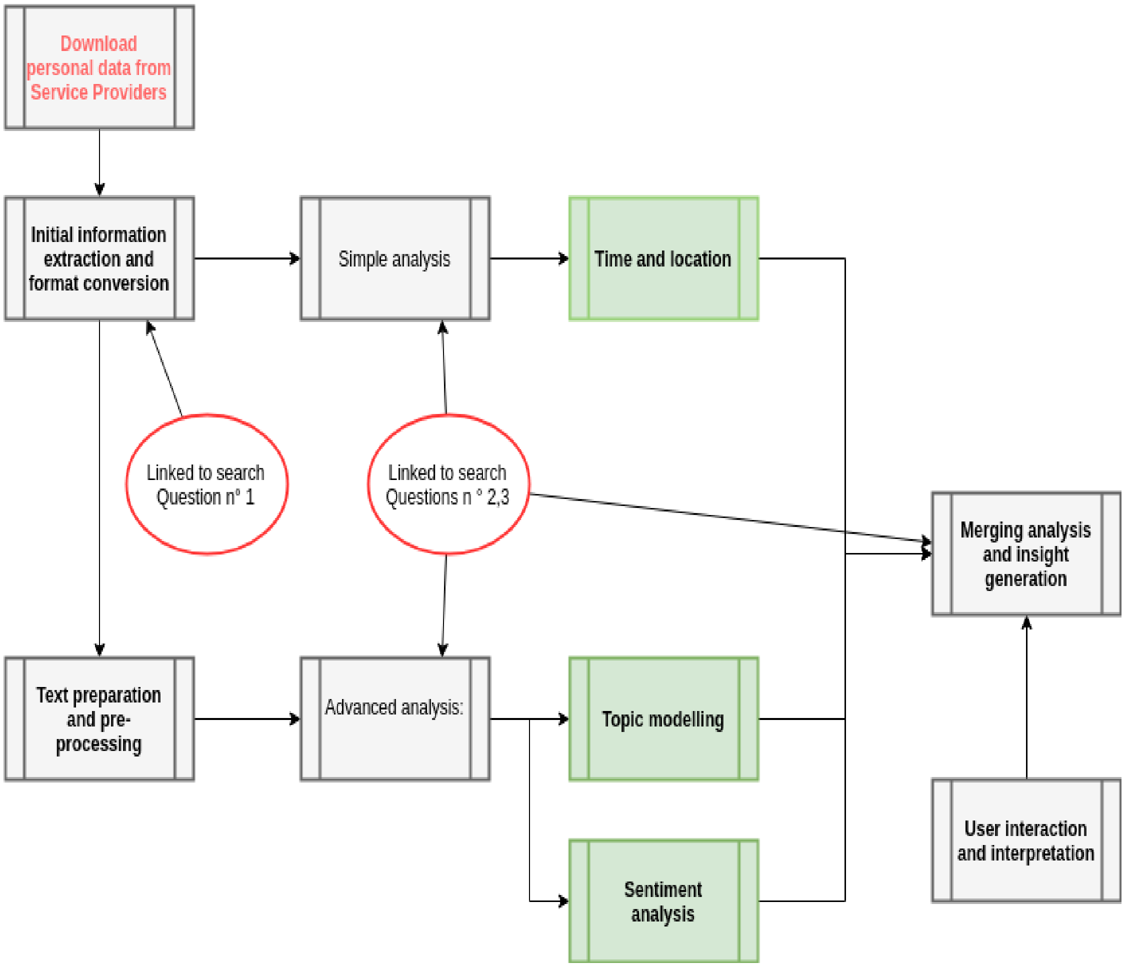

Download personal data from Service Providers
Initial information extraction and format conversion
Simple analysis
Time and location
Linked to search Question n° 1
Linked to search Questions n ° 2,3
Merging analysis and insight generation
Text preparation and pre-processing
Advanced analysis:
Topic modelling
Sentiment analysis
User interaction and interpretation

Figure 1.1: Application Step Representation

1.3.2 Contents of the Thesis

Chapter 2 contains the theory used as background in this project, the initial part of the chapter will present the main steps, performed in any Natural Language Processing problem, as data preparation and text pre-processing. After that it will present a theory recall of the machine learning algorithms used in this project, as Topic Modelling and Sentiment analysis, providing motivations for our choice and considerations.
Chapter 3 will explore more in the detail the aspect related to the information retrieval, presenting which choices are made for obtaining the initial data set, with a small consideration about the Service Providers considered for obtained those data; it will then move more in the initial analysis of the obtained data, their structure and format.
Chapter 4 is the main core of this project, presenting the analysis performed over our data, with a high focus on our proposal for a new Automated Topic Labeling.
This thesis will concluded with Chapter 5, were we provide our final considerations regarding the project and the obtained analysis while presenting some ideas, faced over the course of the project, for future works.

1.3.3 Derived information Consideration

In the following work, in all the examples that will be presented, no further explanations or interpretations of the obtained derived information will be presented: this choice was driven by the fact that, as in every data feature extraction and analysis, the final interpretations are left to the **Field of interest expert**; in our study case, by working with data obtained by the user, the **User** itself is considered our filed of interest expert, and by this definition, is the only one able to derived the final information, by merging the presented insight, and evaluate them.

Chapter 2

Background

As it will be presented in Section 3.3, the format of our data, linked to the Research question 1 and 2, correlates this work to a **Natural Language Processing** problem: this section will give an overview of the main steps taken from literature, while presenting the ML algorithms, respectively used for **Topic Modelling** and **Sentiment analysis**.

2.1 Topic Modelling

A topic modelling is a NLP approach that aims, by taking advantages of machine learning technology, to generate from a given corpora, the abstract topic presented inside; it can be generally performed in two different ways: *Supervised*, with the support of a well formed and well labeled corpus of data, used as a training and testing data set for the chosen algorithms, which, after the training part, will be able to CATEGORIZE the input corpora into one or more category present in the training label set; *Unsupervised* algorithm, which use a more statistical approach, and don't necessary need any training corpora, producing a statistical distribution of potential cluster/topics.
In our case, as explained before, by the absence of any label training corpora, the choice for an UNSUPERVISED approach was forced.

2.1.1 Data Pre-processing

The most important part, in any NLP analysis is related to the pre-process phase of the text of interest, this phase has the aim to prepare the text in order to have a good formatted and clean input for our machine learning algorithms.
This has been done by following several steps, each of them applying several transformation to our text.

StopWords, Punctuation

In any NLP problem, we have text composed by a set of words, belonging to one or more languages, when we are trying to retrieve some abstract information, by analyzing each words, not all of them are caring interesting information, as for instance the most common words used in a given language.
In our application is will be taken as a reference language, for commodity, English.
The next step is the removal of punctuation, or any other information that may interfere with our search, as removing numbers and any non standard digit: this is done in order to clean the words from any piece of information that could eventually bring some noise.
This step are performed due to the fact that input queries are not always well formed and sometimes they are made in a way that would make it difficult to understand an interpret them.

Lemmatization

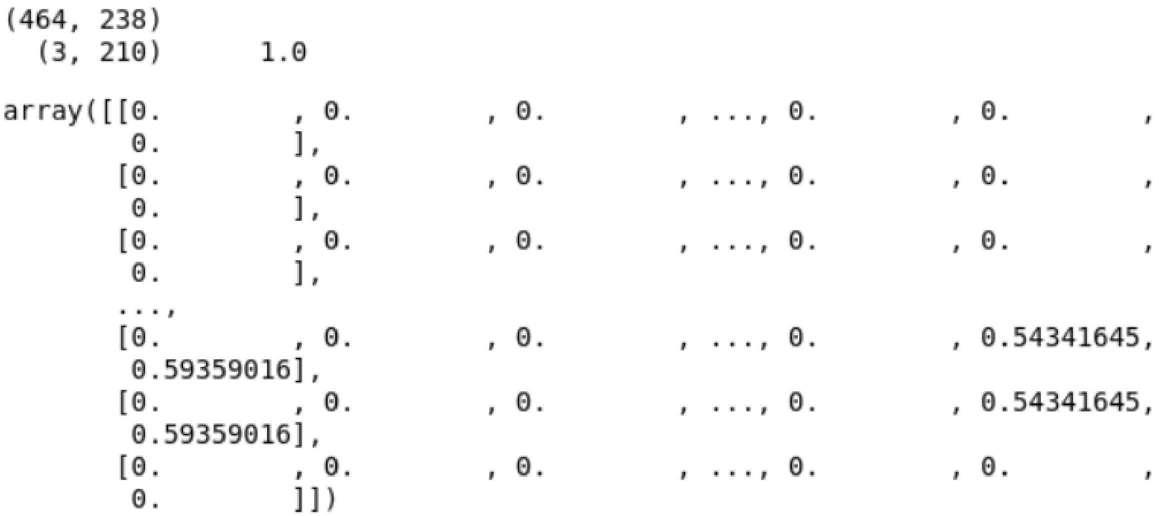
An important part usually performed in any text pre-proces phase, is the**Lemmatization**.
Sometimes, in text processing, it occurs that some words could derive from a common "root", the goal of Lemmatization is in fact to reduce those derived words to a common ancestor base form, that would lead to an improving of the future clustering of those given words. [4]
Lemmatization can be usually performed using a dictionary of words, in our case we take advantages of the **NTLK** that provide a module for Lemmatization, using, as a dictionary the **WorldNet** a wide lexical online database,for English language. [5].

2.1.2 Tokenization: Bag of Words, TfIdf factor

Now that we have a well formed series of cleaned text, we need to convert them into a proper input format for machine learning algorithm, this means converting them into a numerical feature vector.

Bag of Words

The bag of words procedural is the first method used for performing this conversion; this function, split all the words in our corpus of queries, considering each query as an independent corpus, and calculate the occurrences of that given words; finally it generates a matrix of feature number where: each row represent a document, in our case a query, and each elements in the row is related to the frequency of a given words.
Figure 2.1 and Figure 2.2, shows respectively the matrix format generated from the bag of word, and a chart representation of the most frequent words inside a set of queries.



```
(464, 238)
  (3, 210)      1.0

array([[0.         , 0.         , 0.         , ..., 0.         , 0.         ,
         0.        ],
        [0.         , 0.         , 0.         , ..., 0.         , 0.         ,
         0.        ],
        [0.         , 0.         , 0.         , ..., 0.         , 0.         ,
         0.        ],
        ...,
        [0.         , 0.         , 0.         , ..., 0.         , 0.54341645,
         0.59359016],
        [0.         , 0.         , 0.         , ..., 0.         , 0.54341645,
         0.59359016],
        [0.         , 0.         , 0.         , ..., 0.         , 0.         ,
         0.        ]])
```

Figure 2.1: Bag of Words matrix

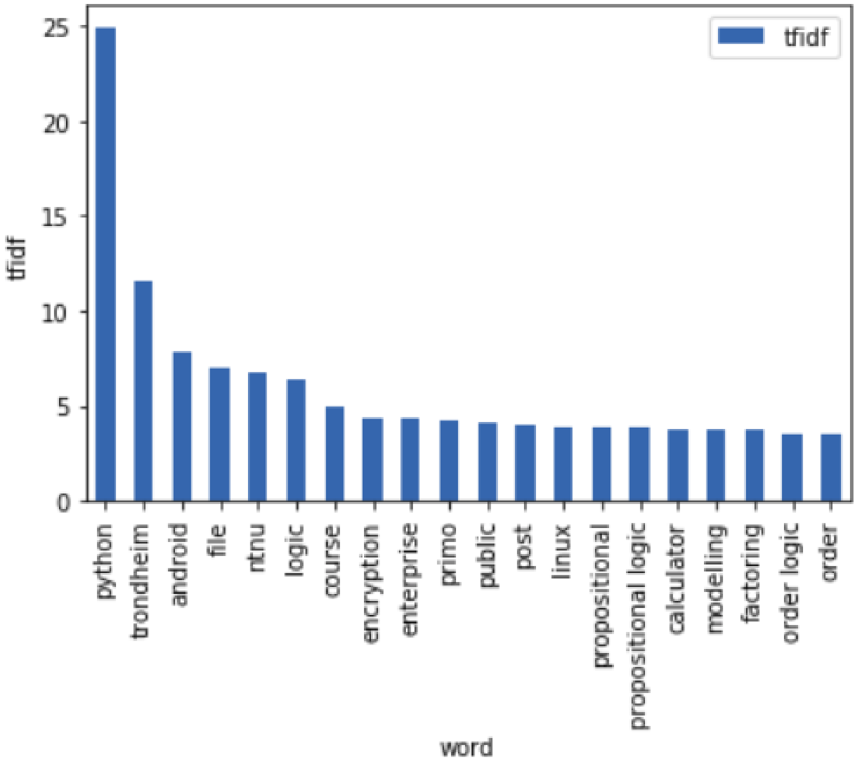


Figure 2.2: Bag of Words graph visualization

TfIdf factor

Some topic modelling algorithms as **Non Negative Matrix Factor** take as input another type of feature numerical vector, generated using a different approach: **Term Frequency inverse document frequency**, known as **TfIdf**.

This approach differs from the previous mentioned Bag of Words due to the fact that this factor tends not only to count the recurrences of a given words inside a text, but also distinguishing between valuable and less valuable words.

It generates two values for each words, it's occurrences inside a single document, and the occurrences across all document: a valuable word is the one that has a high number of occurrences in single documents, and doesn't occurs in many document. This is provided by the fact that usually the words that appear in a high number of document are the ones caring the less information.

An important input factor that has to be taken under consideration for those functions is the **n-gram**, a term that defines the range of words that the algorithms will use, as explained in Section 4.3.3.

2.2 Machine Learning algorithms for Topic Modelling

Related to topic modelling analysis, at the moment of the writing, there is a high variety of algorithms present in literature for this purpose; choosing one over the other is an evaluation strictly connected with the input data type, and may lead to the generation of better results.
For our case we decided to take under analysis the two major Algorithms present in literature: **Non Negative Matrix Factor** and **Latent Direchlet Allocation**.

2.2.1 Non Negative Matrix Factor

The main concept of this algorithm is the decomposition of a given matrix, in our case the tfidf value matrix (V), into two lower dimensional matrix named H, W, characterized by the present of only non negative factors, as follow

$$W * H = V$$

.
This approach allows the reduction of the feature space making easy to operate over them.[6].
It has been proven in literature the intrinsic usage of this approach for Clustering and Topic Modelling, due to the structure of the generated W and H matrix. [7]

2.2.2 Latent Direchlet Allocation

LDA is based on two mayor concepts: defining every document, that composes our corpora, as a distribution of topic, moreover considering each of those topic as a statistic distribution of words.
In order to understand its behavior it's often common to use a Plane Notation, the most common representation for probabilistic graphic model, useful to understand the dependencies between the model parameters. Figure 2.3

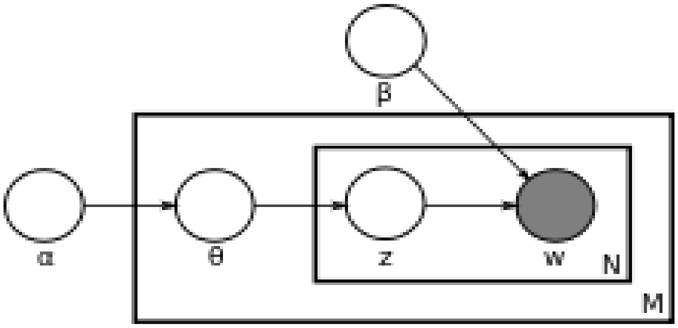


Figure 2.3: LDA Plane Notation

The first two major parameters are the two rectangles named with **M** and **N**, respectively they refer to the total number of document in our corpora, and the total number of words within a single document; this notation express the relation of the other parameters to one of those two level.

- α is the per-document topic distribution, it express the tendency to find within a single document, a high or low number of Topic: with a high value of α it will be most likely that every document will be a mixture of mostly every topic, while vice versa a low value of α says that each document will be compose of just a few of them.
- β is the pre-topic word distribution, express, on the other hand, the same tendency but over the correlation between words and topic: a high value of β will say that every topic may contains a mixture of mostly the words in our corpora.

This two parameters are defined as the Direchlet parameters, and are two input value of the algorithm.
The other two parameters θ and ζ, refers respectively to the topic distribution per document m_i and the topic of the $_l$ words in the document m_i; finally W stands for a single word. [8]
A final consideration about the good performance of LDA is related to the defined

values of α and β which are strongly related to the input, data; in fact those value are defined with the distribution of topic we are expecting to find on our corpora.

2.3 Parameters evaluation

2.4 Topic Labeling

One of the biggest challenges faced over the course of this project was, once obtained the Clustered Topics, how to label them.
In fact the labeling process is usually performed in a SUPERVISED way, using a pre-labeled reference data-set.
Due to the fact that no training data set was present, it was necessary to work in an UNSUPERVISED way, proposing a new solution, as it will be presented in Section 4.2.2

2.5 Accuracy Evaluation

2.5.1 Point Wise Mutual Information

In literature [9] the evaluation of Topic algorithms in Natural Language Processing is performed using different methodologies. All the coefficient used in literature rely on the evaluation of the so called **Point wise mutual Information**, a coefficient that express the likeliness of tow words to happen in the same corpora, in the same context, and is calculated as follow:

$$pmi(x,y) = \log \frac{p(x,y}{p(x) * p(y)}$$

Where p(x,y) is the probability to find both the two words x,y in a external reference corpora, while p(x),p(y), are their respective single probability.
The calculation of the above coefficient is made by taking advantages of an external prepared corpora, as a reference; in our case, the evaluation of this value was not possible due to the fact that, in the first place no external corpora was available; and using the query corpora, by the fact of being an unstructured corpora, has lead to the generation of values not caring any useful information, since a combination of two words, is most likely to appears just once inside the whole corpora.

2.5.2 Human Judgment Based Evaluation (HBE)

One common way to evaluate a topic cluster generation is the so called **Human Judgment Based Evaluation**, this method is based on the evaluation of the given topic by a human perception by actually seen how the words were clustered and which label was linked to them; a good way to perform this evaluation is to use graphic tools, as pyLDAvis graphic library for the evaluation of LDA modelling algorithm.

For the above considerations, there was the need of a new evaluation metrics to use beside the HBE. This new metric, called **Evaluation Using Generated Label Categories (GLC)** will be presented in Section 4.3.5.

2.6 Machine Learning for Sentiment Analysis

Sentiment analysis is a growing field of NLP that aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event [10].

Regarding the purposes of this project, by having a set of messages obtained from Facebook, each of them tagged with a time-stamp value, the idea is to generate an emotional overview, in a time-based scenario, of the user.

Sentiment Analysis can be seen as a classification problem, were a classifier is trained using a labeled corpora data-set. At the moment of the write, a solid an well documented way to perform sentiment analysis is by taking advantages of Twitter as a corpora data set [11].

In fact twitter provides a solid and wide corpora for sentiment analysis; according to this task there are present a set of different API, useful for generating a corpora from Twitter as *Python Twitter Tools* [12] or *Python Twitter API* [13].

For our case, we used a labeled corpora obtained from the source [11], composed of:

- **1393** negative-labeled twitters
- **635** positive-labeled twitters

2.6.1 Data an Text Pre-Processing

In order to improve the performances of the given algorithm, all the steps expressed in Section 2.1.1, were applied both to the training corpora, and to the single entry to label.

2.6.2 Classifier algorithm

A sentiment analyzer can be build using, as a main core classifier, a variety of different algorithms, as NAIVE BAYESIAN, OR SUPPORT VECTOR MACHINE ECC; for our case we decided to use the **Naive Bayesian**.

This decision was driven by that fact that, although its simplicity, Naive Bayesian proven to return satisfactory result, with a good rate of accuracy, recall and precision, as it will be shown in Section 4.5.

Naive Bayesian

Naive Bayesian algorithm is a conditional probabilistic theorem express as follow:

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}$$

given a vector of features $C_k = (k1, k2...)$, in this case the result obtained by applying a Bag-Of-Word method to our training data set, and a set of classes $X = (x1, x2...)$ we have:

- $p(C_k)$ as the prior probability of a single word in our corpora
- $p(x|C_k)$ as the conditional probability that given a class x a word k belongs to it
- $p(x)$ as the prior probability of our given classes, used as a constant for normalizing the result

The above equation can be written in the following way, for a better comprehension:

$$posterior = \frac{prior * likelihood}{evidence}$$

[14][15].

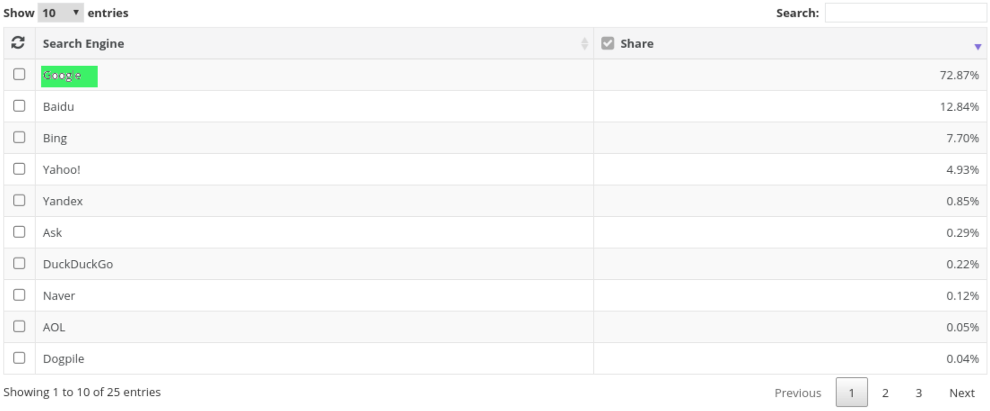
Chapter 3

Data Resources, characteristic and initial considerations

In any data analysis project a high initial focus is made around the understanding of the type of data the application has to work with, and which type of information are they caring.
This chapter will cover the initial process to retrieve those raw data, a briefly explanation on their structure, on their contents, while finally selecting the most relevant ones for performing a insight search over them.

3.1 Service Providers

Right now there are a high number of services that are collecting our information, storing them in different ways and in different formats. The first decision that was mandatory to be made in order to give an initial direction to this job, in this wide range of information, was to define from which services to retrieve the initial pool of information.
For this purpose two aspect were taken under consideration: the *type* of information stored by each candidate service provider, and the *usage*, defined as user usage in a fixed time unit.
Following those two discriminating factors, as it was predictable, the two major service provider are: **Google** and **Facebook**; in fact both of them are storing a high number of information about a single user that, eventually, may lead to some derived sensitive information, such as user behavior or interests; while as shown in Figure 3.1 and Figure 3.2, they are the most widely used Service Providers in their respectively area.



Show 10 ▾ entries

Search:

	Search Engine	☑ Share
□	Google	72.87%
□	Baidu	12.84%
□	Bing	7.70%
□	Yahoo!	4.93%
□	Yandex	0.85%
□	Ask	0.29%
□	DuckDuckGo	0.22%
□	Naver	0.12%
□	AOL	0.05%
□	Dogpile	0.04%

Showing 1 to 10 of 25 entries

Previous 1 2 3 Next

Figure 3.1: Google Monthly Usage [1]

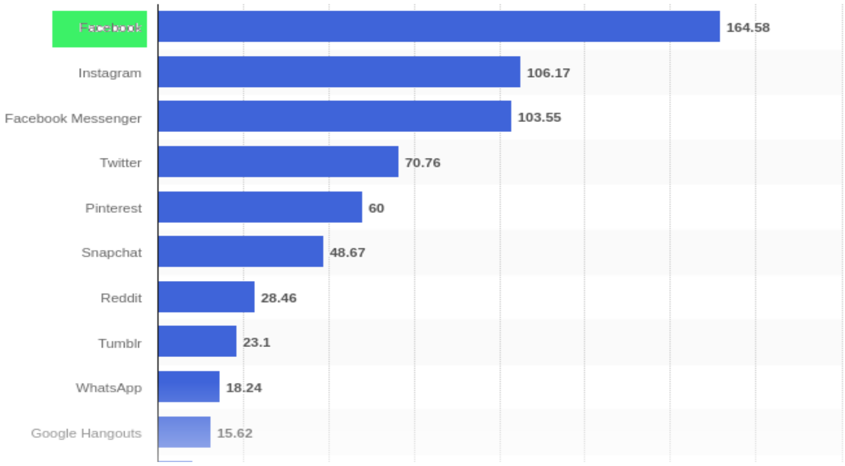


Figure 3.2: Facebook Monthly Usage [2]

3.2 Data Resources and Characteristics

This section presents briefly the steps needed to be performed in order to initially obtain, from the previously mentioned Services, the information needed as an input.

3.2.1 Data Download

The main information used for the analysis were the benchmark of information collected from GOOGLE and FACEBOOK: the process for requesting those information are present respectively in a dedicated page [16] [17];from here a series of file are generated and, within a time window, that could go from couple of days to a week, sent to the user by email.

3.2.2 Format of the Obtained Data

A first initial analysis has been done above those obtained data, to see their folder structure, file type and inner data organization structure.

Google Datas

Regarding the information obtained from the Google Service, those are organized in a folder hierarchy, where a single folder is present for each of the major Google service, some containing configuration files in a JSON format, other containing navigation HTML files. Figure 3.3.

Among those pool of data, the folder containing the most relevant ones, from a user point of view, is the folder named **My Activity**: this folder, following the hierarchy, contains one folder for each USER INPUT QUERIES services, as: *Google search engine queries, Images query, Google maps query*; each of them referred with a HTML file. Figure 3.4

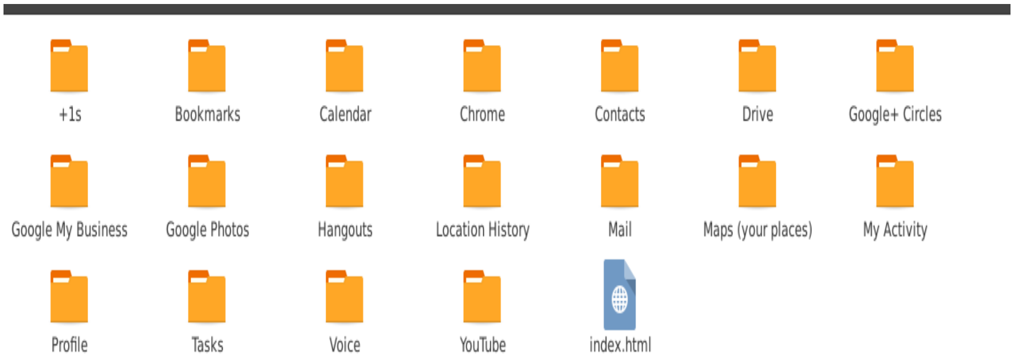


Figure 3.3: Google first directory structure

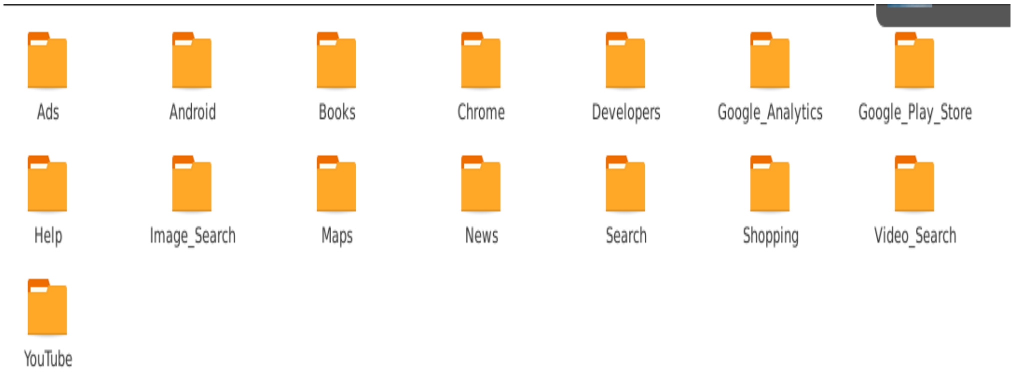


Figure 3.4: My Activity

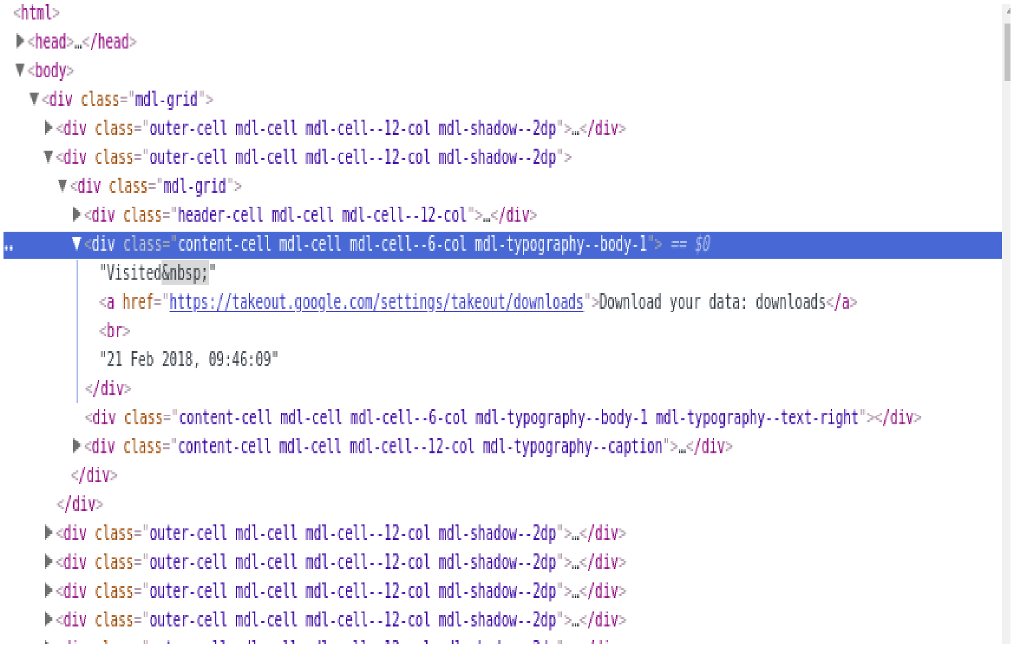
Data Structure

Each ones of the above mentioned files have, with some degrees of differences, the same HTML inner TAG hierarchy, structured as follow as shown in Figure 3.5:for each user input entry it is present a div tag named with the class *outer-cell mdl-cell mdl-cell–12-col mdl-shadow–2dp*, within this div tag a set of the inner div tag are present each one of them containing different information:

- **header-cell mdl-cell mdl-cell–12-col**:
  Here it's store the type of the entry: *Visited* or *Searched*

- **content-cell mdl-cell mdl-cell–6-col mdl-typography–body-1**:
  Here it's stored the effective content of the entry itself, and the time-stamp

- **content-cell mdl-cell mdl-cell–12-col mdl-typography–caption**:
  More footage information.



```
<html>
▶<head>…</head>
▼<body>
  ▼<div class="mdl-grid">
    ▶<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">…</div>
    ▼<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">
      ▼<div class="mdl-grid">
        ▶<div class="header-cell mdl-cell mdl-cell--12-col">…</div>
...     ▼<div class="content-cell mdl-cell mdl-cell--6-col mdl-typography--body-1"> == $0
            "Visited&nbsp;"
            <a href="https://takeout.google.com/settings/takeout/downloads">Download your data: downloads</a>
            <br>
            "21 Feb 2018, 09:46:09"
          </div>
          <div class="content-cell mdl-cell mdl-cell--6-col mdl-typography--body-1 mdl-typography--text-right"></div>
        ▶<div class="content-cell mdl-cell mdl-cell--12-col mdl-typography--caption">…</div>
        </div>
      </div>
    ▶<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">…</div>
    ▶<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">…</div>
    ▶<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">…</div>
    ▶<div class="outer-cell mdl-cell mdl-cell--12-col mdl-shadow--2dp">…</div>
```

Figure 3.5: HTML tag hierarchy

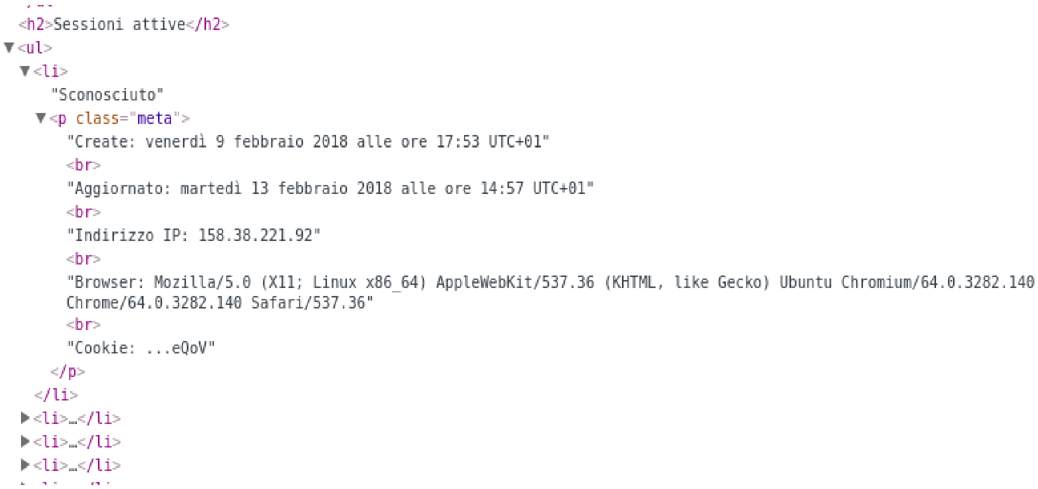
Facebook Data

Differently the content of the information obtained by Facebook, is a set of HTML navigation file, related to the major aspect of the service, as *messages,security,events,adds*, each one of them following its specific inner tag hierarchy.
For the sake of simplicity, it will be explained the structure of two files:**Security.html** and **Messages.html**, which are caring the most interesting information,used in the course of this work.

Data Structure

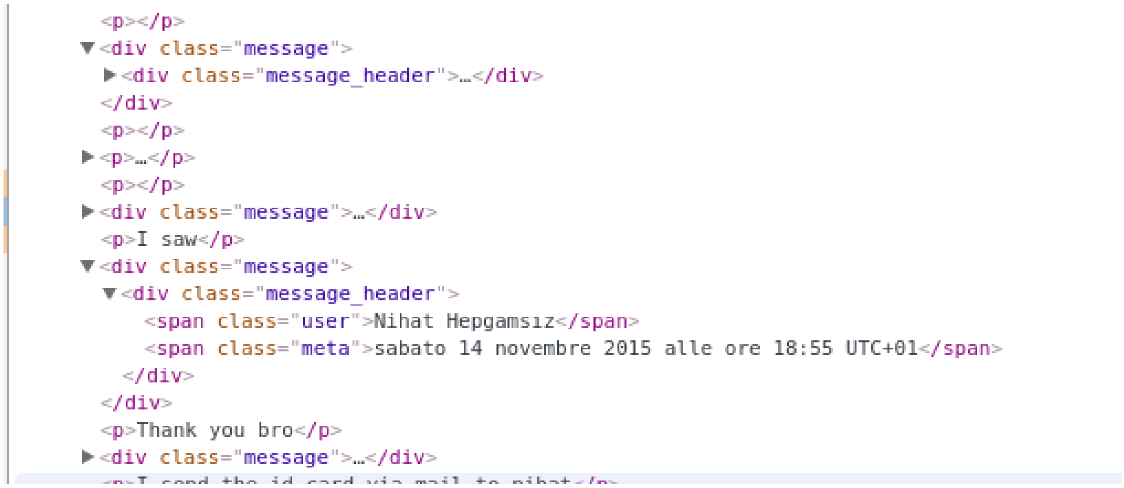
- **security.html**: Inside this files are present all session control information, as session activation,update and end, time-stamp, IP position, cookies, used devices.
  This given file is characterized with a complex and not fixed tag hierarchy. Figure 3.6

- **Message.html** Regarding this set of files, one single file is present of each single messenger conversation or group conversation, each of them with the same tag structure: a list of div labeled *Message* for every entry messages, containing inside two tag: *message-header*, with information regarding the sender of the given message and the time-stamp; end a *p tag*, with the contents of the message.Figure 3.7



```
  <h2>Sessioni attive</h2>
▼<ul>
  ▼<li>
      "Sconosciuto"
    ▼<p class="meta">
        "Create: venerdì 9 febbraio 2018 alle ore 17:53 UTC+01"
        <br>
        "Aggiornato: martedì 13 febbraio 2018 alle ore 14:57 UTC+01"
        <br>
        "Indirizzo IP: 158.38.221.92"
        <br>
        "Browser: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/64.0.3282.140
        Chrome/64.0.3282.140 Safari/537.36"
        <br>
        "Cookie: ...eQoV"
      </p>
    </li>
  ▶<li>…</li>
  ▶<li>…</li>
  ▶<li>…</li>
```

Figure 3.6: Security HTML structure sample

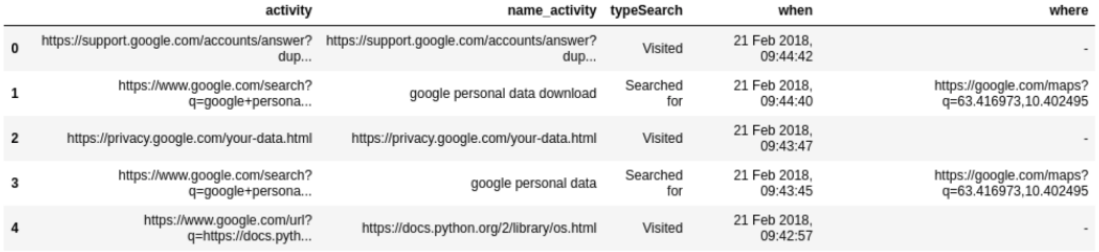


```
    <p></p>
  ▼<div class="message">
    ▶<div class="message_header">…</div>
    </div>
    <p></p>
  ▶<p>…</p>
    <p></p>
  ▶<div class="message">…</div>
    <p>I saw</p>
  ▼<div class="message">
    ▼<div class="message_header">
        <span class="user">Nihat Hepgamsız</span>
        <span class="meta">sabato 14 novembre 2015 alle ore 18:55 UTC+01</span>
      </div>
    </div>
    <p>Thank you bro</p>
  ▶<div class="message">…</div>
    <p>I send the id card via mail to nihat</p>
```

Figure 3.7: Message HTML structure sample

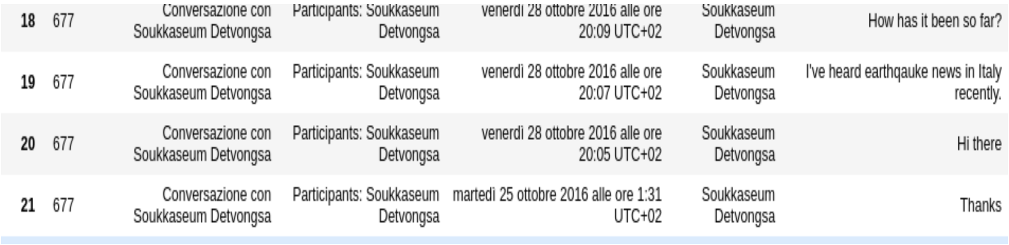
3.3 Format Conversion

Before performing any kind of analysis over the above obtained data-set, a first conversion, to a more suited format for data extraction, is needed: **Comma-Separated-Values**, also referred as **CSV** [18], was chosen due to its fixed structure. Figure 3.8 shows the CSV generalized by *Google Search Queries*, while Figure 3.9 is generated from one Facebook chat message.



	activity	name_activity	typeSearch	when	where
0	https://support.google.com/accounts/answer?dup...	https://support.google.com/accounts/answer?dup...	Visited	21 Feb 2018, 09:44:42	-
1	https://www.google.com/search?q=google+persona...	google personal data download	Searched for	21 Feb 2018, 09:44:40	https://google.com/maps?q=63.416973,10.402495
2	https://privacy.google.com/your-data.html	https://privacy.google.com/your-data.html	Visited	21 Feb 2018, 09:43:47	-
3	https://www.google.com/search?q=google+persona...	google personal data	Searched for	21 Feb 2018, 09:43:45	https://google.com/maps?q=63.416973,10.402495
4	https://www.google.com/url?q=https://docs.pyth...	https://docs.python.org/2/library/os.html	Visited	21 Feb 2018, 09:42:57	-

Figure 3.8: Google CSV-DataFrame



18	677	Conversazione con Soukkaseum Detvongsa	Participants: Soukkaseum Detvongsa	venerdì 28 ottobre 2016 alle ore 20:09 UTC+02	Soukkaseum Detvongsa	How has it been so far?
19	677	Conversazione con Soukkaseum Detvongsa	Participants: Soukkaseum Detvongsa	venerdì 28 ottobre 2016 alle ore 20:07 UTC+02	Soukkaseum Detvongsa	I've heard earthqauke news in Italy recently.
20	677	Conversazione con Soukkaseum Detvongsa	Participants: Soukkaseum Detvongsa	venerdì 28 ottobre 2016 alle ore 20:05 UTC+02	Soukkaseum Detvongsa	Hi there
21	677	Conversazione con Soukkaseum Detvongsa	Participants: Soukkaseum Detvongsa	martedì 25 ottobre 2016 alle ore 1:31 UTC+02	Soukkaseum Detvongsa	Thanks

Figure 3.9: Facebook CSV-DataFrame

Data Consideration

Now that the information are expressed in a more suited format, it is possible to see that there are two type of information:

- **background** information, as *Time-stamp* and *Location-stamp* expressed in their respective format, which will be used as input for the basic statistical analysis.

- **text-based** information, the most relevant ones as engine queries, messages, that will be used as input for the advance analysis.

Regarding the *Research Problem n°2*, by taking this as a starting point, our search naturally relates to a **Natural Language Processing** problem, known also as **NLP**.
With the terms of **NLP** in literature are referred all the technologies and methodologies that aim to analyze any text-based information, as a speech or text extract, generating from them derived or abstract information, as topic modelling, text summarizing... [19].
As it will be express in Chapter 3, for the aims of this work two different approaches, taken from NLP, will be used: **Topic Modelling** and **Sentiment analysis**.
We choose this two approaches for this work, due to the fact that, with our input data, we aimed to obtain information as:

- User topic of interest, in a location or time based scenario, by taking advantages of the pool of search queries

- Sentiment behavior over time, by using the pool messages from Facebook.

3.4 Development Environment

This section will present the software environment used for the developing of this project.

3.4.1 Working Environment

The following project has been entirely developed under **Python 2.7**[20], this choice was leaded by the fact that, in the moment of the writing, python is the most used programming language for data science and data analysis, with a vast and well documented libraries that goes from data manipulation to machine learning algorithms, making it the most suited choice.
As working environment it was used **Jupyter Notebook** [21], a command shell for interactive computing, which is the fast and more responsive environment for Data Analysis.

3.4.2 Libraries

The main core of the project was build around five major python libraries for data analysis and machine learning:

- NUMPY:library designed for high dimensions array and matrix manipulation. [22]

- PANDAS: library offering tools for data manipulation and data analysis, as table of values. [23]

- SCIKIT-LEARN: library containing a high range of machine learning operation, designed to inter-operate with numpy. [24]

- NATURAL LANGUAGE TOOLKIT: library optimized for natural language processing of text. [25]

- BEAUTIFULSOUP: a library for HTML file parsing. [26]

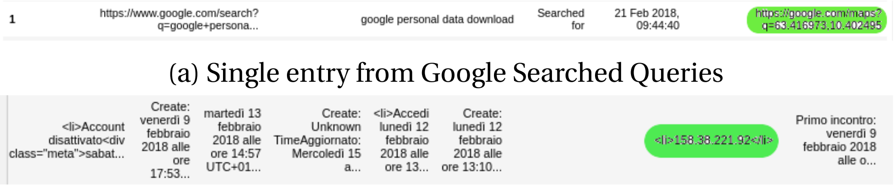
Chapter 4

Insights Search and feature extraction

After the outline of the input information and their conversion to a proper standard structure for feature extraction, this Chapter will cover our search for insights.
Section 4.1 will cover the simple analysis, as time and location extraction, that will be used for obtaining an initial information, consequently those information will be used as based-criteria for the more advance analysis. Section 4.2 presents all the aspect related to the Topic Modelling process performed over the information obtained from Google. In particular a high focus will be putted over the evaluation and selection of the most suited, out of the machine learning algorithms mentioned in Chapter 2, for our case. At the same time in the above Chapter we will present our method for Automated Labeling of a given set of cluster of words, followed with its evaluation. The final result for Topic Analysis over different based-criteria scenario it present in Subsection 4.4.
Section 4.5 will be focus over the result obtained out of the application of the designed Sentiment Analyzer out of the pool of massages obtained by Facebook, presenting different based-criteria scenarios.

4.1 Time and Location

As shown in Figure 4.1,both our input data set are caring relevant information regarding **time** and **Location**, expressed in the case of Google as a set of coordinate, while for Facebook as an IP Address; this section will explore the process and tools used for this purpose.
For the sake of simplicity, due to the fact that this analysis are performed in the same way over the two data-set, it will be taken as reference example the data-set from Google over the course of this section.



1	https://www.google.com/search? q=google+persona...	google personal data download	Searched for	21 Feb 2018, 09:44:40	https://google.com/maps? q=63.416973,10.402495

(a) Single entry from Google Searched Queries

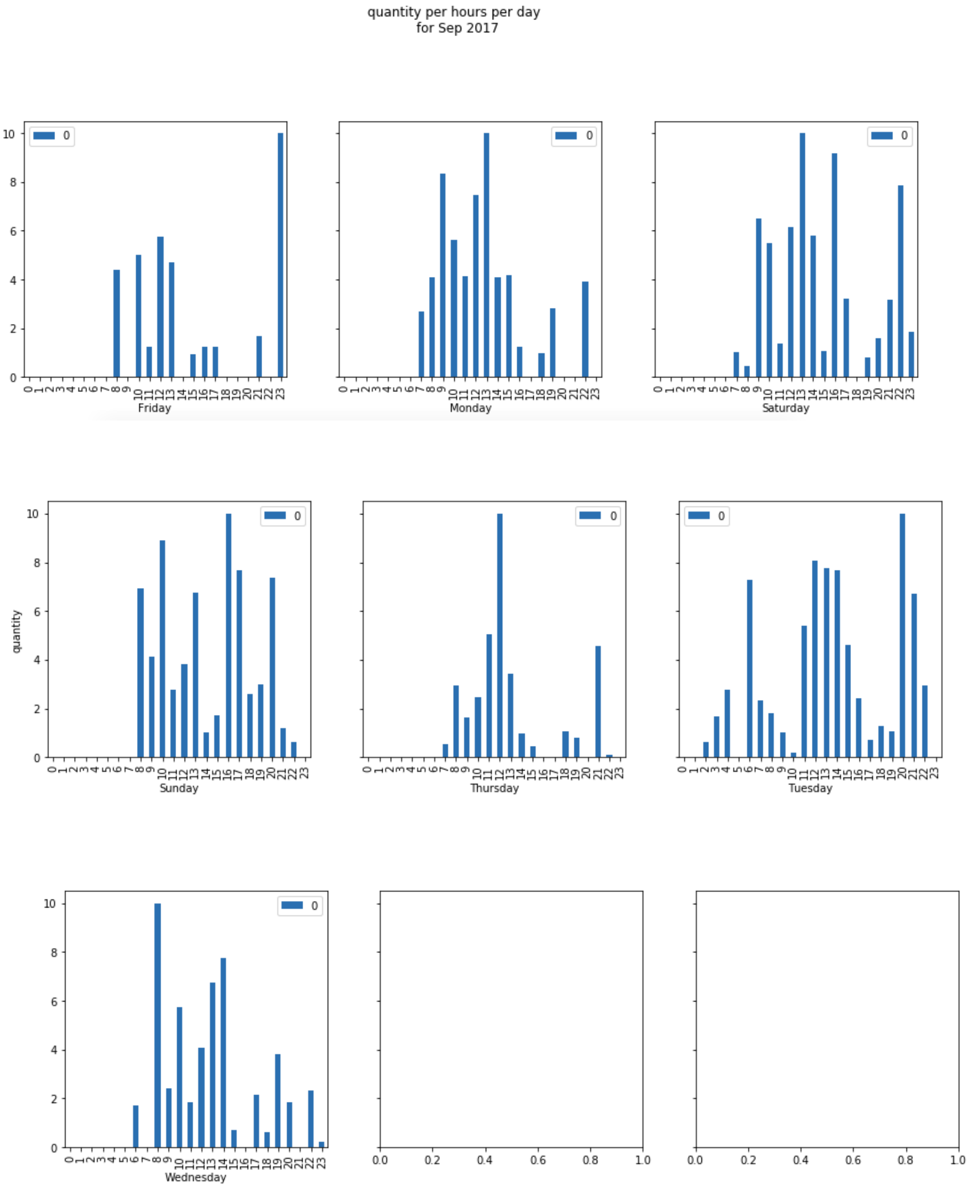
<li>Account disattivato<div class="meta">sabat...	Create: venerdì 9 febbraio 2018 alle ore 17:53...	martedì 13 febbraio 2018 alle ore 14:57 UTC+01...	Create: Unknown TimeAggiornato: Mercoledì 15 a...	<li>Accedi lunedì 12 febbraio 2018 alle ore 13...	Create: lunedì 12 febbraio 2018 alle ore 13:10...		<li>158.38.221.92</li>	Primo incontro: venerdì 9 febbraio 2018 alle o...

(b) Single entry from Facebook Security Log

Figure 4.1: Single Entry samples

4.1.1 Time Analysis

As mention in Section 3.2.2, for every query is present a time-stamp information, referring to the moment where the given query was created by the user, in the following format 21 FEB 2018, 09:44:42; the first information we looked for was the time usage consumption performed by the user over this service, considering as a reference unit a given hour for every day of the week in a given month.
According to the fact that we are dealing with a time-stamp format, the initial operation to perform is retrieving, for every data, its linked day of the week; this was performed thank to a python library designed for performing simple and complex time operation called DATETIME [27].
Then, by considering every occurrence of a given hour of a day, for a given day of the week, an initial time usage is performed.
Due to the fact that, as it was expected, this generated a wide range of value, a *Normalization* process is applied over this obtained values, for a better reading, fixing their range to [0-10].
Figure 4.2 shows the result over a given month.
By using the same process it is also possible to retrieve a more general information about the monthly usage of the given service, as it is shown in Figure 4.2


quantity per hours per day
for Sep 2017
0
quantity
0
2
4
6
8
10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
0.0
0.2
0.4
0.6
0.8
1.0

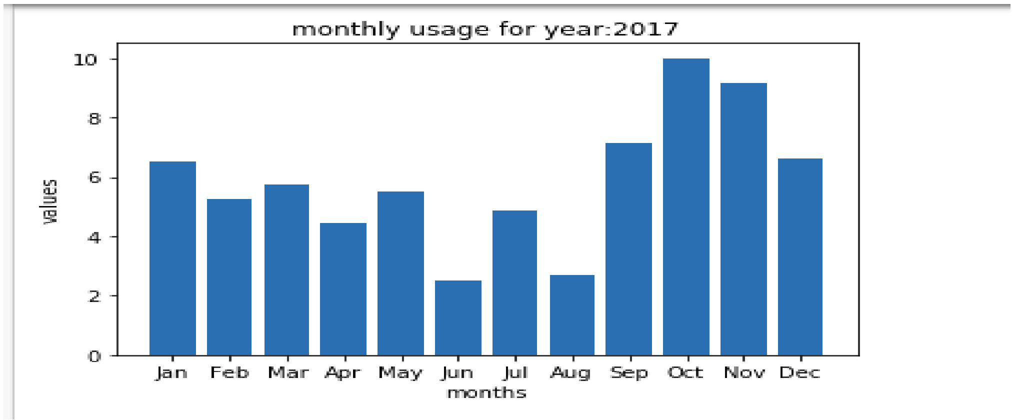


Figure 4.2: Monthly usage

4.1.2 Location Retrieval

As mention in Section 3.2.2, by considering the information stored in the file *Search*, for the *Searched Queries* is present a location stamp: by taking advantages of a location library called **GeoPy** [28], it is possible to retrieve information as **Country,City,Zip Code, Address**.

Figure 4.3 shows a sample, where the two charts represents the Nation visited on a given month, and the cities for those nations, where the user have been.

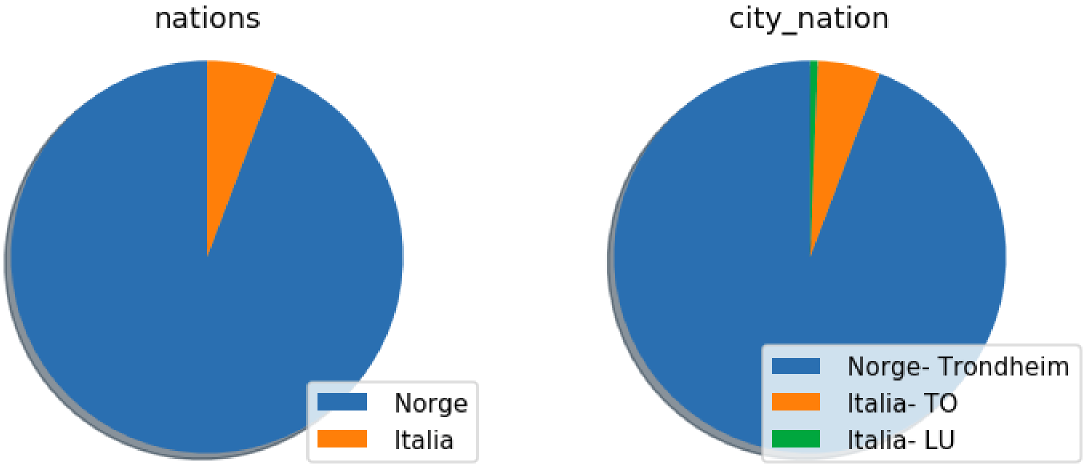


Figure 4.3: Location Sample

4.2 Topic Analysis over Google Search Queries

4.2.1 Data Preparation

By having a wide and large set of input queries performed by the user, in order to improve the performances of our topic modelling algorithm and avoiding a to high rate of sparsity of information, it is necessary to split the information according to a given criteria: usually given set of queries, they prof to have a context relation on a daily based criteria, this means that queries performed over the course of a given day are more likely to be connected. This criteria, in the other hand, generated a to small data set for our algorithms, and same happens with a weekly based criteria: for this reason it was taken a **monthly based** criteria for initially divide our data.
At the same time, as it was shown in Section 3.3, inside our obtained data, beside the **Searched pages** there is another additional piece of information useful for increasing the performances: the **Visited Pages**, expressed with the URL address of the mentioned page.
As Visited pages are considered all the web pages that the user opened from a third page, without using any search engine.
Usually the information present in this mentioned sites, in a time based criteria, are strictly linked to the queries performed in the same time based: for this reason this set of data will be used as support information, by retrieving the titles from this given URL, and ad them to our corpora data set.

4.2.2 Topic Labeling

The purpose of this topic modelling analysis was to obtain some information related to the user behavior and topic of interest, for this reason beside the topic modelling algorithms and parameters evaluation, it was mandatory to have a good tool for generating the most abstract labels out of the obtained generated topics.
This goal is usually achieve in a SUPERVISED way, using a given corpora of pre-labeled data, or in alternative using the most frequent words in every topic as label: the first solution is not feasible in our case, by not having any pre-labeled corpora data set; while the second one cannot be considered a good methodology for obtaining abstract labels.

Wikipedia for Automated topic Labeling

For the reasons explained above we use the ideas proposed in the articles [29] and [30], to take advantages, for topic Labeling, of Wikipedia.
In fact Wikipedia is organized over a top hierarchy of categories; if a query is performed over the Wikipedia engine, it returns a set of categories for that given search, as shown in Figure 4.4, for the query C++ [1].

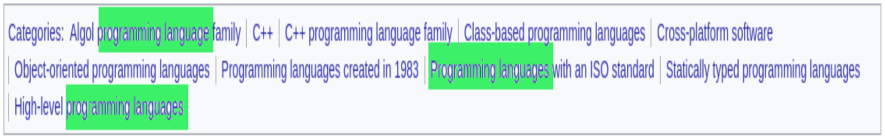

Categories: Algol programming language family | C++ | C++ programming language family | Class-based programming languages | Cross-platform software | Object-oriented programming languages | Programming languages created in 1983 | Programming languages with an ISO standard | Statically typed programming languages | High-level programming languages

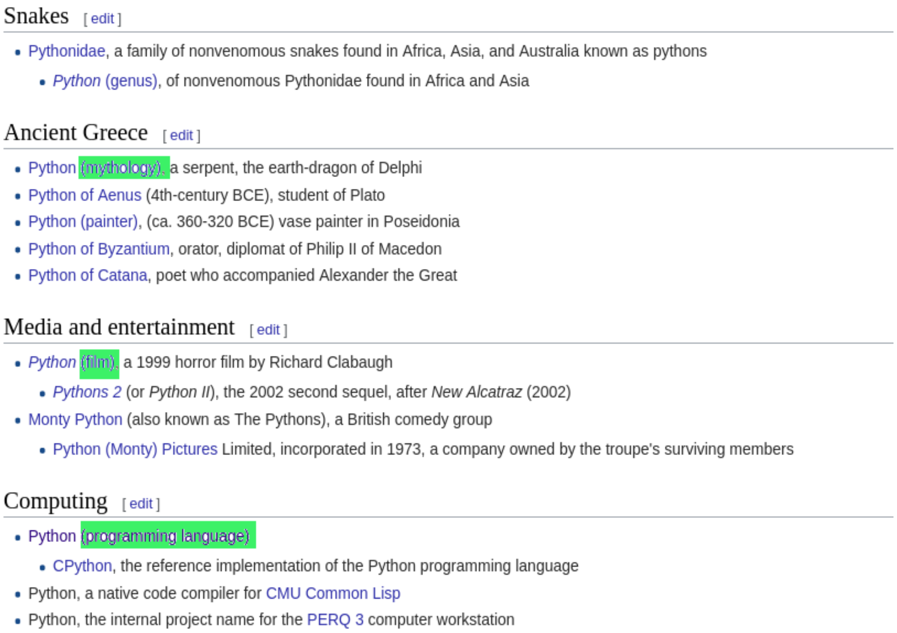
Figure 4.4: Wikipedia Category Example

As it can be seen in Figure 4.4, Wikipedia returns a series of categories from its hierarchy, that the sites relates to the performed query; it is possible to see that all the obtained 'categories' are, with some degrees,related to each other, in our sample to PROGRAMMING LANGUAGE.
Due to the fact that,as it will be presented, Wikipedia return different type of responses according to the query it receives, this first piece of information will be refer as **Wikipedia Standard Response**.

Disambiguation

Usually a given word doesn't belong uniquely to one category, but most likely to a set of not correlated ones: this problem is known as **Disambiguation**.
In a case of Disambiguation, Wikipedia returns a so called DISAMBIGUATION PAGE, Figure 4.5 presents the Disambiguation Page for the query PYTHON[2] ; as it can be seen, this returned page is composed as well with a list of categories, with a reference, inside bracket, to a more general one: taking advantages of this structure, and considering only the category inside brackets, it is possible to obtain some potential label candidates, or an enforcement for the categories already present in our pool of candidates.

[1] This query was taken from our sample queries
[2] another query taken from our sample



Snakes [edit]

- Pythonidae, a family of nonvenomous snakes found in Africa, Asia, and Australia known as pythons
  - *Python* (genus), of nonvenomous Pythonidae found in Africa and Asia

Ancient Greece [edit]

- Python (mythology), a serpent, the earth-dragon of Delphi
- Python of Aenus (4th-century BCE), student of Plato
- Python (painter), (ca. 360-320 BCE) vase painter in Poseidonia
- Python of Byzantium, orator, diplomat of Philip II of Macedon
- Python of Catana, poet who accompanied Alexander the Great

Media and entertainment [edit]

- *Python* (film), a 1999 horror film by Richard Clabaugh
  - *Pythons 2* (or *Python II*), the 2002 second sequel, after *New Alcatraz* (2002)
- Monty Python (also known as The Pythons), a British comedy group
  - Python (Monty) Pictures Limited, incorporated in 1973, a company owned by the troupe's surviving members

Computing [edit]

- Python (programming language)
  - CPython, the reference implementation of the Python programming language
- Python, a native code compiler for CMU Common Lisp
- Python, the internal project name for the PERQ 3 computer workstation

Figure 4.5: Disambiguation Page Sample

Page Not Found

Another interesting case, rises when Wikipedia get, as a query, a malformed entry or an entry that is not present inside its data-based: in this situation it returns a **Page Not Found**.

In our case, by dealing with user entry query, it can happen that a given entry is malformed or composed of a set of word that combined, generate a Page not Found, but singularly bring a valid result.

An example taken from our case is "python listdir", queering Wikipedia with this combination rises a Page not found, while the singular words bring a result.

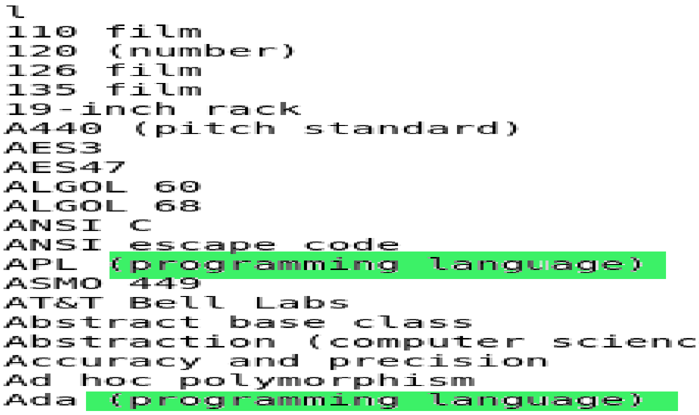
This problem is partially solved through the pre-processing text phase, due to the fact that are taken in consideration combination of n-gram words, where, in our case n-gram is defined as 2 (Section 4.3.3), this means that the algorithm considers combination of 1 and 2 words; but in the other hand, it may generate a new combination of two words, that were not meant to be together, leading to a potential Page not Fount case.

4.3 Algorithm operation

4.3.1 MediaWiki

All the queries to Wikipedia are performed using an ad-hoc library called **MediaWiki**, this library allows to retrieve, beside the type of information listed before, one more piece of information needed for out case study: a list of *related links*, for a given query, in the same format expressed in Section 4.2.2, with a related general hierarchical label present inside brackets for that given link.

Using as sample the same query performed previously, C++, the obtained links are listed in Figure 4.6; as it can be seen the obtained response contains a list of links to other Wikipedia pages, that the site considers belonging to the same categories cluster as our Query; in the example, it is possible to see, that as expected, some of the category obtained in the **Wikipedia Standard Response** from the same query are, as well, listed in this type of response, consequently this piece of information will be used by the system in order to let the algorithm to converge to a given category.



```
l
110 film
120 (number)
126 film
135 film
19-inch rack
A440 (pitch standard)
AES3
AES47
ALGOL 60
ALGOL 68
ANSI C
ANSI escape code
APL (programming language)
ASMO 449
AT&T Bell Labs
Abstract base class
Abstraction (computer science)
Accuracy and precision
Ad hoc polymorphism
Ada (programming language)
```

Figure 4.6: Related link

4.3.2 Algorithm Main Core

The algorithm proposed for automatic labeling a given topic/cluster of words works around this three type of responses: STANDARD WIKIPEDIA RESPONSE,DISAMBIGUATION RESPONSE and LINKS RESPONSE.
For every word inside a given cluster, a query to Wikipedia is generated and, according to the type of response the following considerations are made:

- Page Not Found: a counter called **nMiss** is incremented, this will be used to see how many words, inside that cluster, have not "participated" into the process, leading to a loss of information: a high value of this counter expresses that only a few words have actively brought a result, therefore the obtained label is not really correlated to the general topic of the cluster.
- Disambiguation Page:
- Wikipedia Standard Response: for this piece of information some more considerations are taken in account: initially all the entry present in the response, that are composed with at most three words, due to the fact that a higher number of words inside a given category leads to a lost of abstraction, the algorithms take as *Candidate Categories*, the entry itself and every combination of n-grams=2 words, inside that given entry, this is supported by the fact that, usually, a more abstract category might be contained inside a composed one.

Finally, due to fact that, by having as a starting data set a wide range of different unrelated information, it is most likely that in a given topic more than one category is widely present; for this reason, in order to keep track of this, the two most frequent categories are taken as labels for that given cluster.

Time Response Performances Considerations

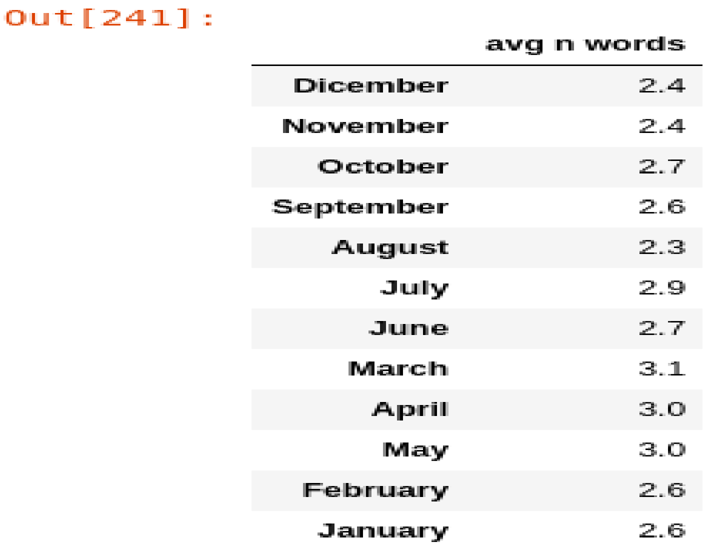
One aspect that has to be taken under consideration for this solution was related to the **Time exceeded** in order to accomplish the automatic labeling process, in fact by relying on queering Wikipedia, for a single word [3], a time delay is generate, before receiving the response estimated of **0,019 second** named t; by considering$M$ clusters, for each of them considering as a sample for generating the topic the N top rated words, the time exceeded in order to generate a cluster for a given month is:$T = M * N * t$.
Considering a case where 7 clusters are generated and the top 22 words for every cluster are taken under consideration, a time delay of $T = 3,08$ *minutes* is generated,
Due to the aims of this work, this issue is left for future works.

[3] performed over a Ethernet connection with 100 mbps

4.3.3 Parameters Evaluation

N-gram parameters

A critical step in the Topic modelling procedural is to select suited factors according to the corpora taken under analysis.
The first factor to take under consideration is the n-grams of words that the Tfidf and the world -count will consider as the granularity of the combinations of words, that would still caring useful information.
In order to define that value a consideration has to be made around the concept of query analysis; in fact inside a query we can expect that the granularity of the combination of words caring information should be up to the number of words composing that given query: for this reason, by taking as example the queries executed over the course of the year 2017, a first monthly average has been calculated, as shown in Figure 4.7; by calculation the average of the shown values, over the course the the considered year, a word average of **2.6** was retrieved. By using this obtained value as



Out[241]:

	avg n words
Dicember	2.4
November	2.4
October	2.7
September	2.6
August	2.3
July	2.9
June	2.7
March	3.1
April	3.0
May	3.0
February	2.6
January	2.6

Figure 4.7: Monthly words avg per query

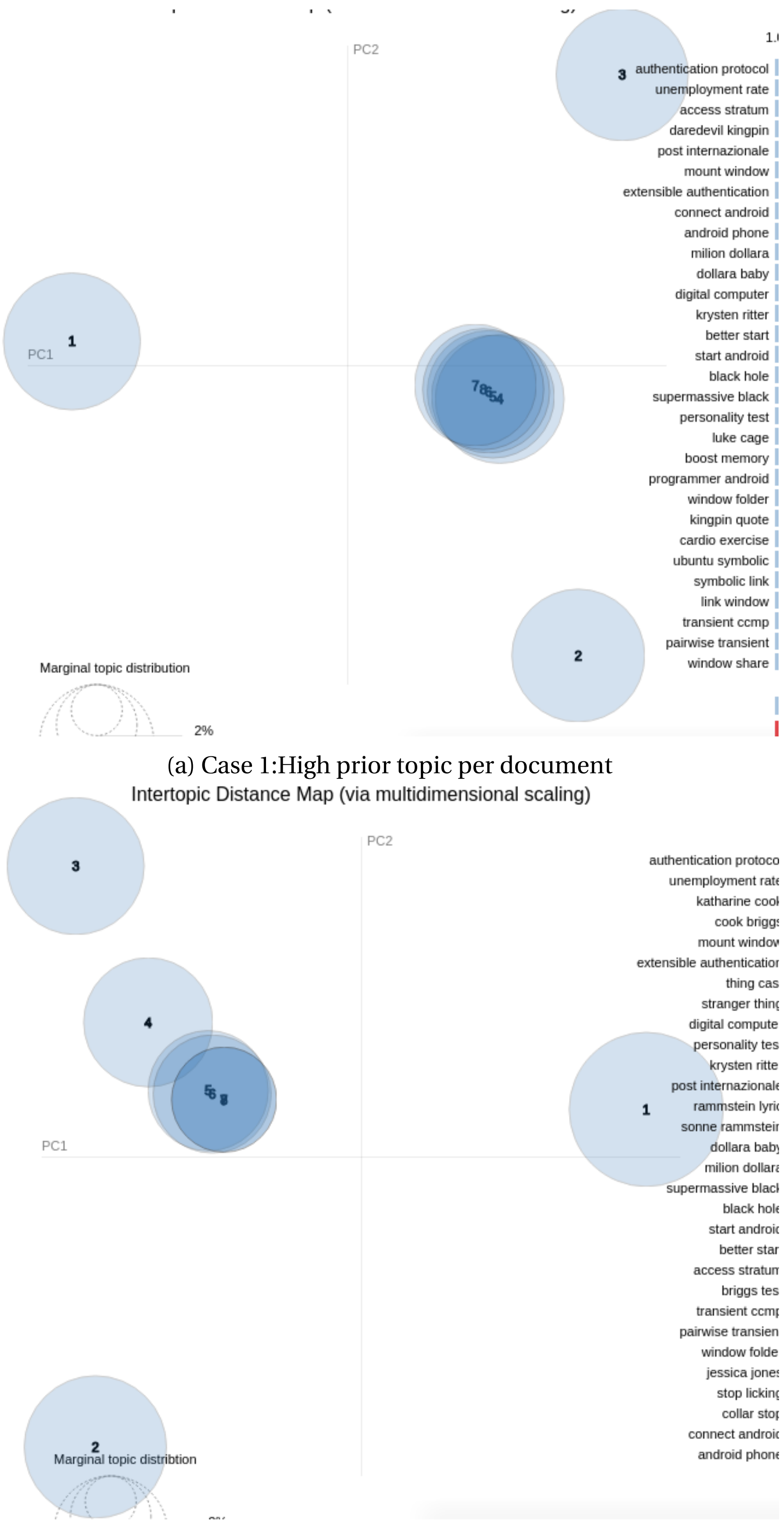
reference, the two integer candidate were **n-grams=**2 or 3; in our case we decided to take as a reference value n-gram=2, since it was generating better results.

4.3.4 LDA parameters

As explained in Section 2.2.2, in order to increase the performance of LDA algorithm it is necessary to considered a proper value of α and β.
Since no prior knowledge about the structure of our pool of queries was given, it was hard to make any assumption related to the Topic per document distribution and word per document distribution to expect: for this reason, by considering as *neutral* value $\frac{1}{T}$, where T is the number of the topic/cluster to generate,in our case defined as 8; we obtained three case studying, as follow

1. High value of α neutral value of β
2. High value of β neutral value ofα
3. Neutral value for α neutral value of β

Taking advantages of a visualization tool designed for LDA: **PyLDAdiv** [31], for each of the above cases, a visual representation has been generated and presented in Figure 4.8a, Figure 4.8b and Figure 4.9, respectively for our three study case.
PyLDAvis generates for each cluster a graphic circle, were the diameter of that given circle is related to the "size", defined as number of words belonging to that given topic: at the same time, if two or more 'circles' are overlapping, it means that the algorithm relates some words as belonging to both those topics, expressing a term of relation between those topic and leading to a lack of accuracy; in fact a pool of good generalized topics is defined as a set of not overlapping circles.
In our case we can see that all the three scenarios are presenting, with different degrees, some overlapping among the generated *'Topics/circles'*, thereafter is natural to see that the case of both neutral values, generates the highest number of unrelated topics.



(a) Case 1:High prior topic per document

(b) case2: High prior words per topic

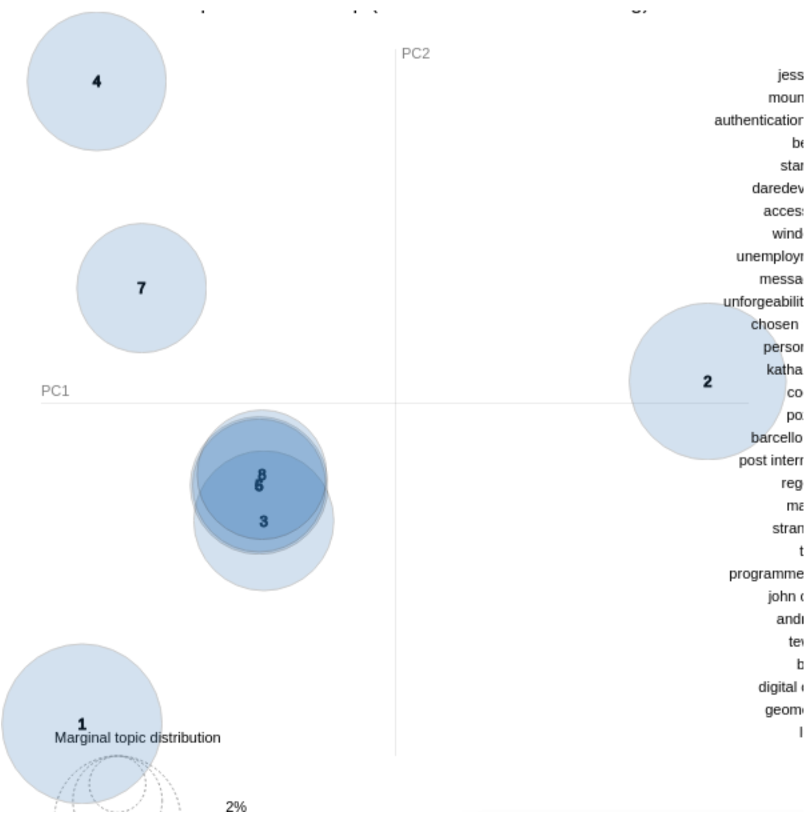


Figure 4.9: Case 3: Neutral state

4.3.5 Accuracy Evaluation Proposed Method

Evaluation Using Generated Label Categories (GLC)

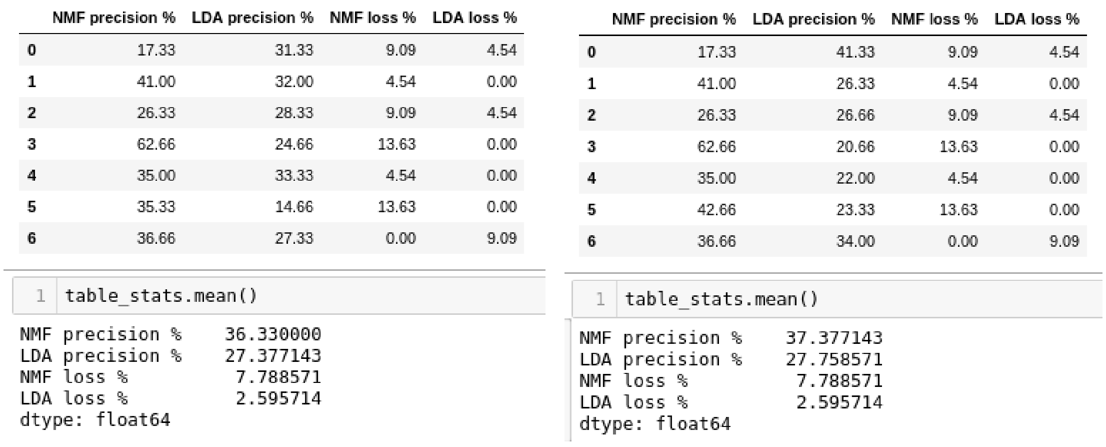
In order to evaluate the precision for our problem, we designed a new term composed by a combination of factors is going to be used: the main concept is that in a good cluster it is most likely to have words belonging to the same category, so the top candidates categories should have a quite good likelihood for that given cluster; in order to see the likelihood we are going to use the following formula:

$$\frac{\sum_{i=0}^{n} f_i}{N}$$

where f_i is the frequency for the top n categories and N is the total number of candidate categories for that given cluster.
In order to have a normalized value, N is a fixed value chosen from the user, in our case taken as 20, a value generated as the mean of the average number of total candidate categories. The final evaluation is performed combining this factor with the *nMiss* value, for having a sense of the loss of information.

4.3.6 Model Comparisons

This section is going to present a comparison between the performances, evaluated using a combination of **HBE** and **GCL**, on NMF and LDA algorithms in our study case. In the following part the two algorithms have been run over the same sample of data, for 6 iterations, in order to see the differences on precision and stability of the algorithms. The example above shows the accuracy calculate over the different clusters



	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	31.33	9.09	4.54
1	41.00	32.00	4.54	0.00
2	26.33	28.33	9.09	4.54
3	62.66	24.66	13.63	0.00
4	35.00	33.33	4.54	0.00
5	35.33	14.66	13.63	0.00
6	36.66	27.33	0.00	9.09

```
table_stats.mean()
```

```
NMF precision %    36.330000
LDA precision %    27.377143
NMF loss %          7.788571
LDA loss %          2.595714
dtype: float64
```

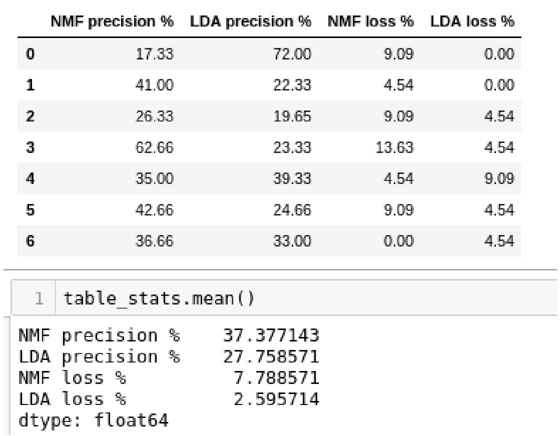
(a) Run n°1

	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	41.33	9.09	4.54
1	41.00	26.33	4.54	0.00
2	26.33	26.66	9.09	4.54
3	62.66	20.66	13.63	0.00
4	35.00	22.00	4.54	0.00
5	42.66	23.33	13.63	0.00
6	36.66	34.00	0.00	9.09

```
table_stats.mean()
```

```
NMF precision %    37.377143
LDA precision %    27.758571
NMF loss %          7.788571
LDA loss %          2.595714
dtype: float64
```

(b) Run n°2

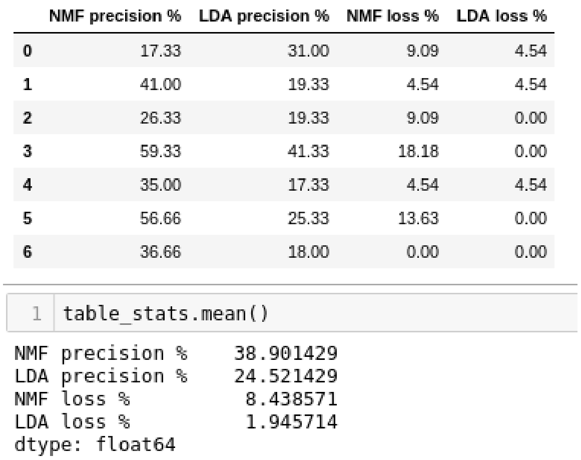


	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	72.00	9.09	0.00
1	41.00	22.33	4.54	0.00
2	26.33	19.65	9.09	4.54
3	62.66	23.33	13.63	4.54
4	35.00	39.33	4.54	9.09
5	42.66	24.66	9.09	4.54
6	36.66	33.00	0.00	4.54

```
table_stats.mean()
```

```
NMF precision %    37.377143
LDA precision %    27.758571
NMF loss %          7.788571
LDA loss %          2.595714
dtype: float64
```

(a) Run n°3

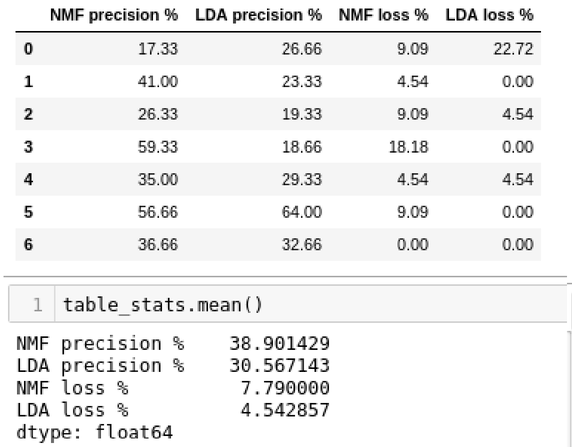


	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	31.00	9.09	4.54
1	41.00	19.33	4.54	4.54
2	26.33	19.33	9.09	0.00
3	59.33	41.33	18.18	0.00
4	35.00	17.33	4.54	4.54
5	56.66	25.33	13.63	0.00
6	36.66	18.00	0.00	0.00

```
table_stats.mean()
```

```
NMF precision %    38.901429
LDA precision %    24.521429
NMF loss %          8.438571
LDA loss %          1.945714
dtype: float64
```

(b) Run n°4

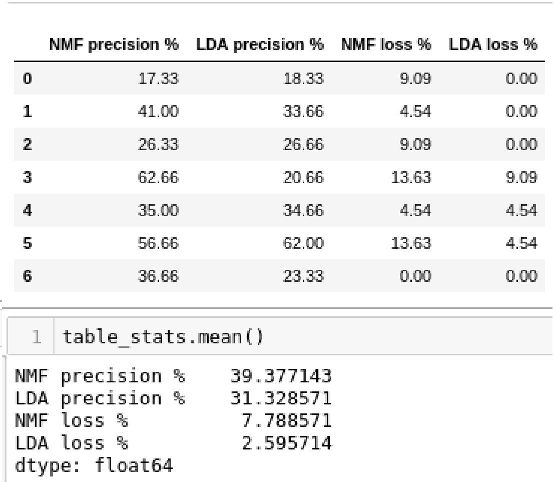


	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	26.66	9.09	22.72
1	41.00	23.33	4.54	0.00
2	26.33	19.33	9.09	4.54
3	59.33	18.66	18.18	0.00
4	35.00	29.33	4.54	4.54
5	56.66	64.00	9.09	0.00
6	36.66	32.66	0.00	0.00

```
table_stats.mean()
```

```
NMF precision %    38.901429
LDA precision %    30.567143
NMF loss %          7.790000
LDA loss %          4.542857
dtype: float64
```

(a) Run n°5



	NMF precision %	LDA precision %	NMF loss %	LDA loss %
0	17.33	18.33	9.09	0.00
1	41.00	33.66	4.54	0.00
2	26.33	26.66	9.09	0.00
3	62.66	20.66	13.63	9.09
4	35.00	34.66	4.54	4.54
5	56.66	62.00	13.63	4.54
6	36.66	23.33	0.00	0.00

```
table_stats.mean()
```

```
NMF precision %    39.377143
LDA precision %    31.328571
NMF loss %          7.788571
LDA loss %          2.595714
dtype: float64
```

(b) Run n°6

and, as a result, the average across all the clusters.
From the above examples it is possible to so that NMF has an average higher accuracy compare to LDA; at the same time, as expected, due to it probabilistic nature, LDA presents an intrinsic instability, represented by a floating value of accuracy going from a max of **31.32** to a minimum of **24.452**;differently NMF proven to have a higher stability maintaining it accuracy around **37-38**.

4.3.7 Searching for optimal number of clusters

One common problem in any topic modelling algorithm is to search for the optimal number of cluster/topic to generate: this factor is strongly linked to the size of data, and their distribution within the data-set.
Usually a large data-set with a sparse data distribution would, most likely, lead to a higher number of cluster to be generated, vice versa for a small one with a low data distribution.
four our case, are considered the data related to a given month, due to the fact that usually within a given month there is a stronger relationship between the topics of queries.

Evaluation method

In the process of seeking for an optimal number of clusters it is needed to rely on a base evaluation method; by considering the factors explained in section 4.3.5, the most suited accuracy evaluation factor chosen is **GLC** factor, using as cluster algorithm *NMF*.

Methodology

The process of looking for an optimal number of cluster is usually performed taking a range of fixed number of clusters and, by evaluating them, the process takes the one with the highest returned accuracy.
By considering a monthly based sub-sets of data, since over a different month a given user may perform a different behavior leading to the generation of a higher or lower size of entries, our pool of sub-set corpora, showed the following size rate: Figure 4.13.
It can be seen that, as expected, the rate size does not follow any particular distribution on a monthly based criteria.
Due to this problem it would have been necessary to performed the following analysis over every single month: this solution was discarded, following the consideration expressed in subsection 4.3.2, it would have lead to a to high time consumption, instead a different approach has been chosen: the **Average Case**.

From Figure 4.13 it is possible to see that an average value may be taken under consideration, expressed with a red line and, with some upper and lower cases, the majority of the distribution resides near this value.

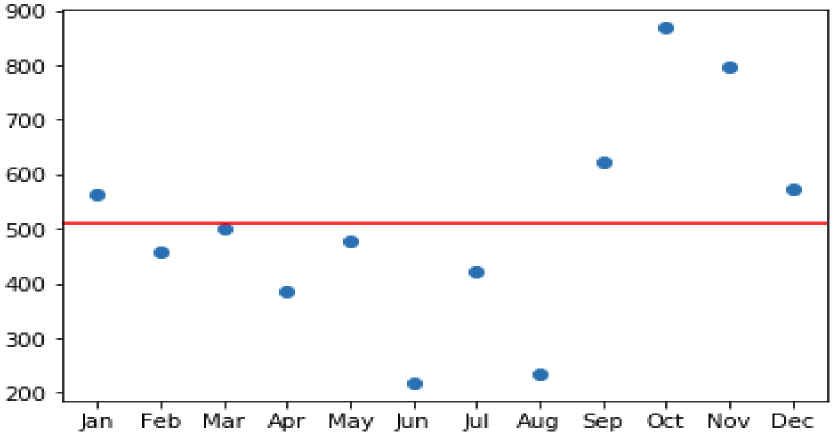
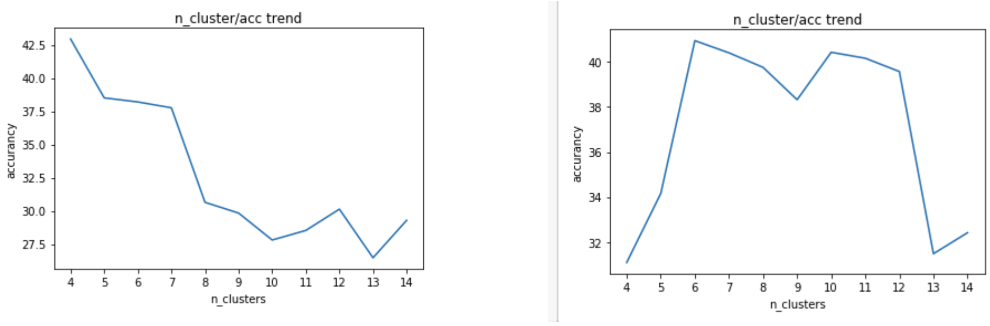


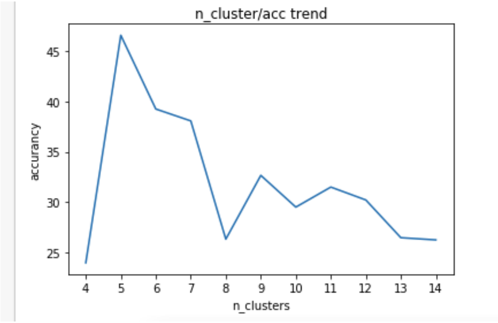
Figure 4.13: Monthly size rate

By considering as a test pool the months that resides closest to this average value, in this case *January,March, May and February*, the above search is performed over this set, leading to the following results.

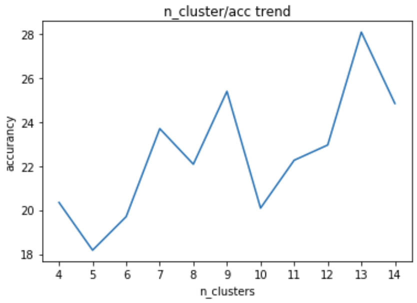


(a) N clusters acc Jan

(b) N clusters acc May



(a) N clusters acc March



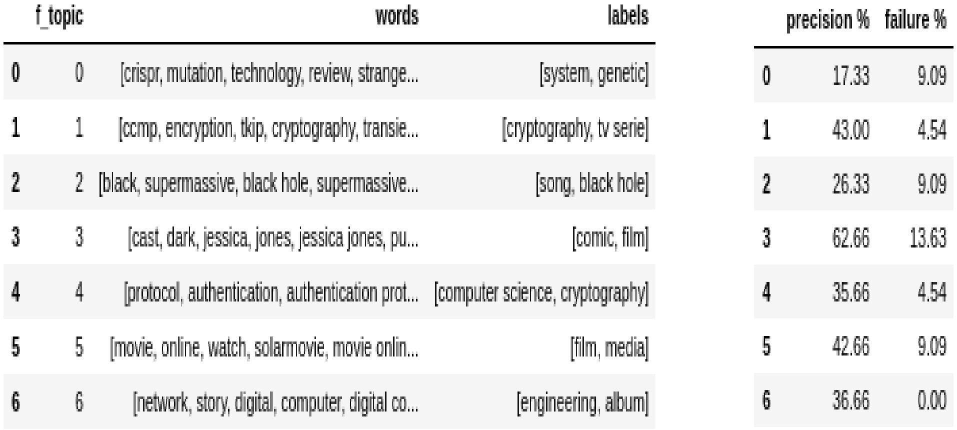
(b) N clusters acc Feb

Even though the four month taken under consideration had a quite close data-set size, they are presenting different behaviour related to the number of clusters likelihood; this is not surprising, in fact this different behaviors are related to the different data distribution within each month.

In order to define the final number of cluster, it can be seen that a good performances are obtained, in all four cases, around the values of **7,8**, then the accuracy start to have a negative slope, so those are the best candidate values as *number of clusters*.

4.3.8 Final Result

After all the consideration expressed, the final result obtained by our clustering algorithm, for a given month, is expressed in Figure 4.16a



	f_topic	words	labels
0	0	[crispr, mutation, technology, review, strange...	[system, genetic]
1	1	[ccmp, encryption, tkip, cryptography, transie...	[cryptography, tv serie]
2	2	[black, supermassive, black hole, supermassive...	[song, black hole]
3	3	[cast, dark, jessica, jones, jessica jones, pu...	[comic, film]
4	4	[protocol, authentication, authentication prot...	[computer science, cryptography]
5	5	[movie, online, watch, solarmovie, movie onlin...	[film, media]
6	6	[network, story, digital, computer, digital co...	[engineering, album]

(a) Final cluster results

	precision %	failure %
0	17.33	9.09
1	43.00	4.54
2	26.33	9.09
3	62.66	13.63
4	35.66	4.54
5	42.66	9.09
6	36.66	0.00

(b) Accuracy and failure

4.3.9 Consideration

Wikipedia proven to be a useful tool as support in the process of Automatic Topic Labeling; unfortunately by the way Wikipedia is structured it has some limitations that may lead to a misplaced label.

In some cases it may happen that Wikipedia returns, in the list of categories for a given word, some that, from a human Judgment point of view, are not good candidate, as nouns or in same cases even names, due to the fact that the algorithm has no way for filtering this categories, it may lead to a misplaced label category, or even to a category with no real meaning.

At the same time it may occur the, if for a given topic, there is no convergence to a given category, in the pool of labels candidate, no label are taken as top candidate, leaving the cluster with no labels.

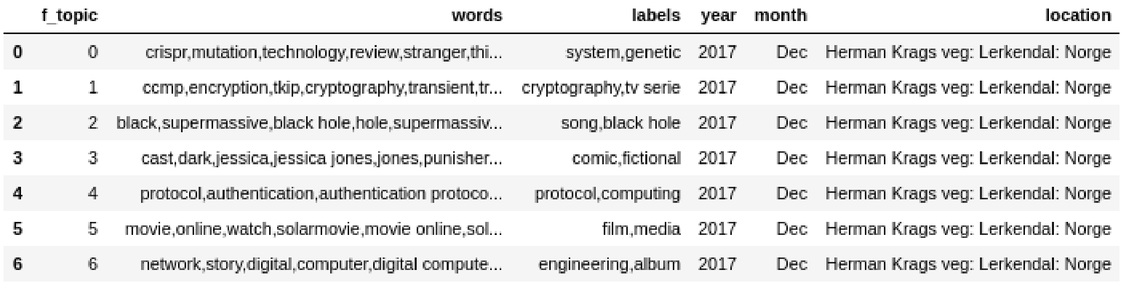
On the other hand, considering all the limitations and constrains, this methodology proven to give a good solution for the given problem.

4.4 Extended Topic Analysis

With a model for performing topic modelling over a pool of data, is now possible to combine this derived information, in order to perform more interesting analysis, with time and location.

Overall Topic Location

By the fact that a single topic is a composition of a series of words, taken from the pool of queries, where a given word can happen in more than one query, while remembering what expressed in Section 4.1, regarding the fact that a single query is tagged with a specific location, it is possible to come at the conclusion that a single word may refer to more than one location, therefore a topic is a mixture of location.
For this reason we aim at generating an **overall location position** for a given topic, by taking all the position linked to a given word, considering the queries in which that word is present and their respective locations, and, by combining them, it is possible to retrieve an overall location.
The final result is presented in Figure 4.17.



	f_topic	words	labels	year	month	location
0	0	crispr,mutation,technology,review,stranger,thi...	system,genetic	2017	Dec	Herman Krags veg: Lerkendal: Norge
1	1	ccmp,encryption,tkip,cryptography,transient,tr...	cryptography,tv serie	2017	Dec	Herman Krags veg: Lerkendal: Norge
2	2	black,supermassive,black hole,hole,supermassiv...	song,black hole	2017	Dec	Herman Krags veg: Lerkendal: Norge
3	3	cast,dark,jessica,jessica jones,jones,punisher...	comic,fictional	2017	Dec	Herman Krags veg: Lerkendal: Norge
4	4	protocol,authentication,authentication protoco...	protocol,computing	2017	Dec	Herman Krags veg: Lerkendal: Norge
5	5	movie,online,watch,solarmovie,movie online,sol...	film,media	2017	Dec	Herman Krags veg: Lerkendal: Norge
6	6	network,story,digital,computer,digital compute...	engineering,album	2017	Dec	Herman Krags veg: Lerkendal: Norge

Figure 4.17: Final Table Information

Yearly-Based and location-based Topic distribution

Taking our topic generated in a monthly based criteria, it is possible to extend the case to a yearly based one, obtaining a more general understanding of the user behavior. The following are same sample of the distribution of the top generated topics, in a yearly based, from our sample data, as shown in Figure 4.18 and Figure 4.19 in the case of the topic PROGRAMMING; while, using the information obtained in the previous section, it is possible to express the distribution of topic in a location based, as shown in Figure 4.20a and Figure 4.20b , for the two most frequent locations.

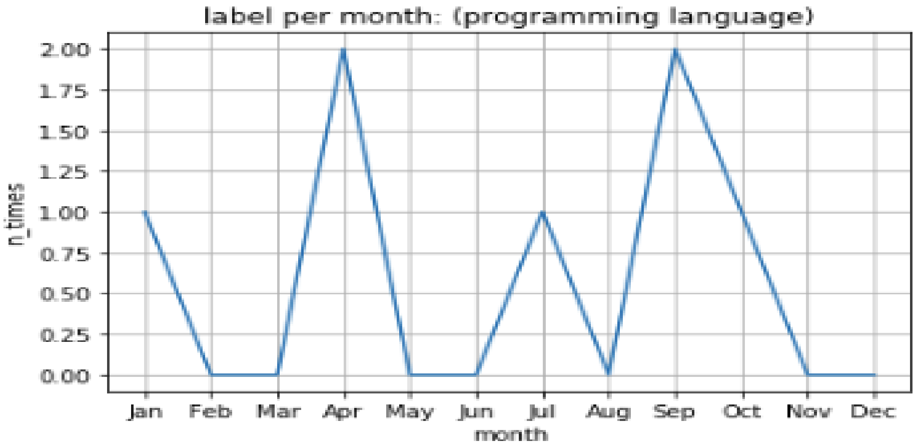


Figure 4.18: Topic distribution over time

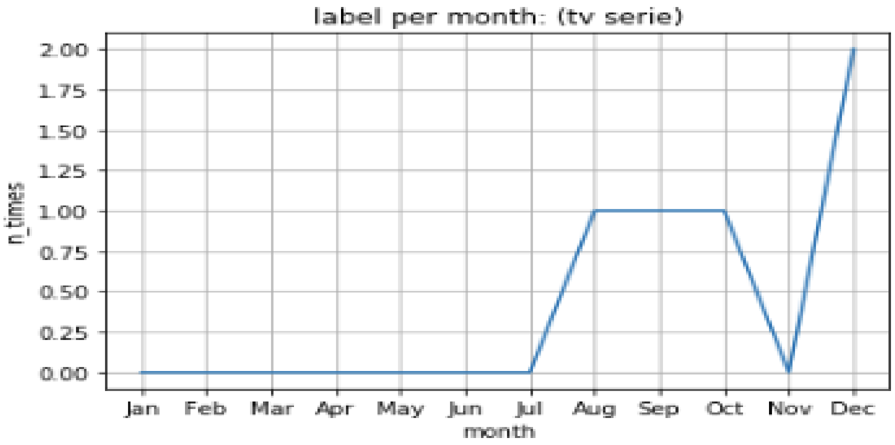
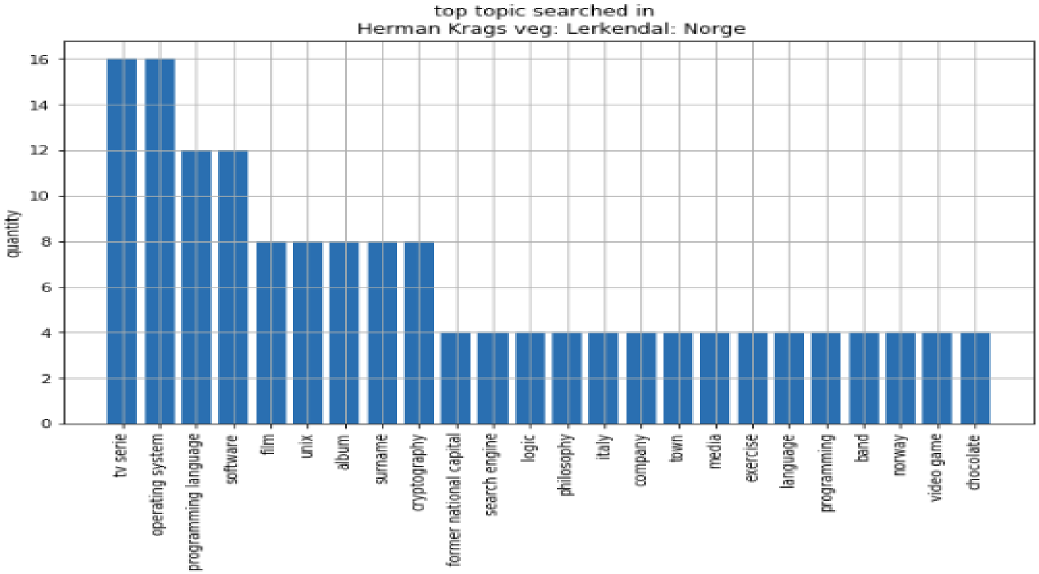
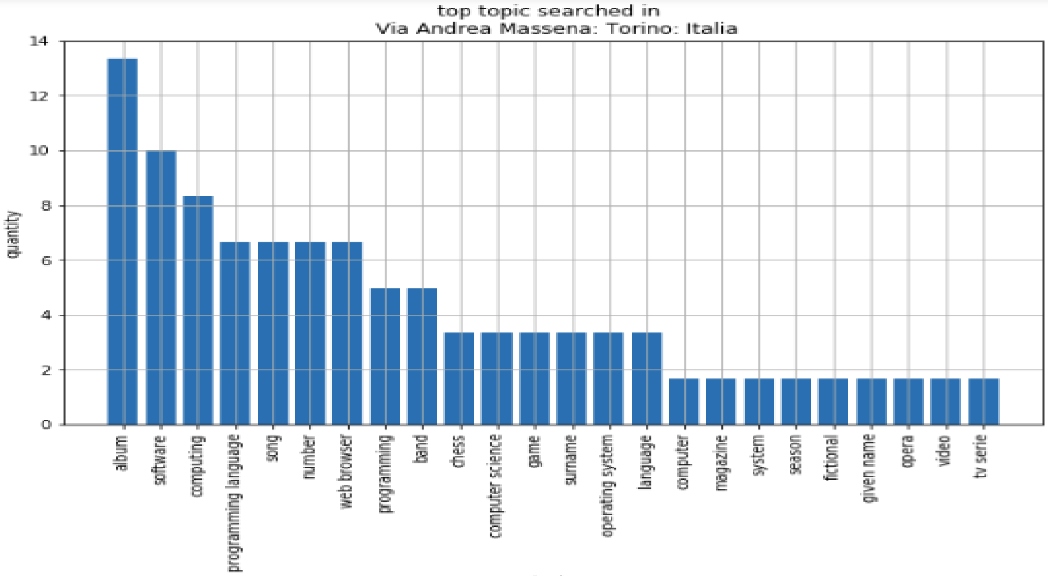


Figure 4.19: Topic distribution over time 2



(a) topic distribution over Location:case 1



(b) Topic distribution over Location:case 2

4.5 Sentiment analysis

The pool of information obtained from Facebook, provides an interesting piece of data, in fact, it is possible to obtain all the messages exchanged by the user over the dedicated messaging application of Facebook: **Messenger**.
As shown in Section 3.3, every entry of the message data-set is composed by a set of useful information as SENDER, MEMBERS OF THE CHAT,TIME-STAMP,AND AN UNIQUE ID ; by considering this input information two type of sentiment analysis will be performed:

1. Sentiment trends over a a monthly based analysis.
2. Sentiment trends over a year based analysis.
3. Sentiment trends over a singular chat.

Sentiment trends over a a monthly based

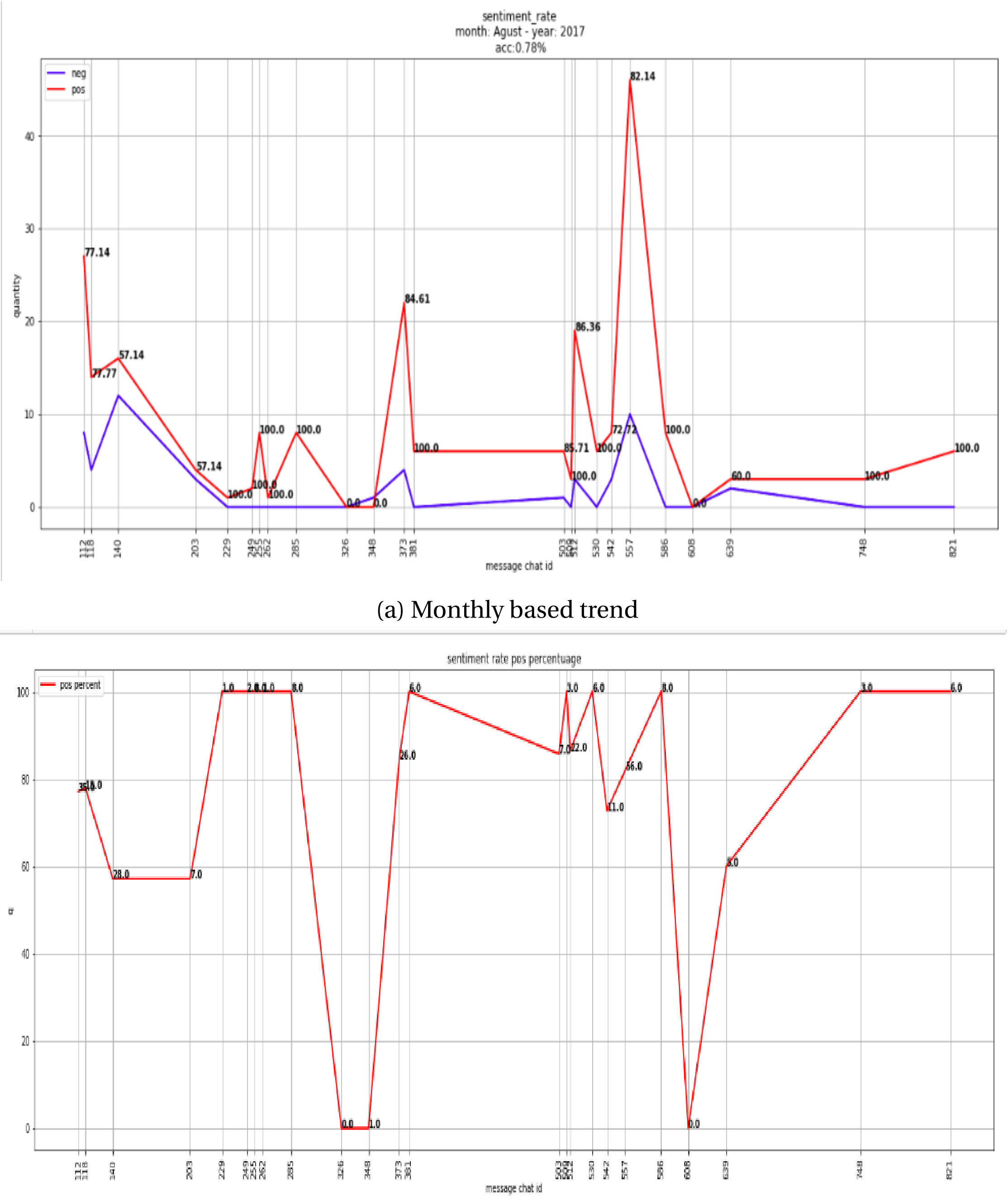
The first analysis was performed considering a MONTHLY-BASED criteria, following the consideration expressed in in Section 4.2.1, in order to have, between this two set of analysis a symmetry for future combined analysis; this results in taking as a sample data the messages, from all the chats, only the messages that were related to that given month: as a result we generate, for every chat that has at least one message sent on that given month, a distribution of positive and negative factors, as shown in Figure 4.21a, were for each chat expressed with its *chat Id*, we have a correspondence of Positive and Negative values.

Sentiment trends over a year based analysis

Keep following the criteria used for the Topic Modelling representation, this analysis was performed taking a higher level of abstraction, and considering a a YEAR-BASED criteria; this follows the same aspect expressed for the previous analysis with the only difference that, instead of considering a single chat as a reference value, it considers the monthly distribution; in fact Figure 4.22a, present in the abscissa axe, the month taken as reference.

Sentiment trends over a singular char

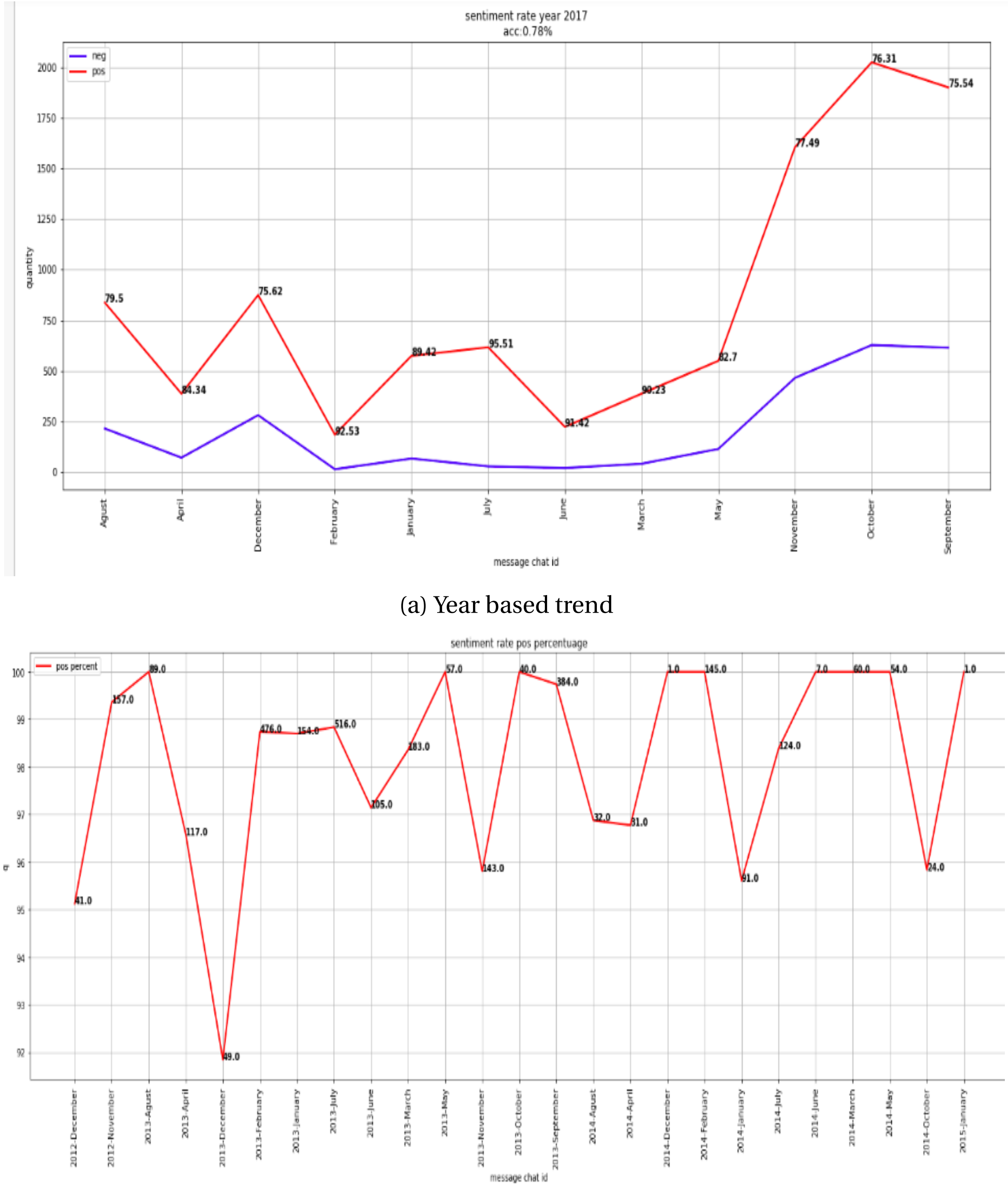
The final analysis is performed using a different type of criteria, taking a lower level of abstraction and considering as reference a single chat, performing the model over all the messages inside the messages belonging to that given chat, as shown in Figure 4.23a
Due to the fact that, by considering different chats, characterized by a different amount of messages, we present, for every of the above samples, a percentage representation for a better understanding, respectively in Figure 4.21b, Figure 4.22b,Figure 4.23b.



(a) Monthly based trend

(b) Monthly based percentage

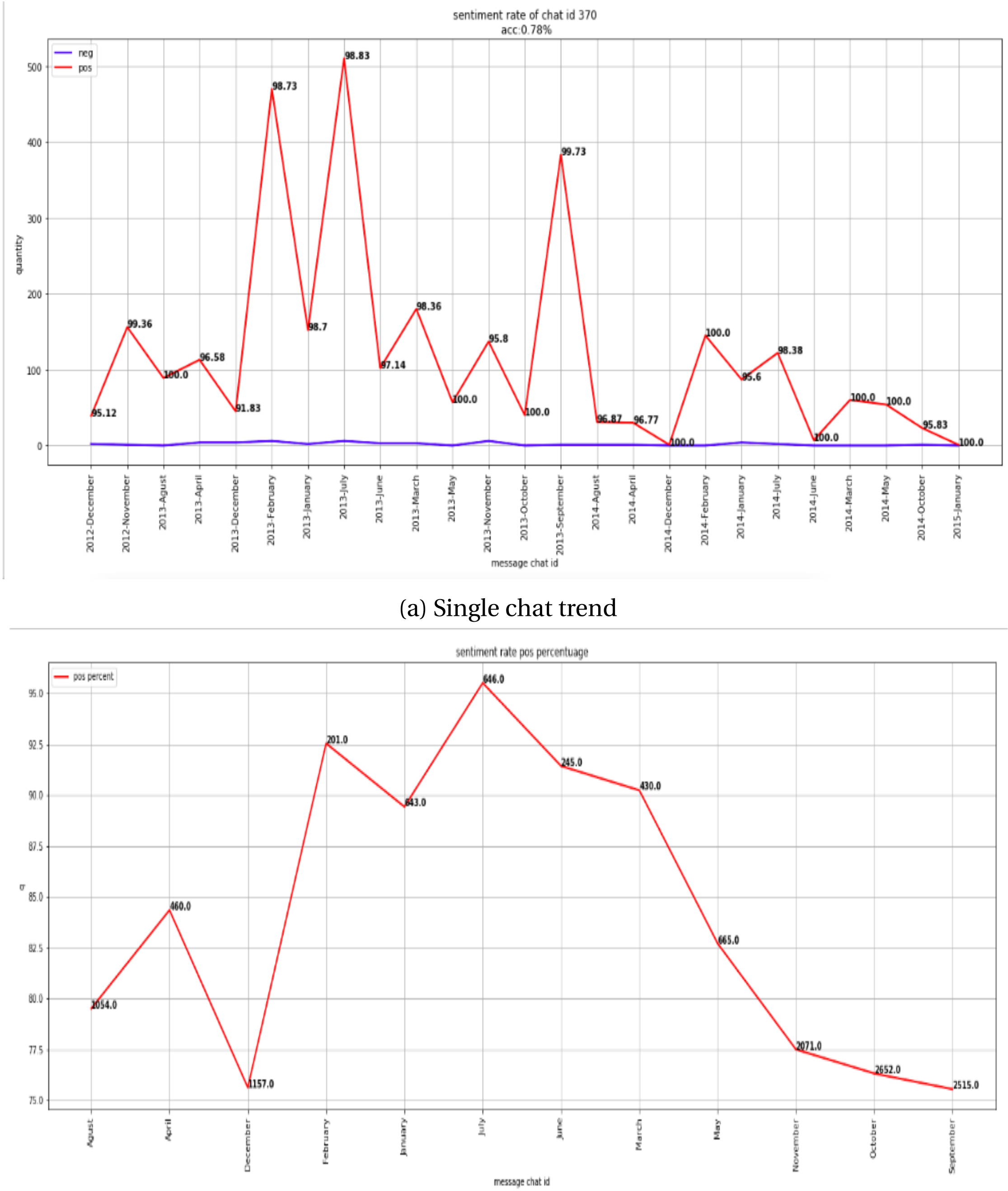
Figure 4.21: Analysis one



(a) Year based trend

(b) Year based percentage

Figure 4.22: Analysis two



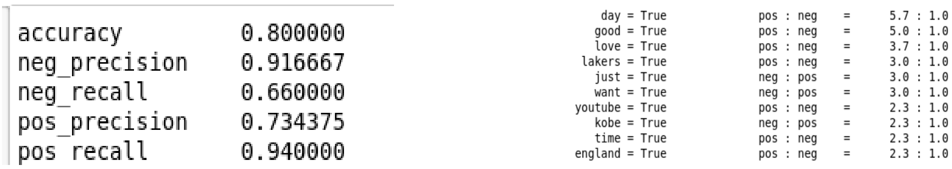
(a) Single chat trend

(b) Single chat percentage

Figure 4.23: Analysis three

4.5.1 Classifier Performances

As it is shown in Figure 4.24a, our classifier has an overall good performances, reaching the 80 % of accuracy; the only factor that we weren't able to improve was the NEGATIVE RECALL, that is around 60 %, this means that the algorithm tends to produce, in some cases, false positive results.
Figure 4.24b shows a sample of the most informative words from the training corpora.



```
accuracy          0.800000
neg_precision     0.916667
neg_recall        0.660000
pos_precision     0.734375
pos recall        0.940000
```

(a) Sentiment analysis stats

```
     day = True              pos : neg    =      5.7 : 1.0
    good = True              pos : neg    =      5.0 : 1.0
    love = True              pos : neg    =      3.7 : 1.0
  lakers = True              pos : neg    =      3.0 : 1.0
    just = True              neg : pos    =      3.0 : 1.0
    want = True              neg : pos    =      3.0 : 1.0
 youtube = True              pos : neg    =      2.3 : 1.0
    kobe = True              neg : pos    =      2.3 : 1.0
    time = True              pos : neg    =      2.3 : 1.0
 england = True              pos : neg    =      2.3 : 1.0
```

(b) Most Informative Words Sample

Figure 4.24: Sentiment analyzer Information

Chapter 5

Conclusions and future work

5.1 Conclusions

This project started as an enterprise search with no mayor ideas on what type of data, or analysis to expect; linking this with the constraints that appeared at the beginning of it, it shown to be a quite challenging case study.
We presented a method able to extract some insights out of a selected pool of information defining, event thought in an initial state, a user online behavior.
Moreover our method for automated cluster and labeling demonstrated, according to the type of input information, good performances evaluated using a new ad-hoc evaluation metric and a Human Based Criteria. This method was designed over the pool of queries obtained by Google, but it proved to perform good in a general context, making it a good tool for topic extraction out of any text-based data, caring any resilient topic information.
We conclude this work by presenting some ideas that came out over the course this project, for future work, that could bring more interesting insights out of the presented problem.

5.2 Future Works

Extending the input Data

As expressed in Chapter 1, due to the initial complexity of this problem, we had the need of taking in consideration a sub-problem, using as input reference only two Service Providers. The new step into this approach will be extending the input reference data to a wider range of Service Providers, including other relevant ones as Instagram, Twitter, LinkedIn ecc.
The interest behind this will be performing not only the above expressed analysis, in order to improve their accuracy and relevance but, concurrently, to see the feasibility of obtaining more type of relevant insights.

Improving Performances

As expressed in Section 4.3.2, the topic modelling part of this project, has a high time consuming rate. Due to the purposes of this project, no mayor interest was putted over the performances evaluation, leaving it for future improvement.

Extended analysis

Having a pool of information related to the user behavior, in a fixed and well formatted structure, could make possible a next step in this search, using them as input data for more derived insights search. According to the GDPR, it would be interesting to extract some privacy concern, or clustering of those information in a Privacy Based context, and use them for predicting the Privacy Level of future information.
Those were just some examples, but the range of applications could be even wider, leading to other interesting analysis that we did not take under considerations.

Time Usage Prediction

Another interesting side analysis that we came out with, but did not have the time to explore, was the usage of the time information in order to obtain some future user usage prediction. We thought to take advantages, for this case, of algorithms as **LSTM**, that stands for Long Short time memory, a Machine learning algorithm based on a recursive Neural Network, designed for time, or forecast prediction [32].

Fuzzy Logic and Data Labeling

As a final result of our work we have a set of entries each of them characterized by some information as, location, time and topic of interest. This entry shown a high heterogeneous properties, making not easy to perform some extended analysis over them, as Privacy Concern labeling, with a gradual range of classification.
For this reason we thought that an interesting way to extract some more information will be by using the **Fuzzy Logic** approach [33].

User Interface application

Finally, by extending this work to more Service Providers, with more insights, this would lead to the necessity of designing of a *User Application Interface*, for a better user experience and control above all the above mentioned information and insight.

Bibliography

[1] *Search Engine Market Share*. https://netmarketshare.com/.

[2] *Most popular mobile social networking apps in the United States as of February 2018, by monthly users (in millions)*. https://www.statista.com/statistics/248074/most-popular-us-social-networking-apps-ranked-by-audience/.

[3] Manisha Valera Riya Shah. Survey of sensitive information detection techniques: The need and usefulness of machine learning techniques. 2017.

[4] *Lemmatization*. https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[5] *WordNet*. "https://wordnet.princeton.edu/.

[6] *NMF wikipedia*. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization.

[7] Yihong Gong Wei Xu, Xin Liu. Document clustering based on non-negative matrix factorization. 2003.

[8] Michael I. Jordan David M. Blei, Andrew Y. Ng. Latent dirichlet allocation. 2003.

[9] Philip Kegelmeyer Keith Stevens. Exploring topic coherence over many models and many topics. 2012.

[10] *Sentiment analysis*. https://en.wikipedia.org/wiki/Sentiment_analysis.

[11] Patrick Paroubek Alexander Pak. Twitter as a corpus for sentiment analysis and opinion mining. 2010.

[12] *Python Twitter Tools*. https://github.com/sixohsix/twitter.

[13] *Python Twitter API*. https://github.com/geduldig/TwitterAPI.

[14] *Naive Beyesian Wikipedia*. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[15] Sanjay Chakraborty Lopamudra Dey. Sentiment analysis of review datasets using naïve bayes' and k-nn classifier. 2016.

[16] *Download your data*. https://support.google.com/accounts/answer/3024190?hl=en.

[17] *Download your data*. https://www.facebook.com/help/1701730696756992?helpref=hc_global_nav.

[18] *Csv wikipedia*. https://en.wikipedia.org/wiki/Comma-separated_values.

[19] *Natural Language Processing*. https://en.wikipedia.org/wiki/Natural_language_processing.

[20] *Python 2.7.0 Documentation*. https://docs.python.org/2/index.html.

[21] *Jupyter doc*. http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html/.

[22] *numpy documentation*. https://docs.scipy.org/doc/.

[23] *Pandas documentation*. http://pandas.pydata.org/pandas-docs/version/0.23/.

[24] *sklearn documentation*. http://scikit-learn.org/stable/documentation.html.

[25] *Natural language toolkit*. https://www.nltk.org/.

[26] *Beautifulsoup*. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[27] *Date Time Documentation*. "https://docs.python.org/2/library/datetime.html.

[28] *Geopy documentation*. http://geopy.readthedocs.io/en/stable/.

[29] Mengdi Zhang Linmei Hu, Xuzhong Wang. Learning topic hierarchies for wikipedia categories. 2015.

[30] Moonyoung Kang Tae Yano. Taking advantage of wikipedia in natural language processing. 2008.

[31] *PyLDAvis documentation*. `http://pyldavis.readthedocs.io/en/latest/`.

[32] Lovekesh Vi Pankaj Malhotra1. Long short term memory networks for anomaly detection in time series. 1999.

[33] Mirko Vujoševi Miroslav Hudec. Selection and classification of statistical data using fuzzy logic. 2010.