UNIVERSITAT POLITECNICA DE CATALUNYA (UPC) - BARCELONATECH

Sentiment Analysis on Twitter

by

Rocco Proscia

Supervised by
Jose Luis Balcazar , Marta Arias (LARCA Research Group)

Company: ServiZurich

Master in Innovation and Research in Informatics
Specialization in Data Mining and Business Intelligence
Facultat d'Informatica de Barcelona (FIB)

February 2016

UNIVERSITAT POLITECNICA DE CATALUNYA (UPC) - BARCELONATECH

Abstract

Master in Innovation and Research in Informatics
Specialization in Data Mining and Business Intelligence
Facultat d'Informatica de Barcelona (FIB)

In recent years more and more people have been connecting with Social Networks. One of the most used is Twitter. This huge amount of information is attracting the interest of companies. One reason is that this huge source of information can be used to detect public opinion about their brands and thus improve their business values.

In order to transform the information present in the Social Networks into knowledge several steps are required. This project aim to describe them and provide tools that are able to perform this task.

The first problem is how to retrieve the data. Several ways are available, each one with its own pros and cons. After that it is necessary to study and define proper queries in order to retrieve the information needed.

Once the data is retrieved you may need to filter and explore your data. For this task a Topic Model Algorithm (LDA) has been studied and analyzed. LDA has shown positive results when it is tuned in the proper way and it is combined with appropriate visualization techniques. The difference between a Topic Model Algorithm and other Clustering/Segmentation techniques is that Topic Models allows each "document" (instance) to belong to more than one topic (cluster).

LDA doesn't natively work well on Twitter due to the very short length of the tweets. An investigation in the literature has revealed a solution to this problem. Another problem that is common in clustering is how to validate the Algorithm and how to choose the proper number of topics (clusters), for this problem several metrics in the literature have been explored.

Afterwards, Sentiment Analysis techniques can be applied in order to measure the opinion of the users . The literature presents several approaches and ways to solving this problem. This work is focused in solving the Polarity Detection task, with three classes , so, classify if a tweet express a positive , a negative or a neutral sentiment. Here reach accurate results can be challenging, due to the messy nature of the twitter posts. Several approaches have been tested and compared. The baseline method tested is the use of sentiment dictionaries, after that , since the real sentiment of the twitter posts is not available, a sample has been manually labeled and several Supervised approaches combined with various Feature Selection/Transformation techniques have been tested.

Finally, a totally new experimental approach, inspired from the Soft Labeling technique present in the literature, has been defined and tested. This method try to avoid the costly task to manually label a sample in order to validate a model. In the literature this problem is solved for the two-class problem, so by considering only positive and negative tweets. This work try to extend the soft-labeling approach to the three class problem.

Acknowledgements

Alla fine questo Master é giunto al termine. Due anni passati in un batter d'occhio in cui ho imparato tantissimo e vissuto le piú assurde esperienze, ho avuto la possibilitá di frequentare un Universitá fantastica, piena di studenti e professori validi e motivati, in una cittá folle e meravigliosa come Barcellona.

Vorrei ringraziare inanzitutto i miei genitori per aver permesso (finanziato) questa esperienza e i miei due supervisori di tesi Prof. Jose Balcazar e Marta Arias (ancora auguri per la piccola!) per avermi sopportato in questi ultimi mesi.

Grazie anche a tutta la gente che , anche solo di passaggio, ho avuto modo di incontrare e passare dei bei momenti, un ringraziamento e saluto , in particolare , anche a chi é lontano, ma resta ancora vicino.

E poi basta fare gli sdolcinati che é giá assai, andiamo a presentare stó progetto vá!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This work is organized as follows. Chapter 2 outlines a theoretical baseline necessary for understanding the following chapters. It describe how to retrieve and store data from Twitter or other Social Networks (the approach can also be applied to other domains). After retrieving the information you may need to clean and explore it. For this reason I describe some fundamental Natural Language Processing concepts and Topic Models algorithms. After that, for performing Sentiment Analysis you may need to apply Supervised Algorithms. A brief description of that is proposed with one example: The Naive Bayes Classifier. Before applying any Machine Learning Methods is important to follow the Natural Language pre-processing steps already described, but also apply Feature Selection Techniques, they are briefly described with a couple of examples: the χ^2 test and Information Gain.

Chapter 3 represents the theoretical research for this project. The state of the art solutions to the two main problems of the project are described: Topic Model Algorithm and Sentiment Analysis. Both are specific to the Microblogs Environment since our work is based on Twitter and in Twitter the posts are expressed in 140 characters or less. A deep search in the literature has been performed in order to describe the actual state of the art of this area. The length is not the only problem, in fact a tweet is usually messy, with a lot of slang expressions and misspelled words. All these aspects can make life difficult to a Data Scientist and make works terribly techniques that are known to perform well on more "clean" and "long" documents. For this reason this Chapter focus on the State of the Art techniques that aims to deal with this domain.

Then Chapter 4 and 5 are dedicated to the experimental part. The first experiment regards Information Retrieval. For this Topic Model Algorithms have shown good results when they are tuned in the proper way, validated with the proper metrics, and visualized with appropriate techniques.

The second experiment deals with Sentiment Analysis, in particular it focuses on the polarity detection task. Here Several approaches have been compared. The most simple just make use of sentiment dictionaries, it represents the baseline for more sophisticated methods. After that a training set is manually labeled and several approaches are performed on it. Finally a new approach has been tested. It is inspired from the techniques of Soft-Labeling, that has shown positive results in several examples in the literature. This experiment try to extend this approach to the three class problem.

Finally on the Chapter 6 there are some visualization and insights obtained from the analysis , plus several ideas for future works.

Below is shown an overview of the macro-components of the project.

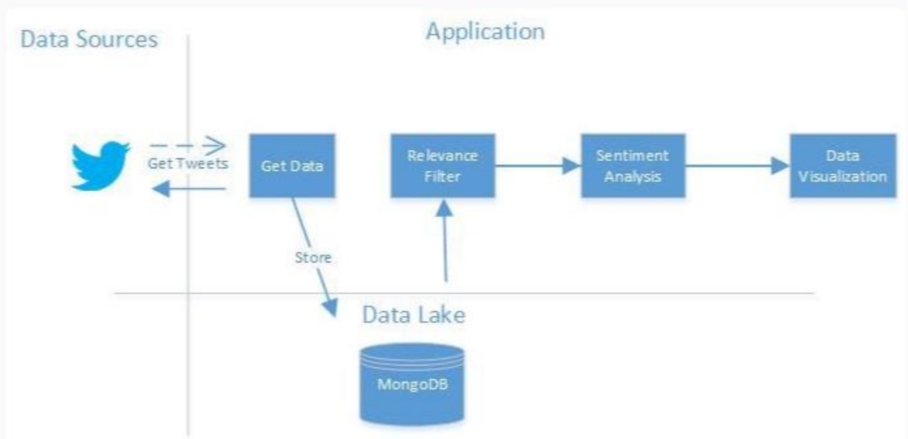


Figure 1.1: Project overview

Chapter 2

Background Theory

2.1 Retrieving Data from the Web

The Web is a huge repository of data. It is estimated that the 90% of the worlds data has been generated in the past 2 years. [1] This is a huge opportunity for researchers. Data can be obtained without the cost of performing questionnaires, surveys , interviews or other traditional ways of collecting data .

But the data on the Web is not ready to be analyzed. It is important to know how to extract and clean it. Furthermore not all the data sources can be used without limits. Some companies are not happy to share their data with everyone. In other cases your data may contain sensitive information (for example in the medical domain). So in order to respect privacy it is important to anonymize [1] the data before performing any kind of analysis.

So it is important to know which data you are allowed to extract and what you are allowed to do with them.

You can extract data in several ways, they can be placed into three categories:

- Get the data trough an API .
- Use a Web Scraper to crawl the Web.
- Buy the data from a reseller.

The first two approaches are described in the following sections. Nevertheless it is not always possible to retrieve the data trough these techniques. Sometimes the data owners

[1] https://www.sciencedaily.com/releases/2013/05/130522085217.htm

can be very conservative and the only way to retrieve your data would be to buy them through an official reseller.

2.1.1 API

Although various APIs exist for a variety of different software applications, in recent times API has been commonly understood as meaning web application API. Typically, a programmer will make a request to an API via HTTP for some type of data, and the API will return this data in the form of XML or JSON. Although most APIs still support XML, JSON is quickly becoming the encoding protocol of choice.

Sometimes the amount of information that you can retrieve with API is limited. Twitter for example limits the number of queries you can perform in a window of 15 minutes [2]. Not only the bandwidth is limited but also the amount of information that you can retrieve. The API.search for example, allows to search tweet by a keyword. The problem is that the results are limited to the last 7 days. [3]

2.1.2 Web Scraping

In theory, web scraping is the practice of gathering data through any means other than a program interacting with an API (or, obviously, through a human using a web browser). This is most commonly accomplished by writing an automated program that queries a web server, requests data (usually in the form of the HTML and other files that comprise web pages), and then parses that data to extract needed information. In practice, web scraping encompasses a wide variety of programming techniques and technologies, such as data analysis and information security . These books are very exhaustive guides on the topic [2] [3] .

There are several real-world use cases in the market where Web Scraping is used right now. For example e-commerce sites use it to identify best-selling products , job-search sites scrape job listings from several sources, as well as flight-comparison websites search the best flight option through a huge number of airlines and the list can continue.

2.1.3 Discussion: Legitimacy of Web Scraping

Web Scraping is a very powerful technique in order to obtain information at a low cost . However it is important to use it in a conscious way. Most of the times Web Scraping is

[2] https://dev.twitter.com/rest/public/rate-limiting

[3] https://dev.twitter.com/rest/public/search

perfectly legal, but there are some cases where it is not. Big companies use web scrapers for their own gain but don't want others to use bots against them. It is difficult to define precisely what is allowed and what is not because there are several factors to consider. From the law of a specific state , to the Terms of Service (ToS) of a specific Web-Service, if the ToS can be applied to your scraper or not, the type of business you want to perform with the data extracted and so on.

Several article on the web have dealed about the legality of web-scraping [4] , [5] , [6] , [7] , [8] , [9]. Below some interesting lawsuits are showed.

- **Facebook v. Power.com - 2009**

  Power.com tried to aggregate various social networking accounts in a single place, so you could manage them all at once through a single interface. Yet Facebook charged the company with all sorts of complaints, including copyright and trademark infringement, unlawful competition and violation of the computer fraud and abuse act. Power.com asked for the case to be dismissed, but at the end the judge sided with Facebook, but did so in a troubling way, by basically suggesting that since Facebook's terms of service prohibited these uses, it made it copyright infringement.

  The court found that even though the data being used by Power.com isn't owned by Facebook (it's the users') the scraping was still copyright infringement, because in order to scrape the non-infringing content, Power.com had to first "scrape" the whole page .

  This lawsuit has sparked a lot of discussions on the Web. First of all: just because the terms of service said you can't do any automated scraping of the site, the scrape becomes illicit? Also, they have stated the scrape as copyright infringement just because the scraper had to first read through copyrighted content to get to the non-infringing stuff. But, that seems to go against the entire purpose of copyright law. The fact that the scraper reads copyrighted content shouldn't mean that it's infringement. It's not doing anything with that content other than using it to find the content it can make use of.

- **QVC v. Resultly - 2014**

[4] http://www.integrity-research.com/mitigating-risks-associated-with-web-crawling/

[5] http://www.bna.com/legal-issues-raised-by-the-use-of-web-crawling-and-scraping-tools-for-analytics-purposes

[6] http://blog.icreon.us/advise/web-scraping-legality

[7] https://www.techdirt.com/articles/20090605/2228205147.shtml

[8] http://www.forbes.com/sites/ericgoldman/2015/03/24/qvc-cant-stop-web-scraping/#7f2a198c4403

[9] http://www.law360.com/articles/389930/collegesource-s-ip-contract-suit-against-rival-tossed

QVC is a well-known TV retailer. Resultly is a start-up shopping app self-described as "Your stylist, personal shopper and inspiration board" Resultly builds a catalog of items for sale by scraping many online retailers, including QVC. Scraping of retailers websites isn't unusual; as the court say, "QVC allows many of Resultlys competitors, e.g., Google, Pinterest, The Find, and Wanelo, to crawl its website." Resultly cashes in when users click on affiliate links to QVC products .

In May 2014, Resultly's automated scraper overloaded QVC's servers, causing outages that allegedly cost QVC $2M in revenue. QVC eventually blocked access to Resultly's scraper. Subsequent discussions were irresolute, and QVC sought a preliminary injunction based on the Computer Fraud & Abuse Act . The court concludes that QVC hasn't shown a likelihood of success because Resultly lacked the required intent to damage QVCs system

The outcome of this lawsuit is completely different from the previous one. In this case, even a massive activity of Scraping that has caused the cessation of a service (not that far from a Denial of Service attack) has been considered completely legal.

- **Collegesource v. AcademyOne - 2015**

  CollegeSource and AcademyOne are competitors in the market that helps prospective students with the college transfer process. CollegeSource maintain its principal place of business in California, while AcademyOne maintained its principal place of business in Pennsylvania. However, both companies seek to serve the transfer market online not bound by state or region. Important to the appeal, AcademyOne targeted prospective transfer students by state through use of Google AdWords, solicited California colleges and state educational agencies through phone and email, and sponsored the keynote speaker at a conference held for the benefit of higher education executive officers meeting in San Diego.

  CollegeSource claimed to own and copyright a digital collection of 44,000 course catalogs from 3,000 colleges, worth allegedly $10 million. The complete digital collection was available through subscription as .pdf files on CollegeSource's websites. Known to CollegeSource, many of the .pdf files were also individually distributed across thousands of institutional websites. AcademyOne, a few months after its founding, made several attempts to inquire about CollegeSource's collection of course catalogs as it researched how to compile a nationwide database of college and university level courses to support its college transfer websites. At least three employees registered for trial membership with CollegeSource that allowed them to download three sample catalogs each. CollegeSource declined AcademyOne's early attempts to partner to keep its competitive advantage in the market place.

Therefore, AcademyOne decided to collect and build its own collection of college and university catalogs to harvest the course information. AcademyOne hired a China-based contractor to collect the catalogs and mine the course descriptions from the files or web pages. The contractor collected over 18,000 .pdf files and thousands of html pages containing course descriptions from a list of schools websites that AcademyOne had provided. During this process, the contractor collected roughly 680 .pdf files that contained CollegeSources splash page and copyright page. CollegeSource also claimed some courses descriptions displayed on AcademyOne's websites were mined and traceable to CollegeSource's electronic catalog versions because they supposedly contained typographical errors and "seeds" introduced by the digitization and conversion effort from years prior. Moreover, some of the course catalog pdf files included a page terms prohibiting redistribution, modification, or commercial use of the catalogs (without consent of CollegeSource) on the second page of the pdf.

The federal judge dismissed claims against AcademyOne by rival CollegeSource over republishing course catalogs and course information digitized and maintained by the latter company, finding that the usage violated neither trademarks nor contracts governing AcademyOne's subscription service. The judge emphasized that AcademyOnes efforts to collect the information did not run afoul of any contracts established between the two.

- **Final Considerations**

  These lawsuits show that the threshold between allowed and not allowed is very tiny and fragile. The Facebook case shows that scraping by using a user credential is not allowed if the term of service of the platform forbids it.

  The second case shows that as long as you allow Web Scraping on your Website, even a heavy scrape that could potentially cause the interruption of your service is not considered as harmful (as long as it is non intentional).

  Finally the third case show that even a heavy scrape activity that aim to obtain a huge amount of information is not considered illegal as long as there is no agreement between the two counterparts that explicitly forbids it.

  If you are interested in going deeper in the argument, this infographic show a wide historical picture of past Court Cases. This section should not be taken as legal council. If you are interested in doing business that involves Web Scraping you should seek for legal advice in order to be sure what you are allowed to do.

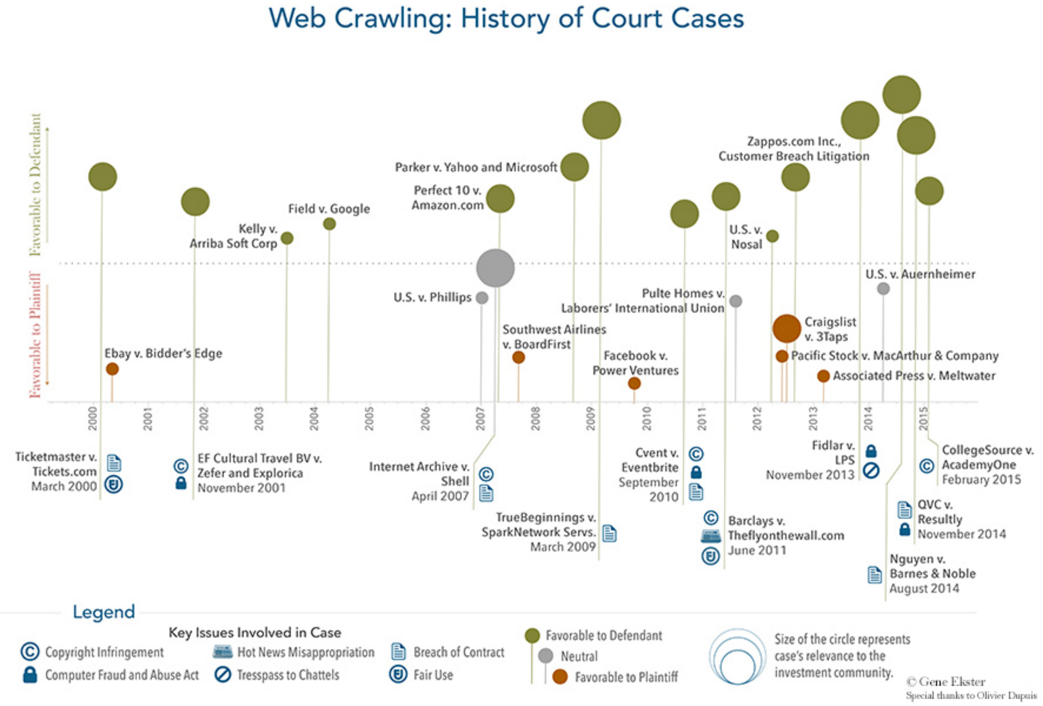


Figure 2.1: Web Crawling - History of Court Cases , source: http://www.integrity-research.com/

2.2 Storing The Data

In the Information Retrieval phase, the data needs to be stored somewhere in order to be analyzed later. In this project we have choosen a NoSQL Database for this task. The reason is that we were not aware of the size of the data that we were going to analyze. Thus NoSQL engines provides more scalability in case the size of the data would be very big.

2.2.1 MongoDB

MongoDB is One of the most popular document stores. It is a document oriented database. All data in mongodb is treated in JSON/BSON format. It is a schema less database which goes over tera bytes of data in database. It also supports master slave replication methods for making multiple copies of data over servers making the integration of data in certain types of applications easier and faster. MongoDB combines the best of relational databases with the innovations of NoSQL technologies, enabling engineers to build modern applications. MongoDB maintains the most valuable features of relational databases: strong consistency, expressive query language and secondary indexes. As a result, developers can build highly functional applications faster than NoSQL databases. MongoDB provides the data model flexibility, elastic scalability and high performance of NoSQL databases. As a result, engineers can continuously enhance applications, and deliver them at almost unlimited scale on commodity hardware. [10]

2.3 Information Filtering

Once the data is retrieved, it is a good idea to start exploring the data, in order to check the quality of your data and eventually filter the one that is non relevant for the analysis. If the data is numerical or categorical, traditional techniques from the Multivariate Analysis are suitable for the task. However these techniques could be unsuitable for contents expressed in natural language. Topic Model algorithms can help in this case.

2.3.1 Topic Modeling

Topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents. Topic models can organize the collection according to the discovered themes. Topic modeling algorithms can be applied to massive collections of documents.Recent advances in this field allow us to analyze streaming collections, like you might find from a Web API. Topic modeling algorithms can be adapted to many kinds of data. Among other applications, they have been used to find patterns in genetic data, images,and social networks. [4]

Some general but useful suggestions from [5]:

- **Work preferably on a large Corpus.**

  Topic modeling is built for large collections of texts. In general is recommended to have at least 1,000 items in the collection to model. The question of ”how big” or ”how small” is ultimately subjective.

- **Familiarity with the corpus.**

  This may seem counter intuitive when is planned to use topic modeling to help find out more about a large corpus, and yet it is very important to have at least an idea of what should be there. Topic modeling is not an exact science by any means. The only way to know if the results are useful or wildly off the mark is to have a general idea of what should be there. Most people would probably spot the outlier in a topic of ”tobacco, farm, crops, navy” but more complex topics might be less obvious.

[10] https://www.linkedin.com/pulse/real-comparison-nosql-databases-hbase-cassandra-mongodb-sahu

- **A way to understand the results.**

  Topic modeling output is not entirely human readable. One way to understand what the program is telling you is through a visualization, but is important to know how to understand the visualization. Topic modeling tools are fallible, and if the algorithm isn't right, they can return some bizarre results.

2.3.1.1 LDA

Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora. The intuitive idea behind it is the following [6] :

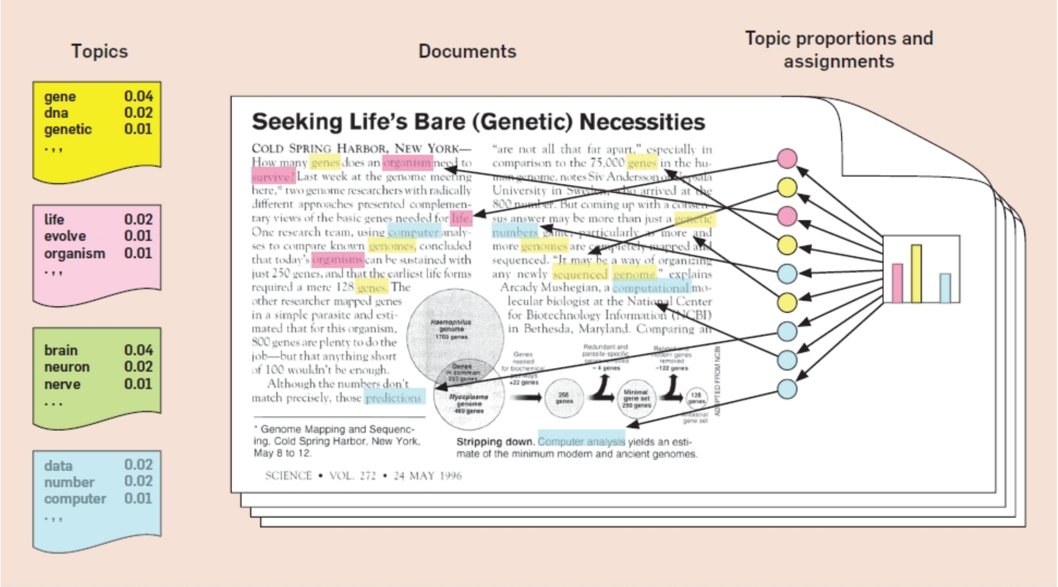


FIGURE 2.2: The intuition behind LDA , source : www.cs.princeton.edu/~blei/kdd-tutorial.pdf

In the picture are described the three fundamental components of LDA:

- Each **Topic** is a distribution over words.
- Each **Document** is a mixture of corpus-wide topics.
- Each **Word** is drawn from one of those topics.

In LDA, the observed data are the words of each document and the hidden variables represent the latent topical structure, i.e., the topics themselves and how each document exhibits them. Given a collection, the posterior distribution of the hidden variables given the observed documents determines a hidden topical decomposition of the collection. Applications of topic modeling use posterior estimates of these hidden variables to perform tasks such as information retrieval and document browsing.

The interaction between the observed documents and hidden topic structure is manifest in the probabilistic generative process associated with LDA, the imaginary random process that is assumed to have produced the observed data. Let K be a specified number of topics, V the size of the vocabulary, D the number of documents, $\vec{\alpha}$ a positive K-vector and η a scalar. Let $Dir_k(\eta)$ denote a K dimensional symmetric Dirichlet with scalar parameter η .

1. For each topic,
   (a) Draw a distribution over words $\vec{\beta}_k \sim Dir_v(\eta)$.
2. For each document,
   (a) Draw a vector of topic proportions $\vec{\theta}_d \sim Dir(\alpha)$.
   (b) For each word,
      i. Draw a topic assignment $Z_{d,n} \sim Mult(\vec{\theta}_d), Z_{d,n} \in \{1,..,K\}$

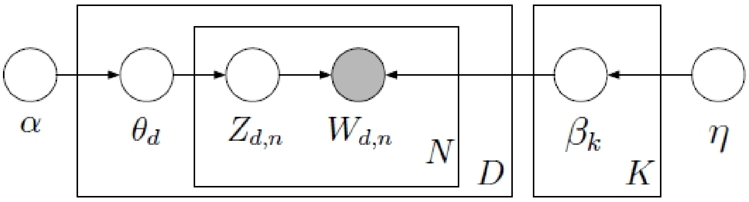


FIGURE 2.3: A Graphical Model representation of the LDA. Nodes denote random variables; edges denote dependence between random variables. Shaded nodes denote observed random variables; unshaded nodes denote hidden random variables. The rectangular boxes are plate notation, which denote replication.

The parameters of the prior are called hyperparameters. So, in LDA, both topic distributions, over documents and over words have also correspondent priors, which are denoted usually with α and η, also because are the parameters of the prior distributions are called hyperparameters.

For the symmetric distribution, a high α value means that each document is likely to contain a mixture of most of the topics, and not any single topic specifically. A low α value puts less such constraints on documents and means that it is more likely that a document may contain mixture of just a few, or even only one, of the topics. Likewise, a high η value means that each topic is likely to contain a mixture of most of the words, and not any word specifically, while a low value means that a topic may contain a mixture of just a few of the words.

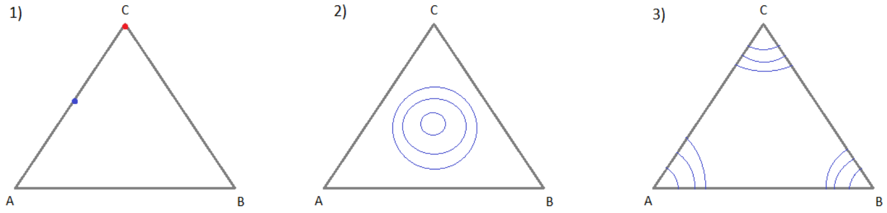


Figure 2.4: The role of the α parameter for the symmetric distribution. Each triangle represent an example of a 3 dimensional topic space. In the first triangle the points represents documents: the red one is 100% of topic C and the blue one is made of 50% of the topic A and 50% of the topic C. The second triangle represent a situation where there is an high value of α, the documents will be more concentrated on the center so, they will be a mixture of most of the topics. In the last figure is represented a value of alpha very low, this will bring the documents to began to few topics or only one.

If, on the other hand, the distribution is asymmetric, a high α value means that a specific topic distribution (depending on the base measure) is more likely for each document. Similarly, high η values means each topic is more likely to contain a specific word mix defined by the base measure.

In practice, a high α value will lead to documents being more similar in terms of what topics they contain. A high beta-value will similarly lead to topics being more similar in terms of what words they contain.

2.4 Natural Language Processing

Once the data is retrieved and cleaned, is ready to be analyzed. Dealing with document in natural language require specific techniques. Here are defined some concepts that will be useful for understand the next chapters. Some definitions are taken from [7] and [8] which I suggest the reading if interested in knowing more.

2.4.1 Structure Analysis and Tokenization

In this first step, documents are parsed so as to recognize their structure (title, abstract, section, paragraphs). For each relevant logical structure, the system then segments sentences into word tokens (hence the term tokenization). This procedure seems relatively easy but (a) the use of abbreviations may prompt the system to detect a sentence boundary where there is none, and (b) decisions must be made regarding numbers, special characters, hyphenation, and capitalization. In the expressions dont, Id, Johns do we have one, two or three tokens? In tokenizing the expression Afro-American, do we include the hyphen, or do we consider this expression as one or two tokens? For numbers, no definite rule can be found. We can simply ignore them or include them as indexing units. An alternative is to index such entities by their type, i.e., to use the tags date, currency, etc. in lieu of a particular date or amount of money. Finally, uppercase letters are lowercased. Thus, the title Export of cars from France is viewed as the word sequence export, of, cars, from, and france.

2.4.2 Stopwords removal

Very frequent word forms (such as determiners the, prepositions from, conjunctions and, pronouns you and some verbal forms is, etc.) appearing in a Stopword list are usually removed. Stopwords, also called empty words as they usually do not bear much meaning, represent noise in the retrieval process and actually damage retrieval performance, since they do not discriminate between relevant and non-relevant documents. Secondly because removing Stopwords allows one to reduce the storage size of the indexed collection, hopefully within the range of 30% to 50%.

2.4.3 Stemming and Lemmatization

Stemming and Lemmatization are the basic text processing methods for English text. The goal of both stemming and Lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

However, the two words differ in their flavor. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma . If confronted with the token saw, stemming might return just s, whereas Lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun. The two may also differ in that stemming most commonly collapses derivationally related words, whereas Lemmatization commonly only collapses the different inflectional forms of a lemma. [11]

2.4.4 Parts of Speech

Part-of-speech (POS) tagging is normally a sentence based approach . Given a sentence formed of a sequence of words, POS tagging tries to label (tag) each word with its correct part of speech (also named word category, word class, or lexical category).

Tag	Description
JJ	Adjective
RB	Adverb
VB	Verb, base form
IN	Preposition or subordinating conjunction
NN	Noun, singular or mass

Table 2.1: Some Part-of-speech tags used in the Penn Treebank Project. The full list is available here: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

For example the sentence ”I like potatoes” tagged with POS become : ”I / **PRP** like / **VBP** potatoes / **NNS** ” .

2.4.5 Dependency Parsing

Syntactic dependency representations of sentences have a long history in theoretical linguistics. Recently, they have found renewed interest in the computational parsing community due to their efficient computational properties and their ability to naturally model non-nested constructions, which is important in freer-word order languages such as Czech, Dutch, and German. This interest has led to a rapid growth in multilingual data sets and new parsing techniques. [9]

[11] http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

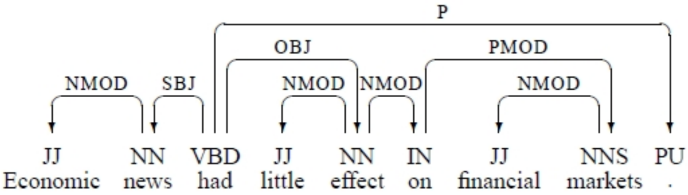


Figure 2.5: A Dependency Tree

The fundamental notion of dependency is based on the idea that the syntactic structure of a sentence consists of binary asymmetrical relations between the words of the sentence. The idea is expressed in the following way in the opening chapters of Tesnire [1959] :

> The sentence is an organized whole, the constituent elements of which are words. [1.2] Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives connections, the totality of which forms the structure of the sentence. [1.3] The structural connections establish dependency relations between the words. Each connection in principle unites a superior term and an inferior term. [2.1] The superior term receives the name governor. The inferior term receives the name subordinate. Thus, in the sentence Alfred parle [. . .], parle is the governor and Alfred the subordinate. [2.2]

2.5 Feature Selection

Feature selection is also called variable selection or attribute selection.

It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on.

Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, where as feature selection methods include and exclude attributes present in the data without changing them.

Examples of dimensionality reduction methods include Principal Component Analysis, Singular Value Decomposition and Sammons Mapping.

Feature selection is itself useful, but it mostly acts as a filter, muting out features that arent useful in addition to your existing features. [10]

2.5.1 The χ^2 test

Definition: The Chi-Square Test is the widely used non-parametric statistical test that describes the magnitude of discrepancy between the observed data and the data expected to be obtained with a specific hypothesis.

The observed and expected frequencies are said to be completely coinciding when the $\chi^2 = 0$ and as the value of χ^2 increases the discrepancy between the observed and expected data becomes significant. The following formula is used to calculate Chi-square:

$\chi^2 = \sum \frac{(O-E)^2}{E}$

Where:

O = Observed Frequency

E = Expected or Theoretical Frequency

The computed value of χ^2 is compared with the table value of χ^2 for a given degree of freedom and at a given significance level. If the calculated value exceeds the table value, then the difference between the observed frequencies and expected frequencies is said to be significant, i.e. it could not have arisen due to the fluctuations in simple sampling.

The following five basic conditions should be met before applying the chi-square test:

- The observation data must be independent of each other.
- The data should be expressed in original units and not in percentage or ratio form so that it can be easily compared.
- The data must be drawn randomly from the target population.
- The sample should include at least 50 observations.
- Every cell must have five or more observations. Each data entry is called a cell. In case, the observations are less than 5, then the value of χ^2 shall be overestimated and will result in the rejection of several Null Hypothesis. [12]

2.5.2 Information Gain

Information Gain is frequently employed as a term-goodness criterion in the field of Machine Learning. It measures the number of of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Let $c_{i=1}^{m}$ denote

[12] http://businessjargons.com/chi-square-test.html

the set of categories in the target space. The information gain of term t is defined to be :

$$G(t) = -\sum_{i=1}^{m} P(c_i)log(P(c_i))$$

$$+P(t)\sum_{i=1}^{m} P(c_i|t)log(P(c_i|t) + P(\bar{t}))\sum_{i=1}^{m} P(c_i|\bar{t})log(P(c_i|\bar{t}))$$

This definition is more general than the one employed in binary classification models. A more general form is used because text categorization problems usually have a m-ary category space(where m may be up to tens of thousands) , and we need to measure to goodness of a term globally with respect to all categories on average. [11]

2.6 Supervised Machine Learning

The aim of supervised, machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. As adaptive algorithms identify patterns in data, a computer ”learns” from the observations. When exposed to more observations, the computer improves its predictive performance.

Specifically, a supervised learning algorithm takes a known set of input data and known responses to the data (output), and trains a model to generate reasonable predictions for the response to new data.

For example, suppose you want to predict whether someone will have a heart attack within a year. You have a set of data on previous patients, including age, weight, height, blood pressure, etc. You know whether the previous patients had heart attacks within a year of their measurements. So, the problem is combining all the existing data into a model that can predict whether a new person will have a heart attack within a year.

You can think of the entire set of input data as a heterogeneous matrix. Rows of the matrix are called observations, examples, or instances, and each contain a set of measurements for a subject (patients in the example). Columns of the matrix are called predictors, attributes, or features, and each are variables representing a measurement taken on every subject (age, weight, height, etc. in the example). You can think of the response data as a column vector where each row contains the output of the corresponding observation in the input data (whether the patient had a heart attack). To fit or train a supervised learning model, choose an appropriate algorithm, and then pass the input and response data to it.

Supervised learning splits into two broad categories: classification and regression.

- In classification, the goal is to assign a class (or label) from a finite set of classes to an observation. That is, responses are categorical variables. Applications include spam filters, advertisement recommendation systems, and image and speech recognition. Predicting whether a patient will have a heart attack within a year is a classification problem, and the possible classes are true and false. Classification algorithms usually apply to nominal response values. However, some algorithms can accommodate ordinal classes.

- In regression, the goal is to predict a continuous measurement for an observation. That is, the responses variables are real numbers. Applications include forecasting stock prices, energy consumption, or disease incidence. [13]

2.6.1 The Naive Bayes Classifier

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. [14]

Given a class variable y and a dependent feature vector $x_1, ..., x_n$. Bayes theorem states the following relationship:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)}$$

Using the naive independence assumption that:

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i)|y$$

For all i this relation is simplified to:

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

[13] https://es.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html

[14] https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/

Since $P(x_1, ..., x_n)$ is constant given the input, we can use the following classification rule:

$\hat{y} = argmax P(y) \prod_{i=1}^{n} P(x_i|y)$

And we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set. [15]

[15] http://scikit-learn.org/stable/modules/naive_bayes.html

Chapter 3

State of the Art

This Chapter extend the previous one by going deep in several topics. First of all are described the State of the Art techniques that needs to be applied to Topic Model Algorithms for reach satisfactive results.

After that are discussed the state of the art techniques for Sentiment Analysis. The last paragraph, deals about an argument that is used in the last experiment: Probability Calibration. Not all Machine Learning methods offer good probability estimations for their predictions, are discussed methodologies that can help improve the probability estimations.

3.1 Topic Modelling on Microblogs

Twitter, or the world of 140 characters poses serious challenges to the efficacy of topic models on short , messy text. While topic models such as Latent Dirichlet Allocation (LDA) have a long history of successful application to news articles and academic abstracts, they are often less coherent when applied to Microblog contents like Twitter.

Several papers are dedicated to this problem and propose various solutions to this. Mehrotra et al. [12] try to obtain better LDA topics without modifying the basic machinery of LDA, in particular they present various pooling schemes to aggregate tweets into ”macro-documents” for use as a training data to build LDA models. The motivation behind tweet pooling is that individual tweets are very short (<= 140 characters) and hence treating each tweet as an individual document does not present adequate term co-occurrence data within documents. Aggregating tweets which are similar in some sense (semantically, temporally, etc.) enriches the content present in a single document from which the LDA can learn a better topic model:

- **Author-wise Pooling**: Pooling tweets according to author.This method show to be superior to unpooled Tweets. For this method, document for each author is built, which combines all tweets they have posted.

- **Burst-score wise Pooling** A trend on Twitter (sometimes referred to as a trending topic) consists of one or more terms and a time period, such that the volume of messages posted for the terms in the time period exceeds some expected level of activity. In order to identify trends in Twitter posts, unusual bursts" of term frequency can be detected in the data.We run a simple burst detection algorithm to detect such trending terms and aggregate tweets containing those terms having high burst scores. To identify terms that appear more frequently than expected, we will assign a score to terms according to their deviation from an expected frequency. Assume that M is the set of all messages in our tweets dataset, R is a set of one or more terms (a potential trending topic) to which we wish to assign a score, and $d \in D$ represents one day in a set D of days.We then define $M(R; d)$ as the subset of Twitter messages in M such that (1) the message contains all the terms in R and (2) the message was posted during day d. With this information, we can compare the volume in a specific day to the other days. Let $Mean(R) = \frac{1}{|D|}\sum_{d \in D} M(R, d)$ over the days $d \in D$. The burst-score is then defined as:

  burst-score$(R, d) = \frac{|M(R,d) - Mean(R)|}{SD(R)}$

- **Temporal Pooling**: When a major event occurs, a large number of users often start tweeting about the event within a short period of time. To capture such temporal coherence of tweets, the fourth scheme and our second novel pooling proposal is known as Temporal Pooling, where we pool all tweets posted within the same hour.

- **Hashtag-based Pooling**: A Twitter hashtag is a string of characters preceded by the hash (#) character. In many cases hashtags can be viewed as topical markers, an indication to the context of the tweet or as the core idea expressed in the tweet, therefore hashtags are adopted by other users that contribute similar content or express a related idea. One example of the use of hashtags is "ask GAGA anything using the tag #GoogleGoesGaga for her interview! RT so every monster learns about it!! " referring to an exclusive interview for Google by Lady Gaga (singer). For the hashtag-based pooling scheme, we create pooled documents for each hashtag. If any tweet has more than one hashtag, this tweet gets added to the tweet-pool of each of those hashtags.

Ramage et al. [13] use a partially supervised learning model (Labeled LDA) that maps the content of the Twitter feed into dimensions. These dimensions correspond roughly to substance, style, status, and social characteristics of posts. So while the latent dimensions in twitter can help identify broad trends, several classes of tweets specific labels are applied to subsets of the posts. For example hashtags ,emoticons and social signals such as replies, mentions (@user) , questions(?) .

3.2 Validation of Topic Models

The validation of the topics obtained with Topic Model can be performed in several ways, by using metrics or by human judgment.

Chang et al. [14] have compared several metrics with human judging techniques. The techniques analyzed are the following:

- **Word intrusion**: For each trained topic, take the six most probable words, substitute one of them with another, randomly chosen word (an intruder) and see whether a human can reliably tell which one it was. If so, the trained topic is topically coherent if not, the topic has no discernible theme . For example, most people readily identify apple as the intruding word in the set {dog, cat, horse, apple, pig, cow} because the remaining words make sense together. While for the set {car, teacher, platypus, agile, blue, Zaire} identifying the intruder is more difficult. This will bring people to choose the intruder at random, implying a topic with poor coherence.

  Let w_k^m be the index of the intruding word generated from the k topic inferred by model m. Let $i_{k,s}^m$ be the intruder selected by the subject s generated from the topic k inferred by the model s. Be S the total number of subjects. The model precision is defined by the fraction of subjects agreeing with the model: $MP_k^m = \sum_s 1(i_{k,s}^m = w_k^m)/S$

- **Topic intrusion**: Subjects are shown the title and a snippet from a document. Along with the document they are presented with four topics. Three of those topics are the highest probability topics assigned to that document. The remaining intruder topic is chosen randomly from the other low-probability topics in the model. The subject is instructed to choose the topic which does not belong with the document. As before, if the topic assignment to documents were relevant and intuitive, we would expect that subjects would select the topic we randomly added as the topic that did not belong. The *topic log odds* is defined as a quantitative measure of the agreement between the model and human judgments on this task. Let ${}^{d}_{m}$ denote model m's point estimate of the topic proportions vector associated with document d. Further let $j^m_{d,s}$ be the intruding topic selected by subject s for document d on model m and let j^m_d denote the "true" intruder. In other words the topic log odds is the log ratio of the probability mass assigned to the true intruder to the probability mass assigned to the intruder selected by the subject:

$TLO^m_d = (\sum_s \theta^m_{d,j^m_{d,*}} - \theta^m_{d,j^m_{d,s}})/S$

- **Log-Likelihood**: A predictive metrics. The dataset need to be splitted in training and test. Let be w_d the documents in the test set and be the model described by the topic matrix Φ . The log-likelihood is defined as:

  $L(w) = \ log\ p(w|\Phi) = \ \sum_d log\ p(w_d|\Phi)$

- **Perplexity** It make use of the log-likelihood. Is defined as :

  $perplexity(test\ set\ w) \ = \ exp - \left\{ \frac{L(w)}{\sum_{d=1}^{D} \sum_{v=1}^{V} n_{jv}} \right\}$ Which is a decreasing function of the log-likelihood. The lower the perplexity, the better the model.

The paper shows that log-likelihood (and consequently perplexity) and human judgment is not correlated. Sometimes are also slightly uncorrelated.

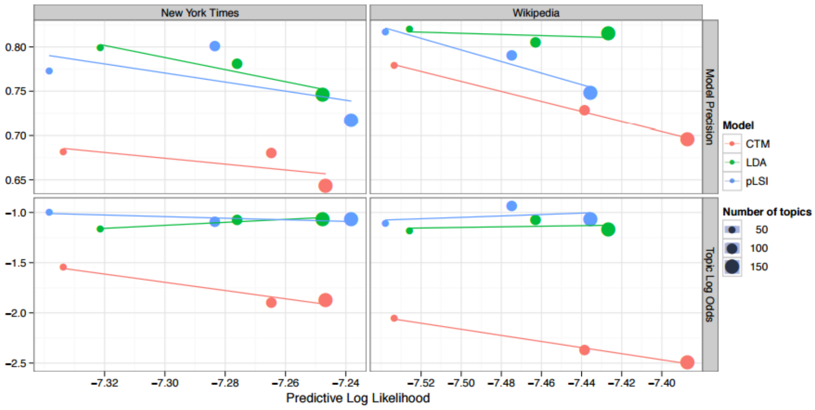


FIGURE 3.1: Comparison of metrics for LDA from Chang et al. . Comparison between Word Intrusion (top row), Topic Intrusion (bottom row) and the log likelihood. Each point is colored by model and sized according to the number of topics used to fit the model. Each model is accompanied by a regression line. Increasing likelihood does not increase the agreement between human subjects and the model for either task (as shown by the downward-sloping regression lines).

Roder et al. [15] compare other metrics with human judging and some of them seems to be very promising:

- **UMass Coherence** : Is an asymmetrical confirmation measure between top word pairs (smoothed conditional probability). The summation of UMass coherence accounts for the ordering among the top words of a topic:

  $C_{UMass} = \frac{2}{n \cdot (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} log \frac{P(w_i, w_j) + \epsilon}{P(w_j)}$

  Word probabilities are estimated based on document frequencies of the original documents used for learning the topics.

  The main idea of this coherence is that the occurrence of every top word should be supported by every top preceding top word. Thus, the probability of a top word to occur should be higher if a document already contains a higher order top word of the same topic. Therefore, for every word the logarithm of its conditional probability is calculated using every other top word that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts. The single conditional probabilities are summarized using the arithmetic mean.

- **UCI Coherence** : based on pointwise mutual information, the formula is:

  $C_{UCI} = \dfrac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j)$

  $PMI(w_i, w_j) = log \dfrac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}$

  The word co-occurrence counts are derived using a sliding window . For every word pair the PMI is calculated. The arithmetic mean of the PMI values is the result of this coherence.

- **normalized PMI** :

  $v_{ij} = NPMI(w_i, w_j)^{\gamma} = \left(\dfrac{log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)}}{-log(P(w_i, w_j) + \epsilon)} \right)^{\gamma}$

- **CV** : Is a combination between the indirect cosine measure with the NPMI and the boolean sliding window.

- **Direct Coherent Measure** (c_p) : Also this is a combination. It combines Fitelsons confirmation measure with the boolean sliding window.

The comparison of the metrics is shown in the plot below. The c_v metrics is the one who is closer to human judgment. It reaches is peak on a sliding window of 110.

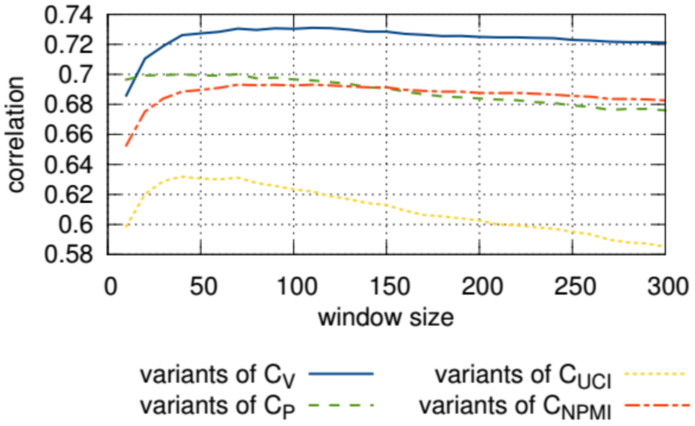
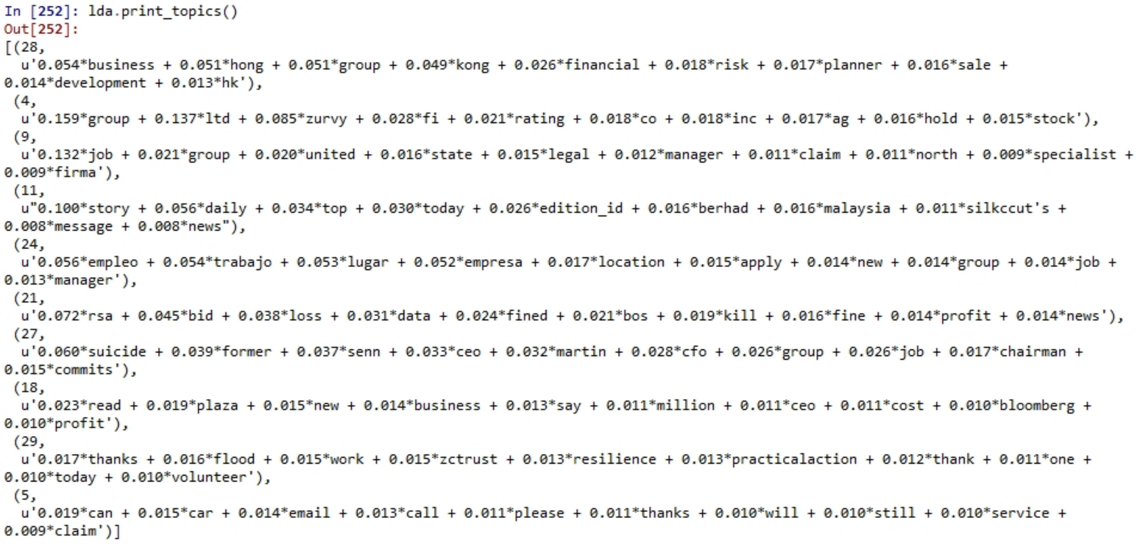


Figure 3.2: Comparison of metrics for LDA from Roder et al.

3.3 Visualization of Topic Models

Interpreting the output of a topic model can be challenging as can be seen in this example.



```
In [252]: lda.print_topics()
Out[252]:
[(28,
  u'0.054*business + 0.051*hong + 0.051*group + 0.049*kong + 0.026*financial + 0.018*risk + 0.017*planner + 0.016*sale +
0.014*development + 0.013*hk'),
 (4,
  u'0.159*group + 0.137*ltd + 0.085*zurvy + 0.028*fi + 0.021*rating + 0.018*co + 0.018*inc + 0.017*ag + 0.016*hold + 0.015*stock'),
 (9,
  u'0.132*job + 0.021*group + 0.020*united + 0.016*state + 0.015*legal + 0.012*manager + 0.011*claim + 0.011*north + 0.009*specialist +
0.009*firma'),
 (11,
  u"0.100*story + 0.056*daily + 0.034*top + 0.030*today + 0.026*edition_id + 0.016*berhad + 0.016*malaysia + 0.011*silkccut's +
0.008*message + 0.008*news"),
 (24,
  u'0.056*empleo + 0.054*trabajo + 0.053*lugar + 0.052*empresa + 0.017*location + 0.015*apply + 0.014*new + 0.014*group + 0.014*job +
0.013*manager'),
 (21,
  u'0.072*rsa + 0.045*bid + 0.038*loss + 0.031*data + 0.024*fined + 0.021*bos + 0.019*kill + 0.016*fine + 0.014*profit + 0.014*news'),
 (27,
  u'0.060*suicide + 0.039*former + 0.037*senn + 0.033*ceo + 0.032*martin + 0.028*cfo + 0.026*group + 0.026*job + 0.017*chairman +
0.015*commits'),
 (18,
  u'0.023*read + 0.019*plaza + 0.015*new + 0.014*business + 0.013*say + 0.011*million + 0.011*ceo + 0.011*cost + 0.010*bloomberg +
0.010*profit'),
 (29,
  u'0.017*thanks + 0.016*flood + 0.015*work + 0.015*zctrust + 0.013*resilience + 0.013*practicalaction + 0.012*thank + 0.011*one +
0.010*today + 0.010*volunteer'),
 (5,
  u'0.019*can + 0.015*car + 0.014*email + 0.013*call + 0.011*please + 0.011*thanks + 0.010*will + 0.010*still + 0.010*service +
0.009*claim')]
```

Figure 3.3: Output of LDA, an example from a model used in our analysis.

A huge amount of words concatenated with numbers is not the best way to interpret the results of a model. Below several visualization techniques are proposed in order to improve the interpretability (They are taken from [16] and [17]) .

- Stacked Bar Chart

  The idea underlying the stacked bar chart is that each text has some proportion of its words associated with each topic. Because the model assumes that every word is associated with some topic, these proportions must add up to one. For example, in a three topic model, text number 1 might have 50% of its words associated with topic 1, 25% with topic 2, and 25% with topic 3. The stacked bar chart represents each document as a bar broken into colored segments matching the associated proportions of each topic. The stacked bar chart below expresses the topic proportions found in the six novels in the austen-bront corpus.

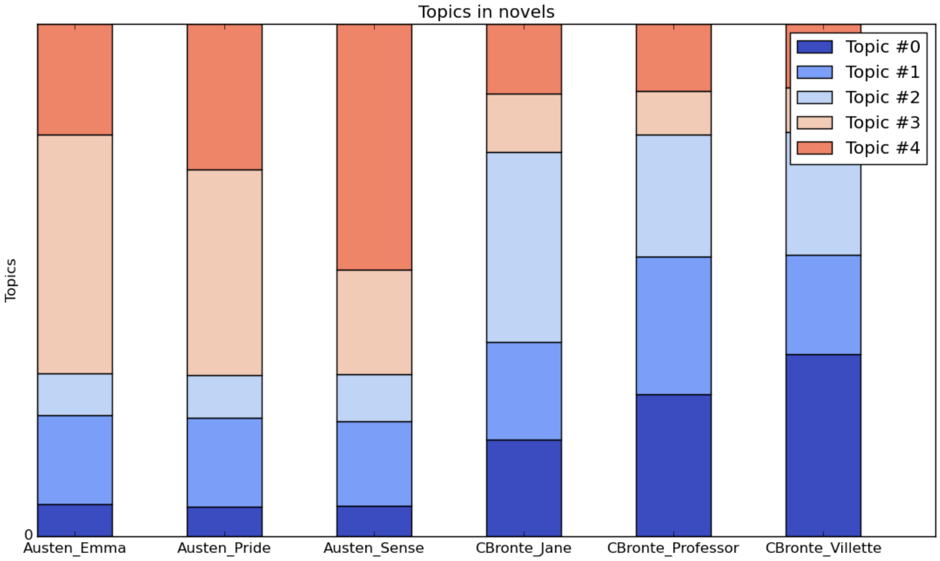


FIGURE 3.4: A Stacked Bar Chart

- Heatmap

  Another useful visualization of topic shares is the heatmap. A heat map (or heatmap) is a graphical representation of data where the individual values contained in a matrix are represented as colors.

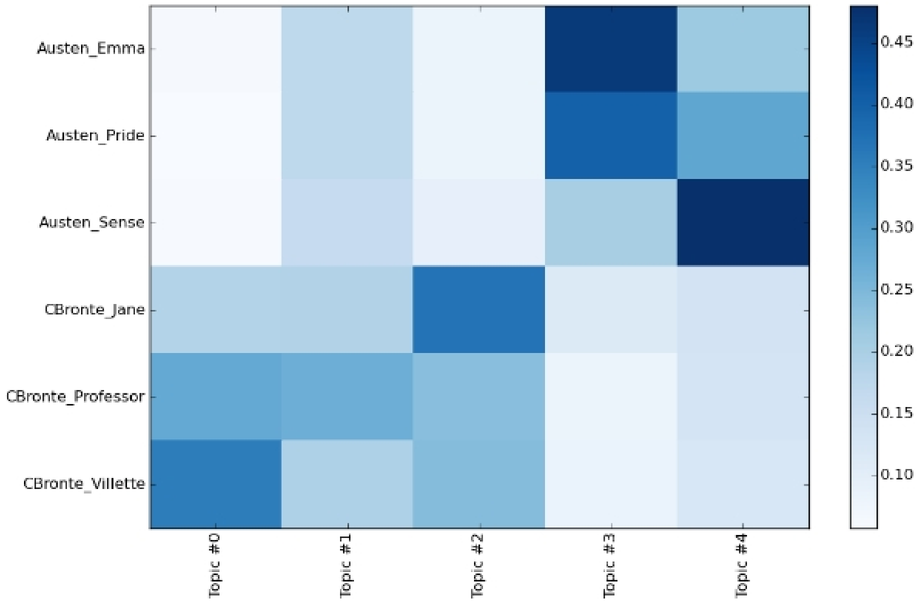


Figure 3.5: An Example of Heatmap, is visible that topics 3 and 4 are quite correlated with the documents of Austen, while topic 0 and 2 dominates in the CBronte ones

- Topic-words Associations

  An alternative to the crude visualization of words and probabilities that is the output of LDA can be to plot the words for each topic . For each topic vary the size of the word based on his weight.

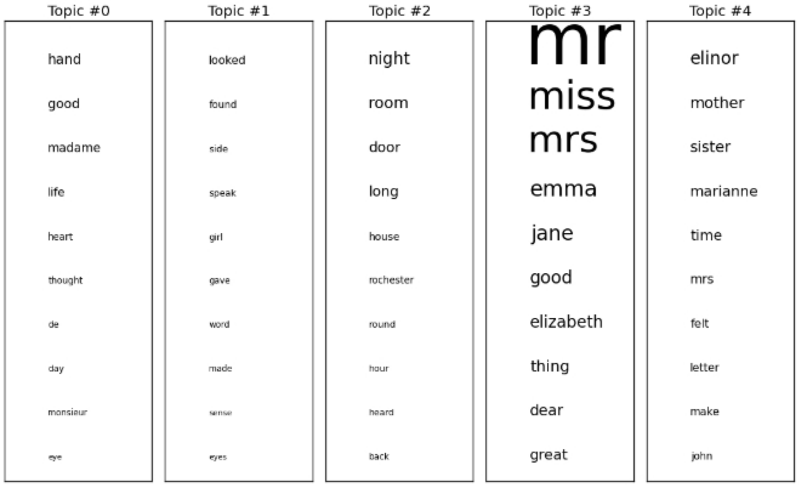
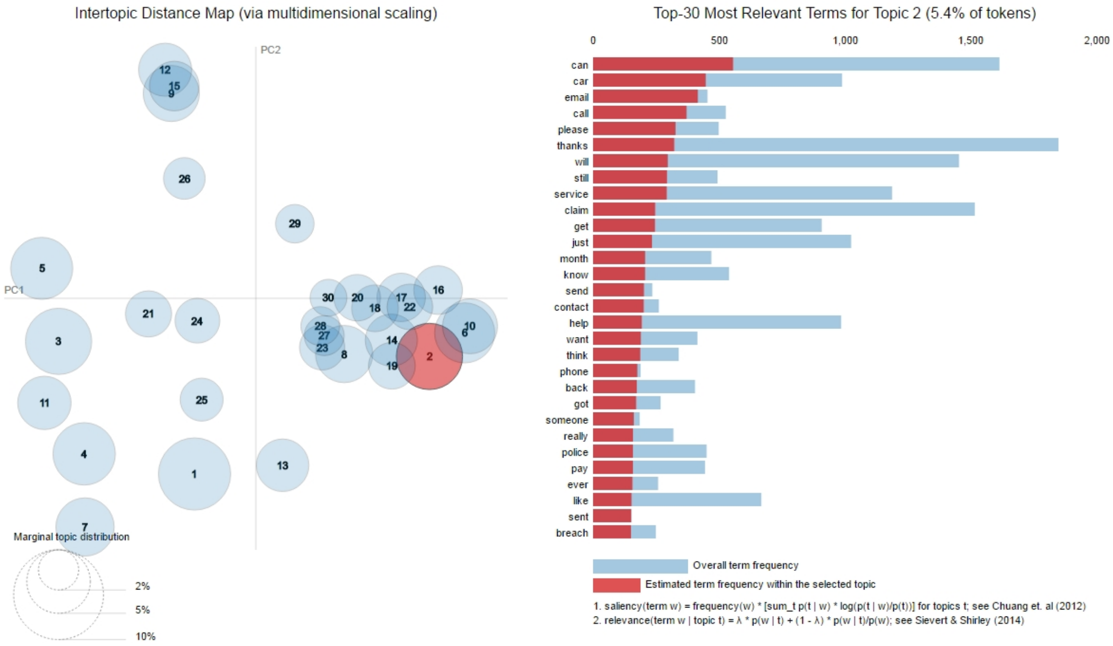


Figure 3.6: Example of Topic-words Association. Is visible that the topic 3 is much more dense than the topic 0.

- LDAvis

  Last but not the least, a very fascinating library for visualizing topic models. Is available in python [1] and R [2] .



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

FIGURE 3.7: Example of LDAvis.

On the left side are represented the first two components of a Principal Component Analysis performed on the topic space. Each topic is a circle, the biggest is the circle more is representative. On the right side is possible to see the most representative words for this topic. The red bar represent the frequency of the world in the topic while the blue bar represent the frequency of the world in the whole corpus.

LDAvis allows a very nice interaction with the user, for example in the screen below I have done two things: first I have moved the slide on top and i set λ to 0.37, this will makes emerge words that are unique to that specific topic. A too low value for λ will makes emerge stopwords or words that are yes unique for that topic but could be not good for interpreting the topic.

In this second example I have put the mouse on the word *"car"*. This change the circles area on the map. Now the size of the circle represents how important is the world "car" for that specific topic.

[1] https://pyldavis.readthedocs.io/en/latest/
[2] https://cran.r-project.org/web/packages/LDAvis/index.html

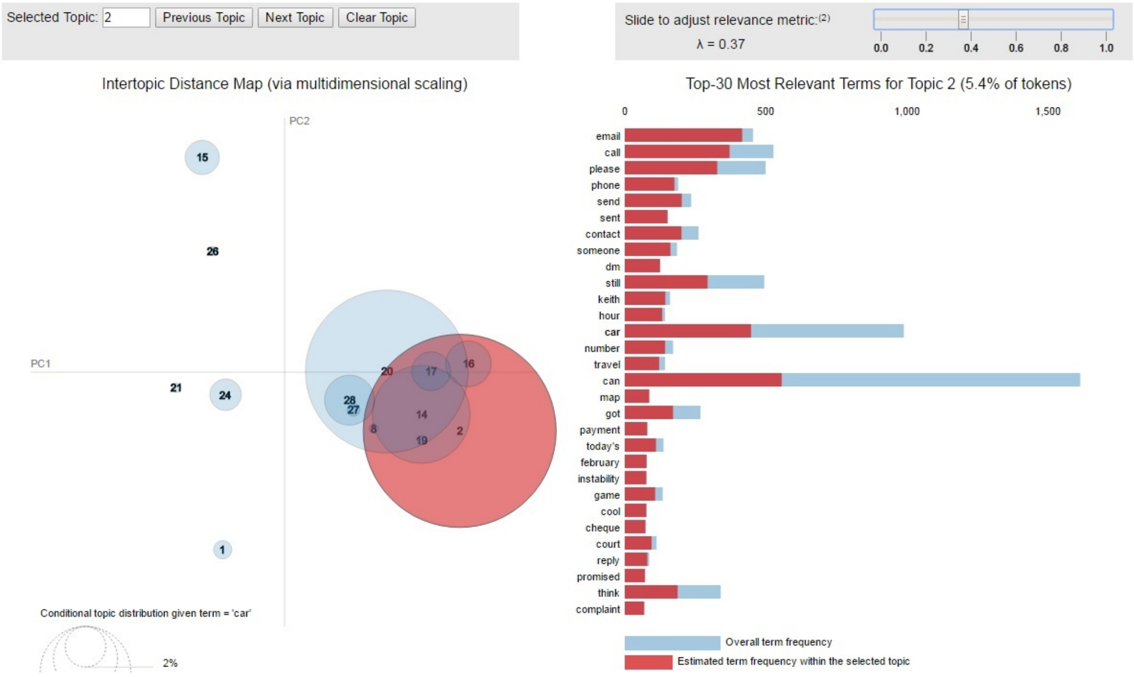


FIGURE 3.8: Another example of LDAvis.

3.4 Sentiment Analysis on Microblogs

Sentiment analysis is a line of research that combines techniques from various fields such as Natural Language Processing and Machine Learning to extract, from a given piece of text, information on the authors personal impressions.

Several approaches are available, for example Pandey et al [18] divide Sentiment Analysis into two main sub tasks:

- **Subjectivity Recognition**: which is usually a binary problem that consists in deciding whether a given text contains personal impressions or not.
- **Polarity Detection**: once obtained the data with personal impressions, try to extract concrete information from the subjective writing.

For the Polarity detection several variants are present in the literature. There is who consider the problem as a binary-problem (positive or negative). Some others consider also the neutral class , thus positive , negative and neutral. Some other works (for example [19]) try to enlarge the spectrum of the emotions, so not only positive or negative but also happy, unhappy, skeptical and playful.

3.5 Lexicons for Sentiment Analysis

Typically, lexicon-based approaches for sentiment classification are based on the insight that the polarity of a piece of text can be obtained on the ground of the polarity of the words which compose it.

Semeraro et al [20] perform a comparison among several lexicon sources available on the Web. They test the lexicons on Twitter Data. They evaluate four lexicons: SentiWordNet , WordNet-Affect , MPQA and SenticNet. They make use of the lexicons in a supervised way. They make use of a labeled training also in order to leverage the weights of the sentiment words. They test several configurations and these are the results.

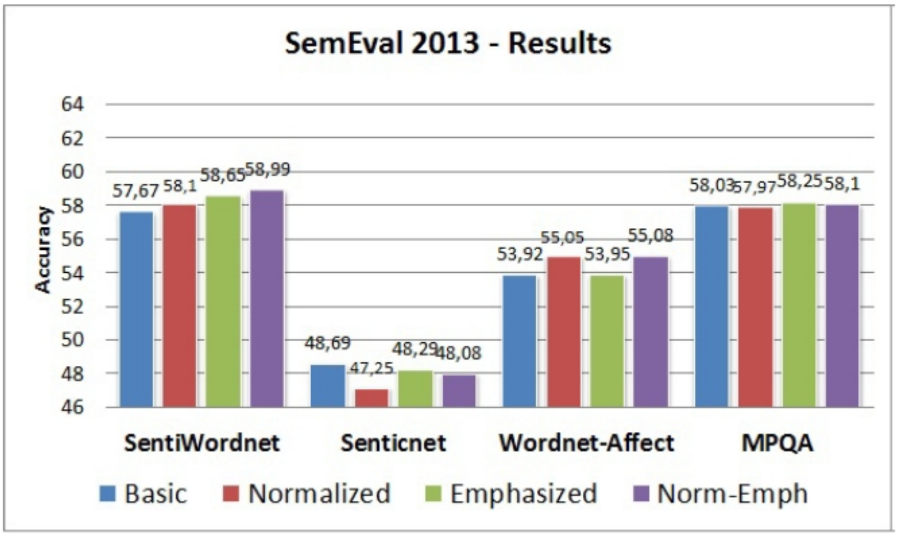
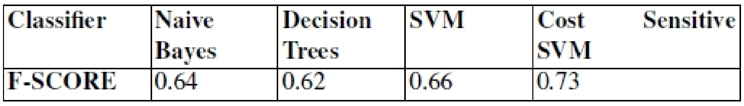


Figure 3.9: Performance of Several Lexicon methods for Sentiment Analysis from Semeraro et al for the three class problem (positive , negative and neutral) performed on the SemEval 2013 Data [21]

On the same dataset Kolchyna et al [22] perform several approaches. Some includes the use of lexicons and others adopt supervised models. The supervised methods outperform the lexicon ones and their results are the following:



Classifier	Naive Bayes	Decision Trees	SVM	Cost Sensitive SVM
F-SCORE	0.64	0.62	0.66	0.73

Figure 3.10: Performance of Supervised methods for Sentiment Analysis from Kolchyna et al for the three class problem (positive , negative and neutral) performed on the SemEval 2013 Data

The domain of a lexicon is also important. A word can have a different polarity on different domains. In general statistical and machine

Marquez et al [23] have built a lexicon specific for twitter. They use several seed lexicons in order to extract sentiment words from unlabeled tweets.

3.6 Soft Labeling on Microblogs

Labeling a training set can be costly and time consuming. Thus for the Polarity Detection task on Twitter is common [24] [25] [26] to label the training corpora automatically by using tweets with smileys. So tweets containing happy faces (:-) , ;) , etc.) will be used as a positive corpus , while the ones containing sad faces (:(, :'(, ...) will constitute the negative corpus.

This approach has shown good results not only in a specific domain but also in the general domain, so even by using a generic domain training corpus is possible to reach good results in a specific domain. However this results are limited to the two class problem. So the sentiment is classified into positive and negative only.

3.7 Probability Calibration of Machine Learning Models

When performing classification you often want not only to predict the class label, but also obtain a probability of the respective label. This probability gives you some kind of confidence on the prediction. Some models gives poor estimates of the class probabilities and some even do not not support probability prediction. The calibration module included in the scikit-learn package in Python allows to better calibrate the probabilities of a given model, or to add support for probability prediction.

Below I provide some example of this library in action, everything is taken from the H. Metzen blog [27].

Well calibrated classifiers are probabilistic classifiers for which the output of the predict_proba method can be directly interpreted as a confidence level. For instance, a well calibrated (binary) classifier should classify the samples such that among the samples to which it gave a predict_proba value close to 0.8, approximately 80% actually belong to the positive class. The following plot compares how well the probabilistic predictions of different classifiers are calibrated:

Logistic Regression returns well calibrated predictions by default as it directly optimizes log-loss. In contrast, the other methods return biased probabilities; with different biases per method:

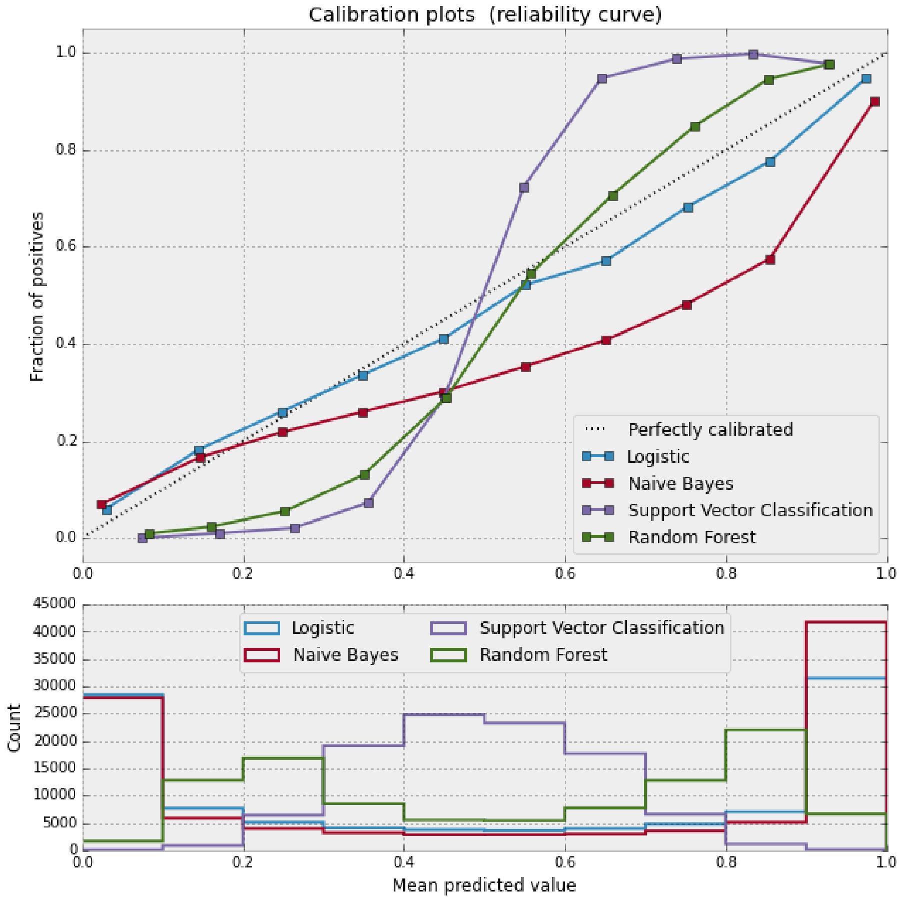


FIGURE 3.11: Comparison of Probabilistic Predictions

Naive Bayes (GaussianNB) tends to push probabilities to 0 or 1 (note the counts in the histograms). This is mainly because it makes the assumption that features are conditionally independent given the class, which is not the case in this dataset which contains 2 redundant features.

Linear Support Vector Classification (LinearSVC) shows an even more sigmoid curve as the RandomForest Classifier, which is typical for maximum-margin methods , which focus on hard samples that are close to the decision boundary (the support vectors).

Two approaches for performing calibration of probabilistic predictions are provided: a parametric approach based on Platt's sigmoid model and a non-parametric approach based on isotonic regression (sklearn.isotonic). Probability calibration should be done on new data not used for model fitting. The class CalibratedClassifierCV uses a cross-validation generator and estimates for each split the model parameter on the train samples and the calibration of the test samples. The probabilities predicted for the folds are then averaged. Already fitted classifiers can be calibrated by CalibratedClassifierCV via the parameter cv="prefit". In this case, the user has to take care manually that data for model fitting and calibration are disjoint.

The following experiment is performed on an artificial dataset for binary classification with 100.000 samples (1.000 of them are used for model fitting) with 20 features. Of the 20 features, only 2 are informative and 10 are redundant. The figure shows the estimated probabilities obtained with logistic regression, a linear support-vector classifier (SVC), and linear SVC with both Isotonic calibration and Sigmoid calibration. The calibration performance is evaluated with Brier score brier_score_loss, reported in the legend (the smaller the better).

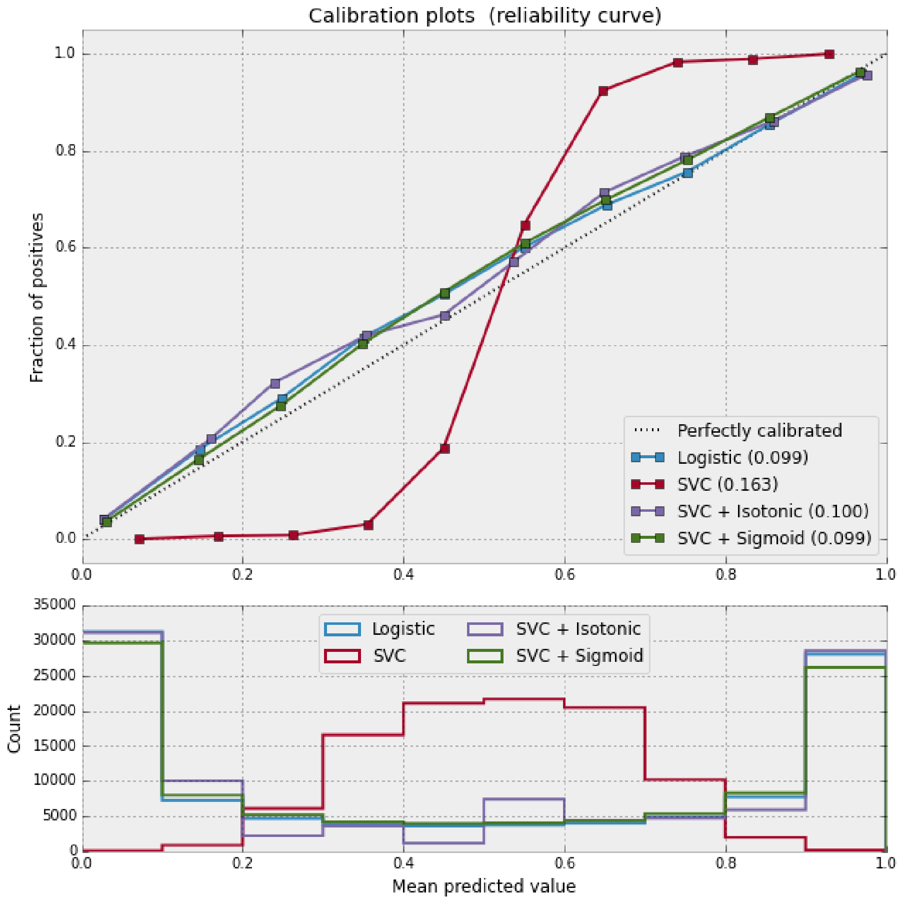


Figure 3.12: SVM with probability calibration

One can observe here that logistic regression is well calibrated as its curve is nearly diagonal. Linear SVC's calibration curve has a Sigmoid curve, which is typical for an under-confident classifier. In the case of LinearSVC, this is caused by the margin property of the hinge loss, which lets the model focus on hard samples that are close to the decision boundary (the support vectors). Both kinds of calibration can fix this issue and yield nearly identical results. The next figure shows the calibration curve of Gaussian naive Bayes on the same data, with both kinds of calibration and also without calibration.

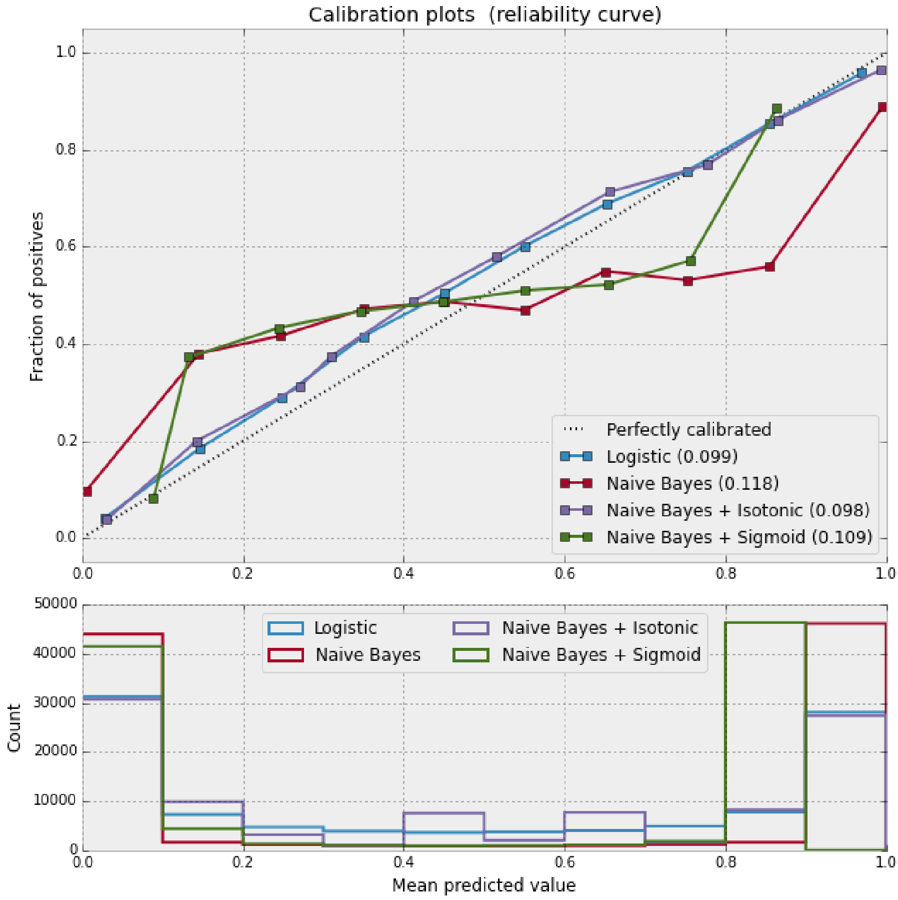


FIGURE 3.13: Gaussian Naive Bayes with probability calibration

One can see that Gaussian naive Bayes performs very badly but does so in an other way than linear SVC: While linear SVC exhibited a Sigmoid calibration curve, Gaussian naive Bayes' calibration curve has a transposed-Sigmoid shape. This is typical for an over-confident classifier. In this case, the classifier's overconfidence is caused by the redundant features which violate the naive Bayes assumption of feature-independence.

Calibration of the probabilities of Gaussian naive Bayes with Isotonic regression can fix this issue as can be seen from the nearly diagonal calibration curve. Sigmoid calibration also improves the brier score slightly, albeit not as strongly as the non-parametric Isotonic calibration. This is an intrinsic limitation of Sigmoid calibration, whose parametric form assumes a Sigmoid rather than a transposed-Sigmoid curve. The non-parametric Isotonic calibration model, however, makes no such strong assumptions and can deal with either shape, provided that there is sufficient calibration data. In general, Sigmoid calibration is preferable if the calibration curve is Sigmoid and when there is few calibration data while Isotonic calibration is preferable for non- Sigmoid calibration curves and in situations where many additional data can be used for calibration.

Chapter 4

First Experiment - Information Retrieval

4.1 Retrieve the data from Twitter

The Zurich Insurance Group reside in different country all over the world. Thus different language are used on twitter. This research focus on tweets written in English. When retrieving tweets is possible to filter them by language with the parameter LANG. However still a small subset of tweet written in other languages is retrieved, in particular tweets written in more than one language (For example : *"Senior Planning Analyst * http:// bit.ly/VmWsX2 * Empresa: Zurich Insurance Company Ltd * Lugar: Zurich #empleo #trabajo #suiza"*) .

Twitter offer an API in order to retrieve information. Several interfaces are available. In order to retrieve tweet by a keyword is possible to use the API.search. In python one way to approach the API is with the Tweepy library [1] .

4.2 Information Filtering

The first problem is to formulate proper queries in order to retrieve the data. Intuitively a specific query like *"zurich insurance group"* would do the job. However performing a specific query could potentially rule out an important amount of tweet. Twitter is the world where everyone express concepts in 140 characters so it reasonable to think that people would refer to the Zurich group also in other ways. In order to include more cases is possible to make a more generic query like: *"zurich"* . The problem of this query is

[1] http://www.tweepy.org/

that introduce an enormous amount of noise due to ambiguities. Just think about the city of Zurich, the airport of Zurich and so on . Furthermore Zurich has got several official pages. It is important to individuate them and retrieve the tweets related to them. They are basically of two categories, messages (@) and hashtags (#).

Finally the following queries has been performed, the results are then combined in order to obtain a ”specific corpus” and a ”generic one”. The tweets obtained below are generated from October 2007 to October 2016

Query	**Cat.**	**Description**	**Size**
1) zurich	Gen.	tweets that contains the keyword zurich	1.182.447
2) zurich insurance	Spec	tweets that contains the keywords zurich insurance	57.078
3) @zurich OR @zurichinsuk OR @zurichinsider OR @zurichaustralia OR @zurichnanews OR @zurichireland OR @zurichmunicipal OR @zurichcanada	Spec	message directed to the official pages	31.009
4) #zurichinsuk OR #zurichinsider OR #zurichaustralia OR #zurichnanews OR #zurichireland OR #zurichmunicipal OR #zurichcanada	Spec	tweets that includes the hashtags of the official pages	48

Table 4.1: Queries Performed on Twitter

The corpora obtained are not independent. Most of the tweet obtained in the queries 2, 3 and 4 are subsets of the first one. Also the sets 2 , 3 and 4 are not mutually exclusive and they share several common tweets. Furthermore the the query 4 doesn't include the hashtag #zurich , because it contains a lot of noise . The first query return as well tweets with the hashtag #zurich .

The size of the first corpus suggest that the generic corpus present an important amount of noise.

4.2.1 Exploratory Analysis of The Specific Corpus

The tweets from the specifics corpora are merged. The redundant tweets are removed. The retweets are also discarded with the following criteria: Consider the following tweets:

- *”I like potatoes”*
- *”RT:I like potatoes”*
- *”I totally agree!!! :) RT:I like potatoes”*

The first retweet doesn't add any new information, furthermore is just redundant and thus is discarded. The third one add contents to the original tweet, thus is kept.

After the merging phase a corpus of 65.504 tweet is obtained. Below the timestamps of the tweets are aggregated by month and the distribution over time is showed. Is visible that until 2012 a very small amount of tweet is published, the trend is growing but very slowly. From 2012 the things start to change. This is because the Zurich Group start join Twitter on this period and begin to create his pages on several countries. The trend keeps growing until the present.

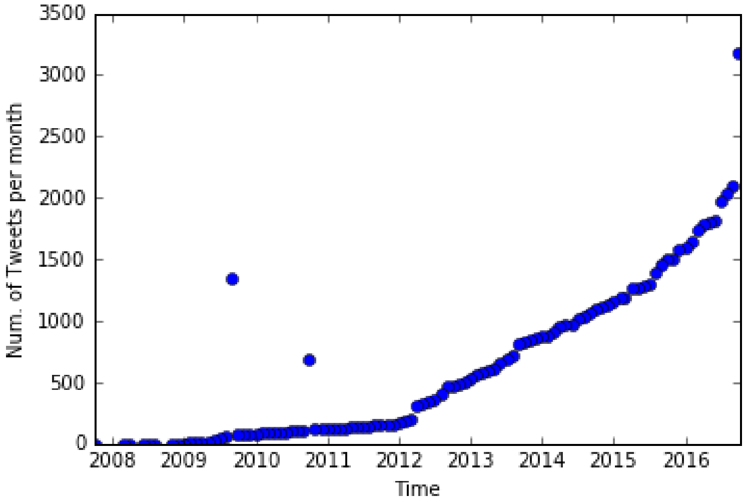


Figure 4.1: Relevant Corpus - Distribution over time

In the histogram are showed the most frequent authors in the relevant corpus. The Zurich pages dominates the rank followed by several news pages and job announces pages. This combined to a quick look on a sample of the corpus suggest that still doesn't show interesting information. Is true that they deal about the Zurich Group but still doesn't show the information that interest us. Much effort is required in order to extract relevant information.

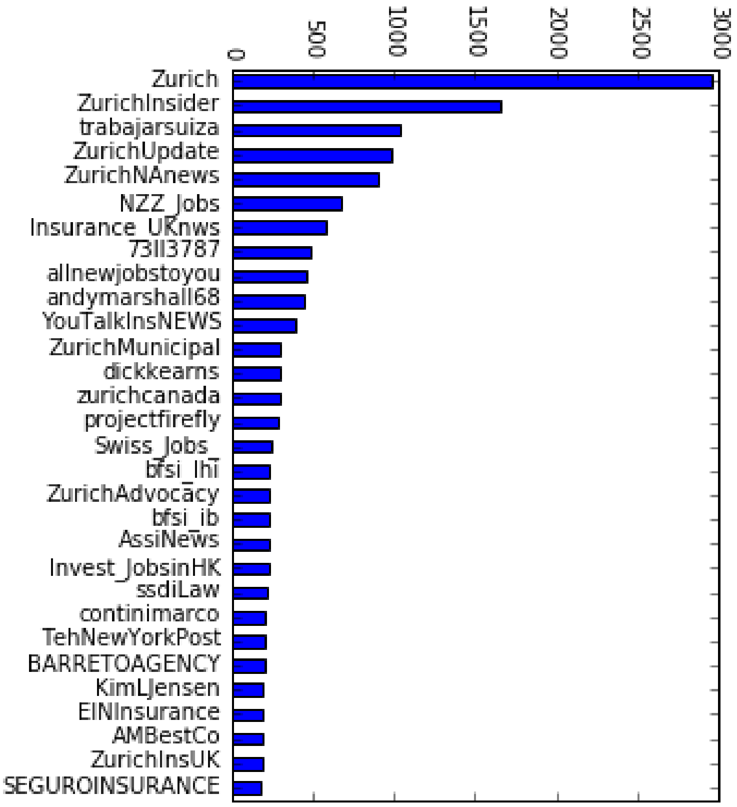


Figure 4.2: Most Frequent authors in the Relevant Corpus

```
'Credit Portfolio Manager: Zurich Insurance Group Location : Milano
LOM IT Zurich is one of the world\xe2\x80\x99s leadin... http:// bit.ly/2dSAgWp'
'#Jobs #Boston (USA-MA-Boston) Medical Stop Loss Underwriter I:
Zurich Insurance is currently looking for a Me... http:// tinyurl.com/za5bfhf'
'Zurich Insurance transformation designed from the customer
back #DF16 very #customer -adaptive'
'Great comments from Emma @Zurich at #DF16 about #b2b customers
bringing consumer experience expectations to the workplace. #customerobsessed'
'Customer stories live at #df16 as #ZurichInsurance transforming
with ZurichFutureYou built on #Salesforce pic.twitter.com/btyfUBl3MH'
'Cyber security & privacy risks for financial institutions @AccentureSecure
@BarclaysUK @CooleyLLP @JonesDay @Zurich http:// bit.ly/2bcsZzo'
'Credit Portfolio Manager: Zurich Insurance Group Location : Milano LOM
IT Zurich is one of the world\xe2\x80\x99s leading... http:// fb.me/47GWKeQt7'
```

Table 4.2: A sample of the tweets included in the 'Relevant' Corpus

4.2.2 Application of the LDA

Is difficult to have an idea of what 65.000 tweets are talking about. My approach try to apply LDA to the corpus in an exploratory way in order to find if and where the significant information is located.

As already stated in the State of the Art chapter, LDA can have problems in dealing with very short document like tweets. Several preliminary models have been applied to the corpus.

A preprocessing phase is performed to the corpus. Each tweet is tokenized using the TweetTokenizer [2] , a tokenizer that is more specialized in dealing with tweets. Each word is converted to lowercase. After that stop words are removed from a list of common stop words in English. The URLs/mail addresses are removed because are considered not relevant for this analysis. Furthermore the hashtags and message characters are removed (For example *@zurich* and *#zurich* are transformed to *zurich* . The repetition of more than 2 letters are truncated because are considered useless and they increase the sparsity in the corpus. Just thing about *goooooooooood*, is a way to empathize the world good but the same concept can be expressed with just *good* . Finally each token is lemmatized . [3]

Before	**After**
info@zurich.com	REMOVED
www.google.com	REMOVED
#Zurich	zurich
GoooOOOOod	good
65.45 , :)	:)

Table 4.3: Example of the pre-processing used for LDA.

Each tweet is considered a separate document. Several preliminary models are performed. The result are not satisfying.

id	topic words
0	zurich , insurance , rsa , bid , new , zurickinsuk , group , news , global , takeover
1	zurich , insurance , zurichnanews , zurickinsuk , follow , suicide , will , news , expansion
2	zurich , insurance , zurichinsider , global , can , zurichnanews , risk , help , social
3	zurich , insurance , zurichnanews , risk , posted , job , hong , kong , company , group
4	zurich , insurance , ceo , senn , martin , former , kill , bos , zurichnanews , group
5	zurich , zurichinsurance , insurance , zurichnanews , guy , zurickinsuk , zurn , read , new
6	zurich , insurance , group , risk , now , job , ltd , zurickinsuk , fi , inc
7	zurich , insurance , company , job , business , analyst , group , ltd , firma , risk
8	zurich , insurance , thank , zurickinsuk , sandy , talk , great , zurichnanews , today
9	zurich , insurance , group , ltd , zurvy , otcqx , international , premier , company , news

Table 4.4: First attempt with LDA , the topics are confused, several words are repeated in several topics. In red are showed words that will be discarded for the future models.

Further pre-processing steps are required. The LDA algorithm is very sensitive to stopwords. The use of standard list of stopwords is not enough, further criteria are needed. An approach used is to remove the most frequent and less frequent words. Removing the most frequent words is a delicate step, in fact you could end by removing words that you

[2] http://www.nltk.org/api/nltk.tokenize.html

[3] http://www.nltk.org/_modules/nltk/stem/wordnet.html

may think are not useful but can be important for the model. For this reason I decide to remove only the words that I use in the query (like *zurich* , *insurance, zurichnanews , ...).* For the less frequent word I decide to remove the words that appear in less than 5 documents.

Furthermore as already discussed in the State of the art chapter, treating each tweet as a separate document could be a problem for LDA because the tweet are very short. Furthermore I decide to aggregate tweets, and the criteria is to group by user. So now each document will be constituted by all the tweet posted by a specific user.

In order to detect the ideal number of topics a validation is performed, with several values of k. The α and β parameters are left to their symmetric default values.

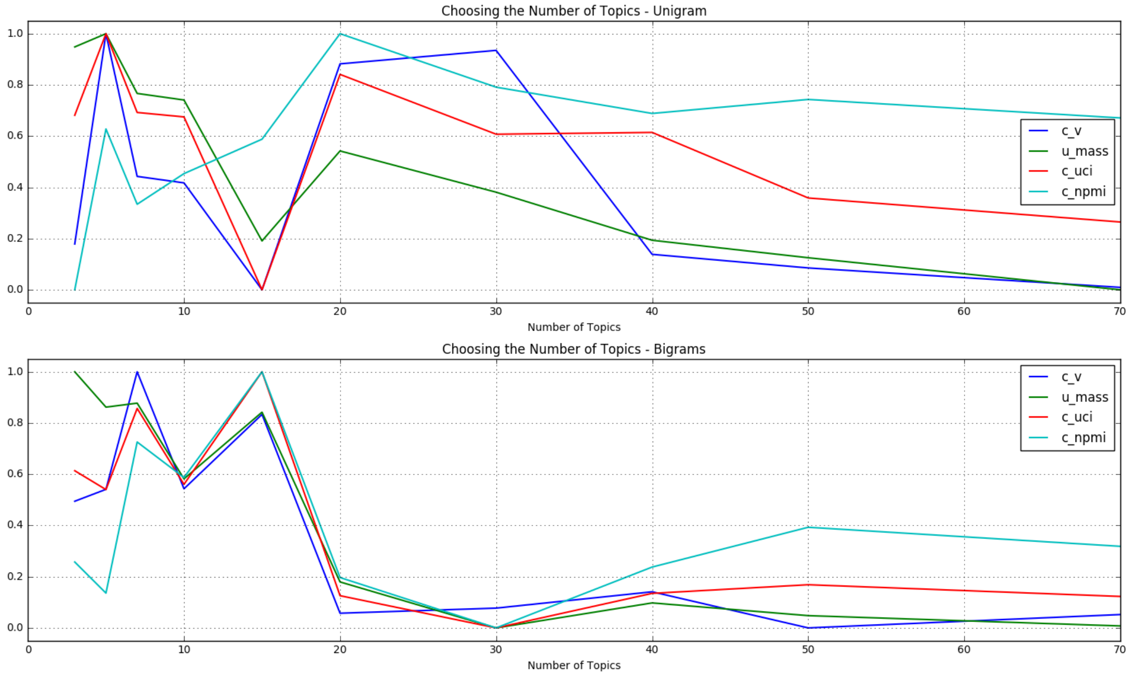


Figure 4.3: Validation for choosing the proper number of topics. The metrics in the plots have been standardized for a fair comparison. It is difficult to choose a best model among the ones with Unigrams. With Bigrams the situation is a little bit more clear. The peaks are at 7 and 15. The best models are inspected: for the Unigram models, the confusion on the metrics it's reflected also on the models, with Bigrams the models are more clear, the best one has shown to be the one with k=15.

Several models have been inspected, and it result that the metrics reflect the quality of the clusters obtained, by the way have not to be taken as the only criteria to select the best model, still a manual inspection among the best ones is needed.

So I choose the model with bigrams and a k=15 to be the candidate. The model is good but I try a further improvement. Another validation is performed. This time the k will be fixed to 15 and I will try to find the best α and η parameters.

4.2.3 Tuning LDA hyperparameters

The metrics from Roder et al. have been tested also for trying to tune the hyperparameters of the LDA. However they have shown to not behave well in this case.

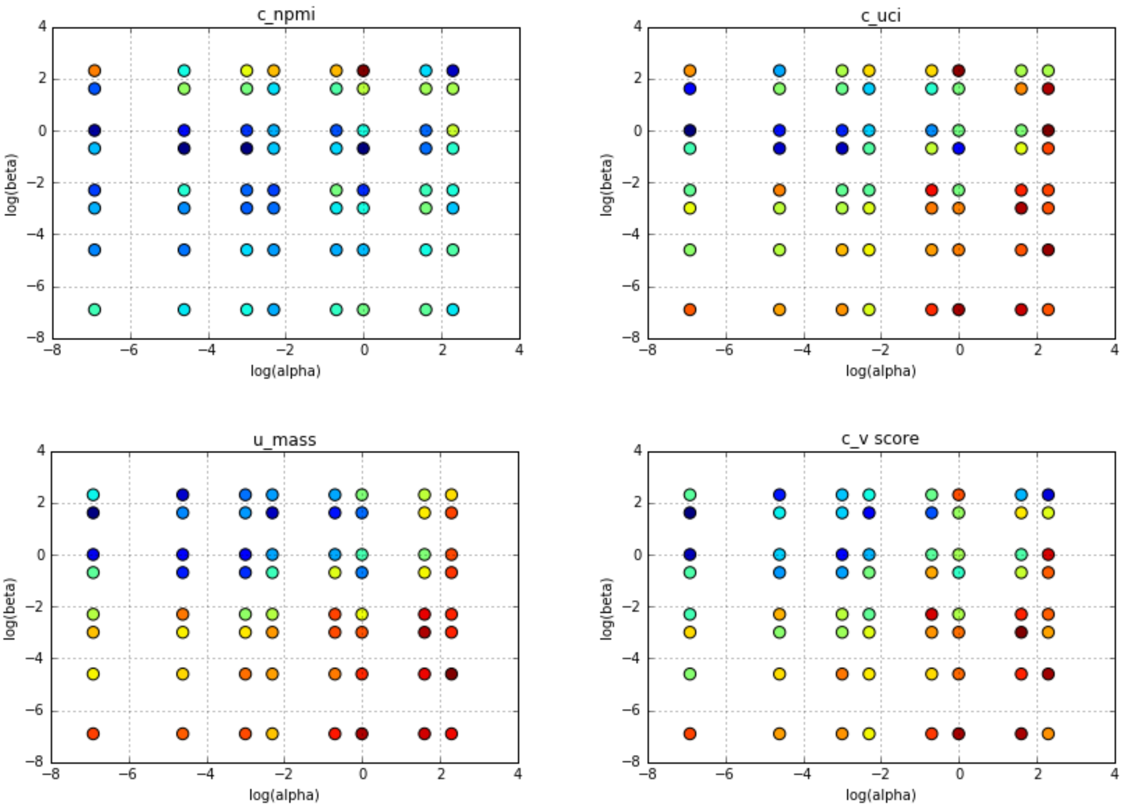


Figure 4.4: Validation for the α and η parameter . The plots show the same 64 models performed: 8 values for α and 8 values for η have been tested. Each plot show a different metrics, cold colors represent low values while the hot ones represent high values. For a better visualization the log scale is used for both axis. Apart from the c_{npmi} all the other metrics show the same pattern.

This validation has shown the limits of these metrics. In fact, while for choosing the proper amount of topics, they were revealed a good estimator, I cannot say the same for the tuning of the α and η parameters.

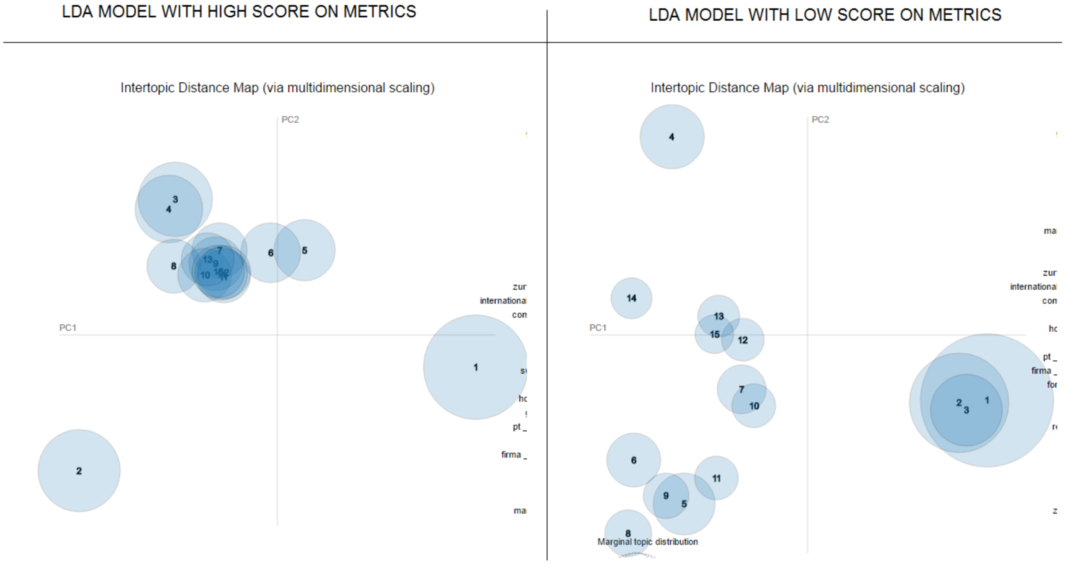


FIGURE 4.5: Comparison of Two LDA Models. The model on the right ($\alpha = 0.001, \beta = 0.01$) has obtained one of the lowest score among all metrics. On the left side one of the best models ($\alpha = 10, \beta = 1$) according to the metrics. The reality is quite the opposite. The second model, since it has high values on α and β produce topics that are very vague and close to each other. The other one is the opposite, with lower parameters it has more spread and clear topics. Thus we cannot rely on these metrics for tuning the model hyper-parameters.

In the literature exists some techniques that are able to estimate the posterior parameters of the LDA [28] . However are not taken in account in this experiment , mostly because a faster solution to the problem has been found.

The key component has been in the stopwords. Removing the tokens that appears in less than 5 document was not enough, I increased this threshold to 10 and I have obtained a much better and interpretable model. This simple fix has shown incredibly great results.

4.2.4 Interpretation of the topics obtained

The interpretation of the topic has been made with the visualization library pyLDAvis. The interpretation of the topics is very straightforward in a dynamic page. Here I write some keywords that belongs to each topic and I provide an interpretation of them.

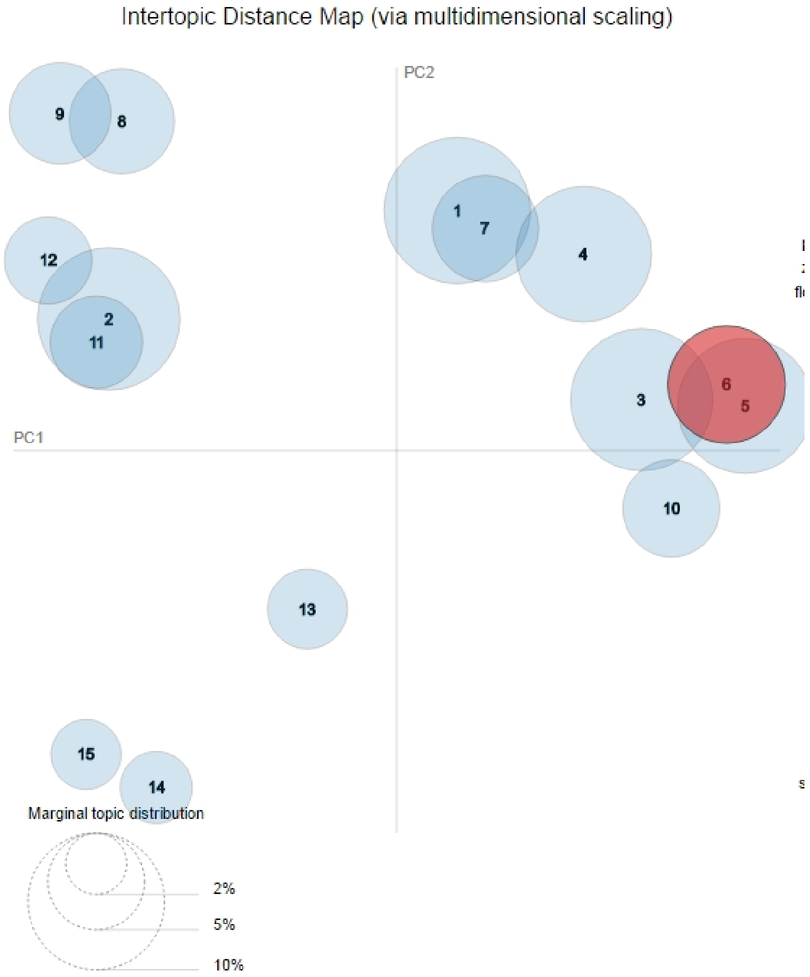


FIGURE 4.6: Final LDA Model visualized.

- **Topic 1: risk , report , business, global , global_risk , cyber , climatechange**

  This topic deal about news, in particular business news that regards the Zurich Ins. Group, the topic are mainly cyber risks , global risks and climate change.

- **Topic 7: read , business, new , follow , risk, case , advice, fraud**

  This topic is close to the 1st , in fact it also deal about news and articles, the area is a little different, here we deal about risks frauds and advices.

- **Topic 4: today , award , event , join , great , conference, speaker**

  This topic is about various events , awards and conferences.

- **Topic 3: claim , get , can , car , company, call , service , policy**

  This is the topic that most of all interests us. There are the opinion , questions, and complaints of the clients about claims, car (insurance), call, email service and so on.

- **Topic 6: great , golf , congrats , win , zurich_classic , flood_resilience**

  Topic 5: thanks , great , team , zctrust, support , volunteer , charity

  These two topics are close, because they share something in common. First of all they both talk about topics that in general have a positive sentiment among the users: the Zurich Classic, a golf tournament sponsored by the Zurich Group and Zurich Community Trust, which are charity initiatives organized by the Zurich Group.

- **Topic 10: pt_indonesia , one , plaza_sentral, pic, floor, go**

  This topic is definitely not clear and it is likely to represent just noise.

- **Topic 8: job , group , company_ltd , finance , manager , senior**

  Topic 9: group , job , bid , job_summary , underwriter , manager

  These two topics are very similar , in fact they both deal about job announces.

- **Topic 11: group_ltd , zurvy , zurvy_otcqx , international_premier**

  Topic 2: group , say, business, group_ag , ceo, profit, reuters

  Topic 12: news , group , data_loss , loss , britain , reuters

  These three topics deals about economic news . Zurvy is the identifier of Zurich in the Stock Market while OTC and OTCQX are stock markets.

- **Topic 13: company, former_ceo , martin_senn , commits_suicide , killed_say**

  Topic 14: story , ex-zurich_boss, suicide , kill , global_corporate , boss_martin

  Topic 15: hong_kong , martin_senn , group_hong , kill , former_boss , ltd

  These three topics deal about a sad episode, the suicide of the CEO of the Company, which has been a big argument in all the media and Social Networks.

4.2.5 Final Filter

For the final filter a binary classifier is designed. The LDA model was designed to explore the data and see what it actually talk about. Once the topic of interest are identified, the tweet of that topics are extracted.

The reason is that Topic Models provide a good description of the data but when is the time to discriminate among relevant and non-relevant, supervised techniques do better the job.

In this case I was interested to the tweet from the customers. I extract them from the topic they belong and I use them as a positive class. For the non-relevant I use tweets from the non-relevant topics plus non relevant tweet from the generic Zurich corpus. (Remember that the main purpose of this filter is to extract relevant tweets from the generic corpus obtained with the "Zurich" query).

Feature Selection	N of features	Lin. Discr. An.	Multin. Naive B.	Log. Regr
stopwords/urls removal	4802	0.998 / 0.666	0.948 / 0.864	0.987 / 0.864
χ^2 test	520	0.917 / 0.817	0.873 / 0.832	0.915 / 0.847
LDA Features	50	0.739 / 0.716	0.536 / 0.508	0.739 / 0.734
Drop tokens with $freq \leq 5$	535	0.928 / 0.794	0.849 / 0.822	0.925 / 0.862

TABLE 4.5: Validation for the Final Relevance-Filter Model. The most basic feature selection is the one who obtain the highest accuracy. However the others feature selections techniques allow to obtain still a very good accuracy but with a much lower number of features and consequently, with a more general model. Very surprising is that by just dropping tokens that appears in less than 5 documents we obtain very good results. This simple feature selection techniques produces the best results. For this reason I choose the logistic regression with the drop tokens feature selections.

Chapter 5

Second Experiment - Sentiment Analysis

There are several ways to perform Sentiment Analysis, each one usually have more than one task. Here are presented three experiments. All of them try to solve the Polarity Detection Task. However in an environment like a Social Network also a Polarity Detection Module is needed.

5.1 Experiment with Sentiment dictionaries

This method represent the baseline for our Sentiment Analysis. Two Sentiment Vocabularies have been tested. One from Bing Liu [1]. It contains 2006 Positive words and 4783 Negative words. The domain of this vocabulary is general.

The other vocabulary is specific on twitter. [2] It is automatically obtained among a big set of tweets and include positive, neutral and negative words.

Dictionary	Method	F1 Score
Bing Liu	count sentiment words	0.587
Twitter Lexicon Waikato	count sentiment words	0.515
Twitter Lexicon Waikato	use sentiment word weights	0.433

Table 5.1: Sentiment Analysis with Dictionaries - Results

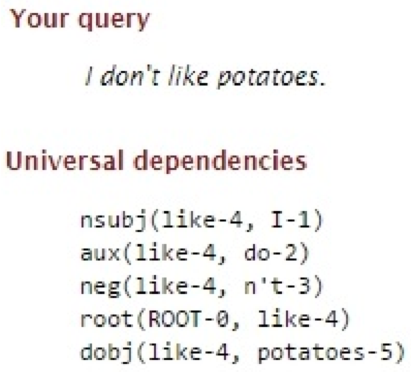
[1] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[2] http://www.cs.waikato.ac.nz/ml/sa/lex.html

5.2 Negation Detection

Negation detection can play an important role in sentiment analysis. For example: imagine to have in your training corpus the phrases ”Bob is good” as a positive instance and ”Bob is bad” as a negative one. Then if you try to classify ”Bob is not good” any classifier is likely to classify it as a positive one. Consequently a negation detection module is used.

The Stanford Dependency parser is chosen. Among the various logical relations that it capture, negation is also included. It is implemented in Java. However the NLTK library of Python include an interface to connect it directly without the need to write a single line in Java.



Your query

I don't like potatoes.

Universal dependencies

```
nsubj(like-4, I-1)
aux(like-4, do-2)
neg(like-4, n't-3)
root(ROOT-0, like-4)
dobj(like-4, potatoes-5)
```

Figure 5.1: An example of a dependency tree obtained with the Stanford Parser. Is visible that the also the negation relation between don’t and like is captured. http://nlp.stanford.edu:8080/parser/

The approach is the following: if the negation is detected, change the negated word to NOT_word . With this transformation a word and its negated counterpart are considered to be two different words, increasing the size of the features, thus a larger dataset is preferable.

5.3 Experiment with manually annotated training

Another path used is to manually label the data and perform Machine Learning Algorithm on it. 1000 tweets have been sampled and manually labeled into positive , negative and neutral ones. Then the 80% is used as a training data and the remaining 20% as test data.

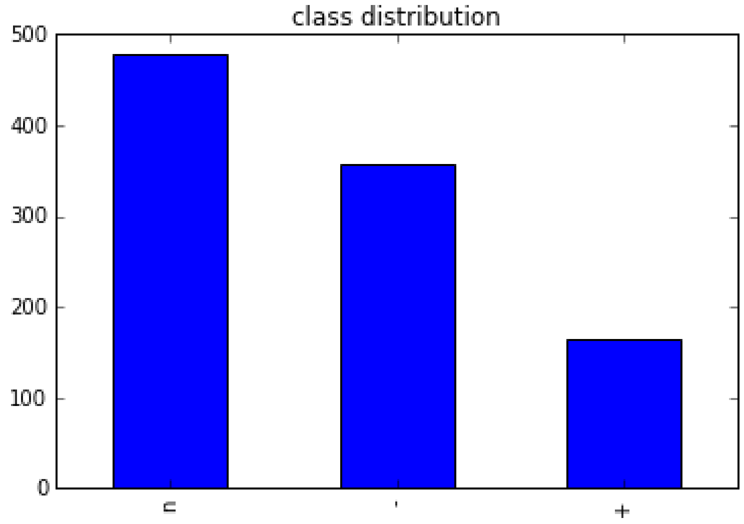


Figure 5.2: Sentiment Analysis - Class Distribution. There is a majority of neutral tweets, followed by the negative ones. The positive is the minor class.

The key here has revealed focusing on feature selection and transformation techniques. In fact complex methods like QDA or Polynomial or RBF kernel has shown very poor results. The reason could be that our Data Space is already very sparse, so complex classification methods are not the key in this case.

Classifier	**F1 Train**	**F1 Test**
LDA	0.78	0.64
QDA	0.49	0.45
linear svm, C=1	0.78	0.68
poly svm , C=100, d=4	0.31	0.31
RBF svm , C=10 , $\gamma = 10$	0.94	0.35

Table 5.2: Sentiment Analysis with manually annotated corpus - polynomial models results. In this case model are performed with unigrams as features, the features are then selected with a chi^2 test. For the svm kernels a 10 fold CV is performed on the training set in order to select the best parameters(the test set is not considered for tuning the models). Is visible that more complex methods doesn't help in reaching better results. For this reason, further approaches will be focus on other aspects rather than tuning complicated models.

For the SVM and logistic regression class weights are used in order to deal for the unbalance of the classes. For the Linear-SVM a 10-fold cross validation is performed on the training set in order to choose the best Cost parameter.

The Feature Selection techniques tested are: Just the basic stopwords removal, then the other two to this add also a χ^2 test , the removal of the tokens that appears in less then 5 documents and Information Gain. As features are used first only unigrams and then also bigrams. The bigrams are calculated in the following way: if two words co-occurs in more than 5 documents are then merged to a single one, if not is left the unigram.

Is tested also the use or not of the negation detection with the Stanford parser. Finally several models are tested: linear Svm , Logistic Regression , Naive Bayes and LDA.

The validation procedure is the following: on the training set is performed a 10-fold cross-validation. The cross-validation procedure is not used only for tuning the models parameters (at least the models who got parameters to tune) but also for the feature selection. In fact on each fold is performed the feature selection only on the training (folds). Then is calculated the average F1 Score for each test-fold.

Here are shown the best methods, the full validation matrix is available in the appendix.

Feature Selection	**Features**	**Neg.**	**Classifier**	**F1 Train 10-CV**
χ^2 test	bigr.	yes	SVM lin. C=1	0.775
Information Gain	bigr.	yes	Naive B.	0.742
Information Gain	bigr.	yes	LDA	0.728

Table 5.3: Sentiment Analysis with manually annotated corpus - best models

The best model is then tested on the test set and has obtained an F1 Score of **0.729** . Belowed is showed the confusion matrix of the best model. Is visible that among the positive and the negative class there are very few mistakes, the most errors are between the negative and the neutral and the neutral and the positive class.

		Predictions		
		-	**N**	**+**
TrueLabel	**-**	44	22	2
	N	16	75	5
	+	2	7	27

Table 5.4: Confusion Matrix of the best model.

5.4 Experiment with Soft Labeling

In the literature the Soft Labeling approach has revealed successful in several case for the binary problem. In this section I will try to extend the soft-label approach to the three class problem.

Query	Description	Size
"product OR service :)"	Positive Class Corpus	891.009
"product OR service :("	Negative Class Corpus	687.663
":)"	Positive Class Corpus	1.571.606
":("	Negative Class Corpus	2.225.751

TABLE 5.5: Queries Performed for the Soft-Labeling approach. The first two queries try to capture a domain that is not too far from our, while the last two are the most generic possible.

The idea is to apply probabilistic classification algorithms in order to obtain the probability of the tweets to be positive or negative. If is neutral, the classifier should be not sure how to deal with it, thus is probability should lay in between the two extremes.

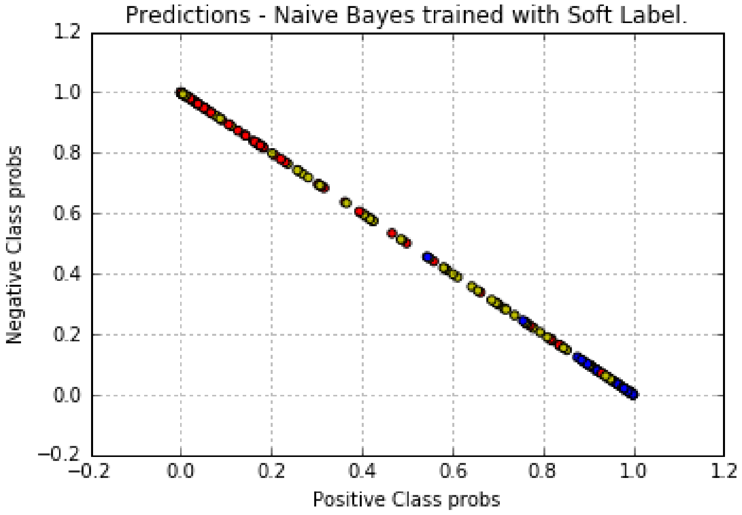


FIGURE 5.3: The intuition behind soft-label for a 3 classes problem. This preliminary model is trained with 20.000 equally balanced tweets. The neutral tweets, in this case, tend to concentrate in the middle.

5.4.1 Computational Limits

Dealing with big dimensional datasets makes some problems. First of all the negation detection approach with the Stanford Parser is very costly. Thus I use a lighter approach, the one described in [22] .

I use a list of negation tokens. The original approach say that if a negation token is found, to all the words that follow it until the next dots is added a prefix "NOT_".

Since I don't think that in the twitter domain the users makes a proper use of the syntax grammars rule. I just replace the next two words. Example: "You aren't very good" become "You aren't NOT_very NOT_good" .

hardly	lack	neither	nor
cannot	daren't	don't	doesn't
didn't	hadn't	shouldn't	hasn't

Table 5.6: Some of the negation tokens for Soft Labeling

Furthermore there is the problem of deal of running Classification Algorithms on very big corpus needs a very big amount of memory. Our Training Matrix will be very sparse. So there are basically two possibilities. Classify using sparse matrix [3] or using algorithms that allows incremental training. [4]

However can be time consuming, and our time is limited. As we are experimenting a totally new approaches, I started first simpler. I perform incremental training size, and for each size I replicate the experiment 20 times, for each replication a different sample is taken from the big corpora. In this way I can calculate confidence intervals and see how it goes with bigger samples.

5.4.2 Pre-processing

The pre-processing is similar to the past approach. What change is that this time I remove the emoticons , the mentions and all the retweets. Also I remove the tweets that contains both happy and sad emoticons. Due to time constraints the unique feature selection technique used is the χ^2 test, the reason is that has shown to be the best in the previous experiment.

5.4.3 Experimental Setup and Results

The approach used is the following: the data is divided in training, validation and test. The training data consist of the two-class, soft-labeled data. The validation set consist of a balanced sample from the training set used in the labeled approach(381 instances). The test set is the same of the other experiments in order to have a fair comparison.

Several models are performed on the training set. After that the validation and test set are predicted, the probability of being positive or negative is taken. After that on the probability space of the validation set is applied a linear SVM (with a 10 fold CV

[3] http://scikit-learn.org/stable/auto_examples/text/document_classification_20newsgroups.html

[4] http://scikit-learn.org/stable/modules/scaling_strategies.html

in order to establish the best C parameter) in order to establish the threshold on the probability space to be positive negative or neutral. Finally the model is used to predict the test set.

As stated in the State of the Art chapter not all models provide a reliable probabilistic estimation. Logistic Regression is in general quite good in this but we cannot say the same for the Naive Bayes Classifier.

For this reason I apply probability calibration for Naive Bayes, because , at least theoretically, it improves the probability estimation.

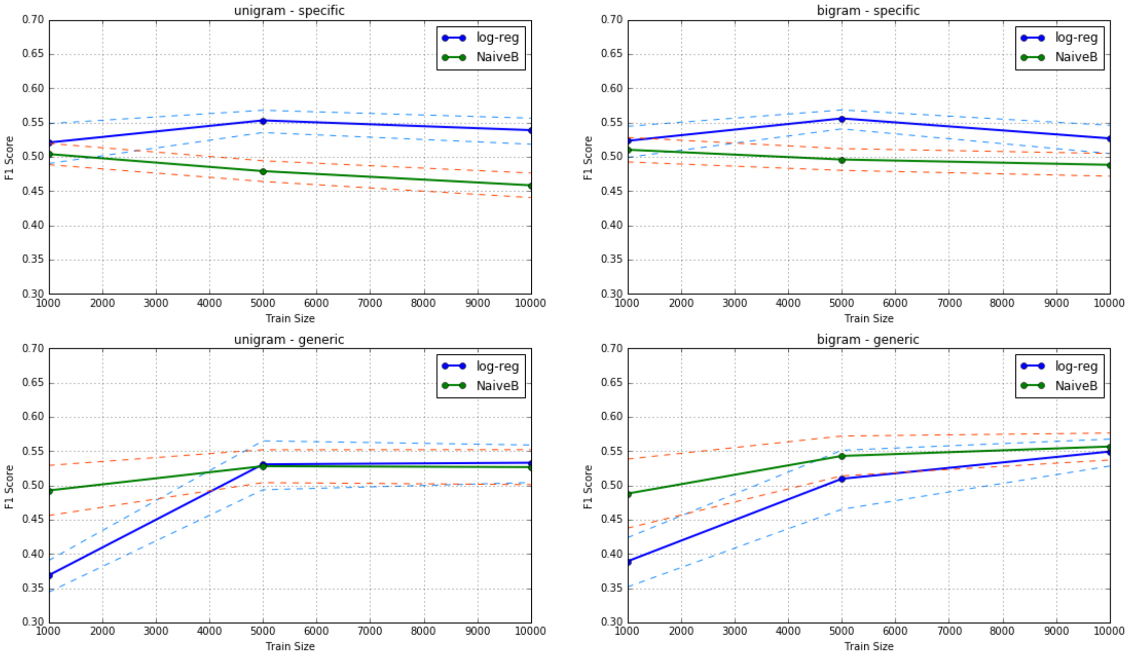


Figure 5.4: Soft Labeling approach - results. Calibration is not applied in this case.

The results are not very satisfactive, in fact under no circumstances we obtain more than 0.57 of F1-Score. With the corpus ”product OR service” there is something wrong because in some cases the performance decreases with the increasing of the training size, while for the generic corpus all the methods with the increasing of the training size tend to converge to the same value.

Not even the calibration of the Naive Bayes has shown great results, in most of the cases the calibration even makes the performance worst.

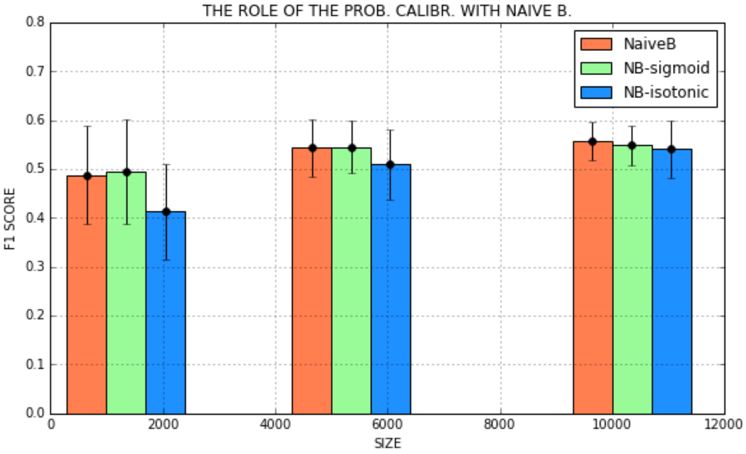


Figure 5.5: Calibration Results. This are the model obtained with the generic corpus. In general the calibration doesn't bring improvements.

Chapter 6

Business Insights and Data Visualization

The focus of this work is on the Data Mining aspect rather than the Business implications. Of course an analysis is not complete without proper conclusion obtained from the data. In this chapter I show some business insights obtained from the data. In the second part I add some insights that creates the basis for future works.

6.1 Customers Opinion

There are 1805 tweets that contain customer opinions and comments.

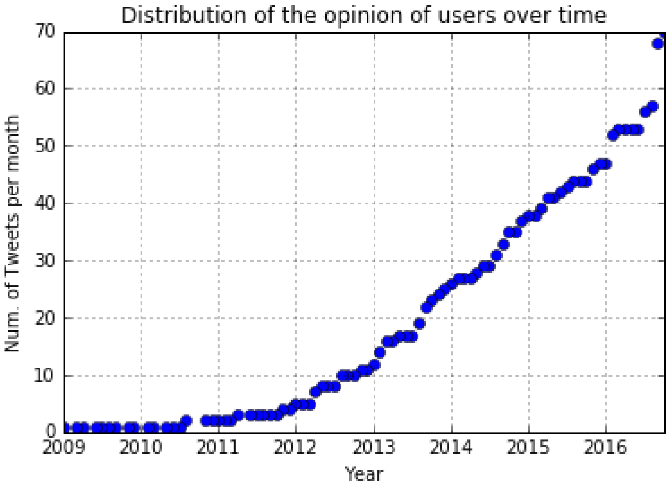


Figure 6.1: Distribution of the customers tweets over time.

The amount of tweets is not very big , however the trend shown in the figure 6.1 suggests that this is going to change.

For the following Pie charts I use the tweets from January 1st 2015 until the most recent (October 31st 2016) in order to have a recent overview. The topics are not mutually exclusive , in fact a tweet can regard a claim about a car accident or when you signs for a car insurance you can talk about the risks as well.

The sentiment in most of the cases is mainly neutral and negative. But there is also to consider the lack of an Opinion Detection module (also called Subjectivity Recognition) . So most of the tweet that are being classified as neutral, whether it is true that they are neutral they do not actually are expressing any opinion about the Zurich Group.

Apart from this, the negative tweets are in general much more than the positive. This aspects alone is not enough to state that the clients are not satisfacted of the service. In fact the customers tend more often to write if they have a problem. But this information is not useless, in fact can be compared with the sentiment in the social of the competitors. The comparison of the sentiment among several companies it will be a good indicator of the clients satisfaction.

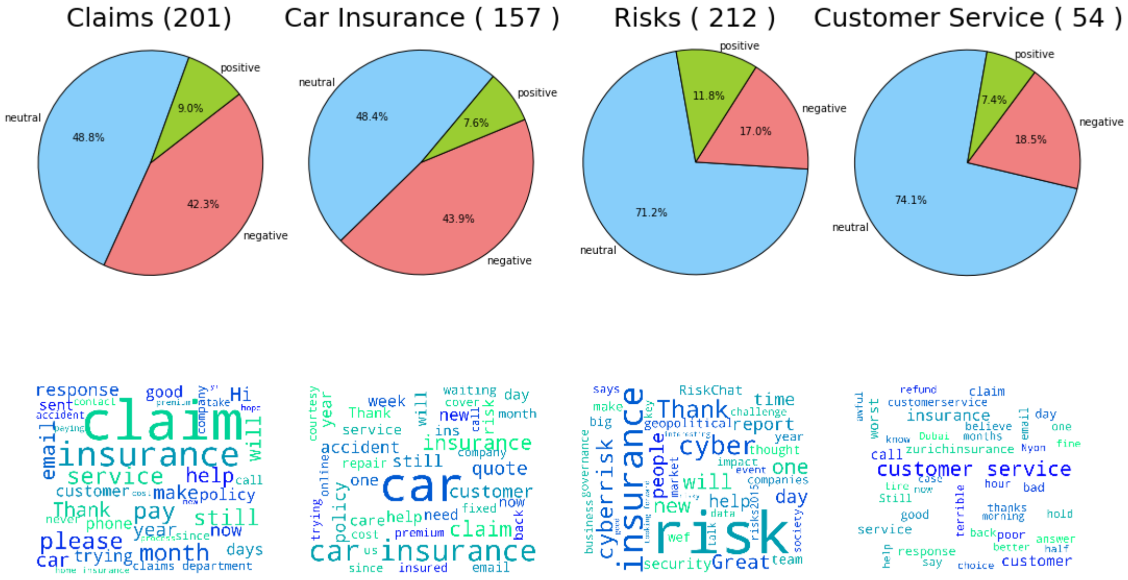


Figure 6.2: Sentiment Pie charts and WordClouds for the period 2015-2016 (In brackets is stated the number of tweets for each category) .

6.2 Comparison with competitors and future works

Let's analyze the amount of traffic. The following plots take into account the amount of tweets that include the name of the Zurich company and other competitors.

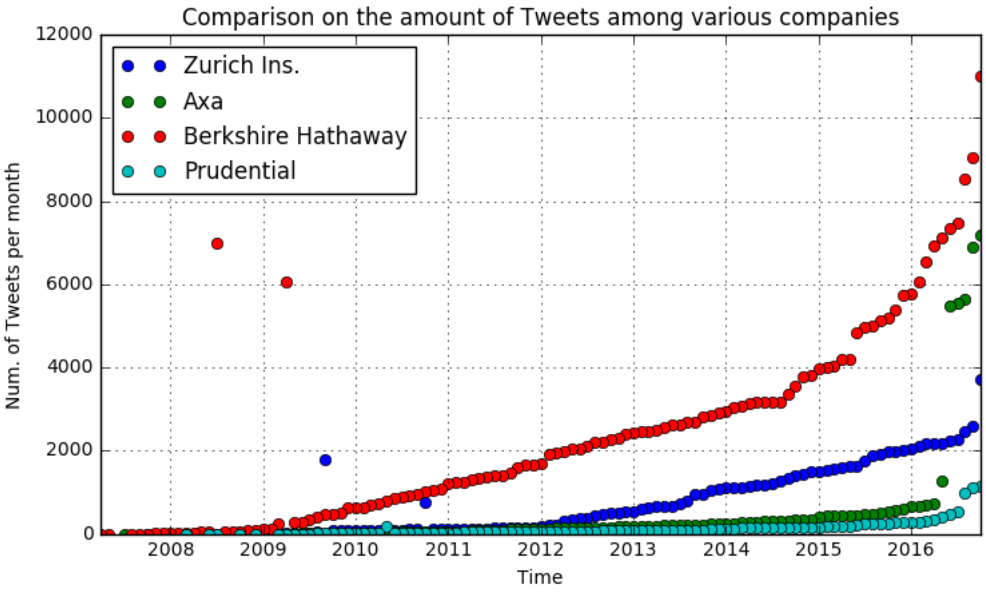


Figure 6.3: Comparison of the presence of different companies on Twitter over time.

Berkshire Hathaway has been the precursor in Twitter since 2009 and in general always dominates , at least by a quantitative point of view. After that there is the Zurich Group, however, in the 2nd half of the 2016 Axa overtook the Zurich Group in Twitter presence. Prudential is the last one in the rank.

Let's take a look at the most frequent authors for each company. For the Zurich Group we have the official pages first followed by job pages and insurance blogs and pages. For the Axa group the majority of tweets are authored by Axa pages. This could mean two things: the first is that the Axa group has a more massive marketing campaign or that there is a greater interaction between the pages of Axa and their customers. For the Berkshire Hathaway group , like the Zurich Group , apart from the activity from the official pages also job and insurance pages that talk about them. For the Prudent Group there is not significant activity from his official pages but are more others than talk about them, but in general less than the other pages.

For future research it could be interesting to analyze the sentiment of the competitors. Having a generally negative sentiment from your clients doesn't necessarily mean that your company is performing poorly. Customers tend to communicate with the company more often if they have problem rather than to thank the company because everything

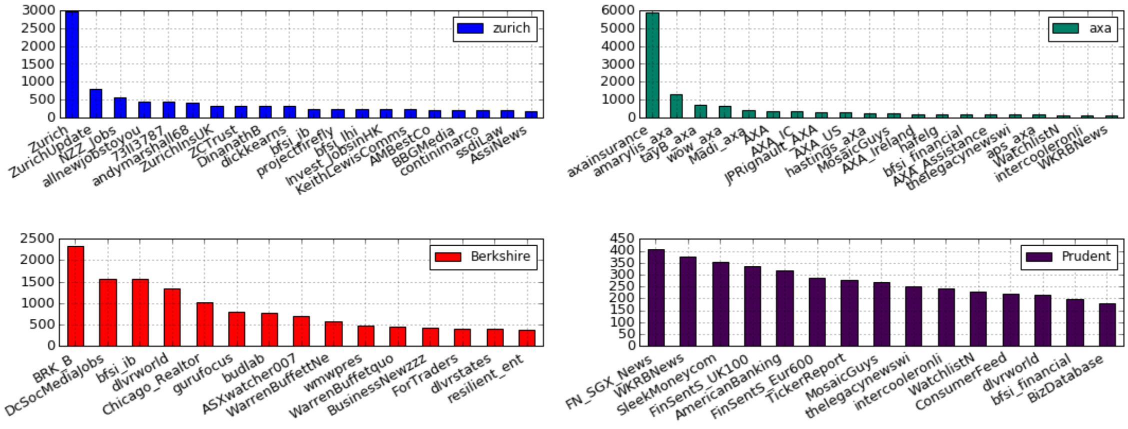


Figure 6.4: Who talk about insurance companies?

is good. But this is true for all the companies so a comparison among them can be a better indicator.

Also it is not only the opinion of the customers that is important. There are also several pages that talk about Insurance Companies, can be interesting also to analyze their opinion as well. Furthermore can be interesting to analyze the interaction (Social Network Analysis [29]) between the customers and the pages (official and non official ones, like the job and news pages) .

Chapter 7

Conclusions

This project has covered several phases of the Data Science process [1] . It started from thinking about how to retrieve the data on the Social Networks . After that, the data has been explored by using state of the art techniques that deal with data expressed in Natural Language. The key here has been the use of the proper metrics to validate the model, pre-processing steps on the data, such as url/mail removal, standardization, and Stopwords removal (not only from standard lists). Also visualization techniques has been fundamental not only for interpreting the model obtained but also for validating it, in fact the metrics alone are not enough to choose the best model, a manual inspection is still needed among the best ones, and the visualization techniques help a lot with this task.

After the data has been explored and filtered, sentiment analysis is performed on it. Here the focus was on the three class problem, so classifying if a tweet is positive, negative or neutral (this problem is more challenging than considering only two class: positive or negative). Several approaches from the literature have been explored and tested. The simpler one that has been used as a baseline for more advanced methods has been the use of dictionaries of sentiment words. This approach have not reached great results. After that a sample of the dataset has been manually labeled and has been divided in training and test. Several approaches have been tested . The key has shown not to be the tuning of complicated models but in the feature selection , extraction and transformation phase. The best model has reached quite satisfactive results similar to state of the art results (of course the results in the literature are not directly comparable because, even if they are also in the twitter domain, the datasets are different). However the performance can still be improved, in particular by adding an Opinion Detection module. This can

[1] http://www.kdnuggets.com/2016/03/data-science-process-rediscovered.html

help to discard the tweet that does not contain an opinion and thus makes the life easier to the Polarity Detection module.

The last approach has been a totally new one. It is inspired from the soft-labeling approach in the literature. In order to avoid the manual work of labeling the tweets, they are automatically labeled by using the tweets that include (happy or sad) emoticons. This approach has shown good results in the literature even with a general training set (a big enough general domain training set is able to reach decent results in a specific domain). The soft-labeling approach has been shown to work for the two class problem, I tried to extend it to the three class problem. Unfortunately the results have shown that this approach is not very effective for the three class problem. However finding which ones do not work is an important step in the process of finding the most effective one.

7.1 Future Works

Unfortunately, every idea has not made it into the final work. Future works will be focused in the following directions:

- For the Latent Dirichlet Allocation the metrics used in the experiment have proven effective for estimating the number of Topics (K) but not good for estimating the other hyperparameters (α and η) . For the final model, I left α and η to their default values and I have reached a good model, however it would have been interesting to explore the approaches present in the literature [28] in order to tune them.
- Attempting to improve the performance of the Sentiment Polarity Detection. In particular an Opinion Detection module [30] would help to increase the performance. In fact, at the moment in the neutral class are present also tweets that do not have an opinion and this makes the life of the classifier more challenging. Detecting the non-opinionated tweet in advance and discarding them can help improve the performance and to have more reliable results.
- As already stated in chapter 6 there is still a lot to investigate. Further research can include Social Network Analysis, applied both to Zurich and to its competitors. It is important to also extend also the Sentiment Analysis to the competitors in such a way to have a better interpretability of the results obtained.

Appendix A

Sentiment Analysis - Validation Results

Here there is the full matrix of the validation of the models with the manually labelled training samples.

Feature Selection	**Features**	**Neg.**	**Classifier**	**F1 Train 10-CV**
stopwords/urls removal	uni.	no	Naive-B	0.612
stopwords/urls removal	uni.	no	LDA	0.399
stopwords/urls removal	uni.	no	Log. Regr.	0.636
stopwords/urls removal	uni.	no	SVM lin. c=0.1	0.614
drop tokens with $freq \leq 5$	uni.	no	Naive-B	0.648
drop tokens with $freq \leq 5$	uni.	no	LDA	0.495
drop tokens with $freq \leq 5$	uni.	no	Log. Regr.	0.668
drop tokens with $freq \leq 5$	uni.	no	SVM lin. c=0.1	0.643
χ^2 test	uni.	no	Naive-B	0.693
χ^2 test	uni.	no	LDA	0.694
χ^2 test	uni.	no	Log. Regr.	0.676
χ^2 test	uni.	no	SVM lin. c=1	0.629
Information Gain	uni.	no	Naive-B	0.667
Information Gain	uni.	no	LDA	0.681
Information Gain	uni.	no	Log. Regr.	0.662
Information Gain	uni.	no	SVM lin. c=1	0.683
stopwords/urls removal	bigr.	no	Naive-B	0.610
stopwords/urls removal	bigr.	no	LDA	0.456
stopwords/urls removal	bigr.	no	Log. Regr.	0.638
stopwords/urls removal	bigr.	no	SVM lin. c=0.01	0.643
drop tokens with $freq \leq 5$	bigr.	no	Naive-B	0.674
drop tokens with $freq \leq 5$	bigr.	no	LDA	0.495

Feature Selection	**Features**	**Neg.**	**Classifier**	**F1 Train 10-CV**
drop tokens with $freq \leq 5$	bigr.	no	Log. Regr.	0.671
drop tokens with $freq \leq 5$	bigr.	no	SVM lin. c=1	0.691
χ^2 test	bigr.	no	Naive-B	0.697
χ^2 test	bigr.	no	LDA	0.650
χ^2 test	bigr.	no	Log. Regr.	0.677
χ^2 test	bigr.	no	SVM lin. c=1	0.693
Information Gain	bigr.	no	Naive-B	0.694
Information Gain	bigr.	no	LDA	0.695
Information Gain	bigr.	no	Log. Regr.	0.652
Information Gain	bigr.	no	SVM	0.645
stopwords/urls removal	uni.	yes	Naive-B	0.672
stopwords/urls removal	uni.	yes	LDA	0.413
stopwords/urls removal	uni.	yes	Log. Regr.	0.695
stopwords/urls removal	uni.	yes	SVM lin. c=0.1	0.678
drop tokens with $freq \leq 5$	uni.	yes	Naive-B	0.687
drop tokens with $freq \leq 5$	uni.	yes	LDA	0.605
drop tokens with $freq \leq 5$	uni.	yes	Log. Regr.	0.686
drop tokens with $freq \leq 5$	uni.	yes	SVM lin. c=1	0.669
χ^2 test	uni.	yes	Naive-B	0.703
χ^2 test	uni.	yes	LDA	0.696
χ^2 test	uni.	yes	Log. Regr.	0.699
χ^2 test	uni.	yes	SVM lin. c=1	0.692
Information Gain	uni.	yes	Naive-B	0.727
Information Gain	uni.	yes	LDA	0.732
Information Gain	uni.	yes	Log. Regr.	0.692
Information Gain	uni.	yes	SVM c=1	0.721
stopwords/urls removal	bigr.	yes	Naive-B	0.675
stopwords/urls removal	bigr.	yes	LDA	0.451
stopwords/urls removal	bigr.	yes	Log. Regr.	0.695
stopwords/urls removal	bigr.	yes	SVM lin. c=0.1	0.686
drop tokens with $freq \leq 5$	bigr.	yes	Naive-B	0.675
drop tokens with $freq \leq 5$	bigr.	yes	LDA	0.580
drop tokens with $freq \leq 5$	bigr.	yes	Log. Regr.	0.685
drop tokens with $freq \leq 5$	bigr.	yes	SVM lin. c=0.1	0.666
χ^2 test	bigr.	yes	Naive-B	0.717
χ^2 test	bigr.	yes	LDA	0.692
χ^2 test	bigr.	yes	Log. Regr.	0.696
χ^2 test	bigr.	yes	SVM lin. c=1	0.745
Information Gain	bigr.	yes	Naive-B	0.742
Information Gain	bigr.	yes	LDA	0.728
Information Gain	bigr.	yes	Log. Regr.	0.716

Feature Selection	**Features**	**Neg.**	**Classifier**	**F1 Train 10-CV**
Information Gain	bigr.	yes	SVM lin. c=1	0.710

Table A.1: Sentiment Analysis with manually annotated corpus - results

Appendix B

Sentiment Analysis with Soft Labeling - Full Results

Below there are the complete results for the Soft Labeling approach. Each configuration has been replicated 20 times. The replications are then used to calculate the average F1 Score on the test set and the confidence intervals.

Size	Method	$F1_{test}$	95% C.I. -	95% C.I.+
1000	log-reg	0.519	0.490	0.549
1000	NaiveB	0.504	0.489	0.520
1000	NaiveB-sigmoid	0.503	0.484	0.523
1000	NaiveB-isotonic	0.478	0.455	0.501
5000	log-reg	0.552	0.536	0.568
5000	NaiveB	0.479	0.464	0.494
5000	NaiveB-sigmoid	0.468	0.455	0.481
5000	NaiveB-isotonic	0.474	0.462	0.486
10000	log-reg	0.538	0.519	0.557
10000	NaiveB	0.459	0.441	0.476
10000	NaiveB-sigmoid	0.437	0.422	0.453
10000	NaiveB-isotonic	0.462	0.447	0.477

Table B.1: Model performed on the corpus "product OR service". The models use only unigrams

Size	Method	$F1_{test}$	95% C.I. -	95% C.I.+
1000	log-reg	0.522	0.499	0.545
1000	NaiveB	0.510	0.493	0.528
1000	NaiveB-sigmoid	0.503	0.483	0.524
1000	NaiveB-isotonic	0.498	0.474	0.522
5000	log-reg	0.555	0.541	0.569
5000	NaiveB	0.496	0.480	0.512
5000	NaiveB-sigmoid	0.479	0.460	0.497
5000	NaiveB-isotonic	0.487	0.474	0.501
10000	log-reg	0.526	0.505	0.546
10000	NaiveB	0.488	0.472	0.505
10000	NaiveB-sigmoid	0.465	0.449	0.481
10000	NaiveB-isotonic	0.482	0.470	0.493

Table B.2: Model performed on the corpus ”product OR service”. The models use also bigrams

Size	Method	$F1_{test}$	95% C.I. -	95% C.I.+
1000	log-reg	0.367	0.344	0.390
1000	NaiveB	0.493	0.456	0.529
1000	NaiveB-sigmoid	0.506	0.473	0.540
1000	NaiveB-isotonic	0.394	0.367	0.420
5000	log-reg	0.529	0.494	0.565
5000	NaiveB	0.528	0.504	0.552
5000	NaiveB-sigmoid	0.526	0.502	0.549
5000	NaiveB-isotonic	0.520	0.493	0.546
10000	log-reg	0.532	0.504	0.559
10000	NaiveB	0.527	0.501	0.552
10000	NaiveB-sigmoid	0.510	0.485	0.535
10000	NaiveB-isotonic	0.514	0.487	0.541

Table B.3: Model performed on the generic corpus. The models use only Unigrams

Size	Method	$F1_{test}$	95% C.I. -	95% C.I.+
1000	log-reg	0.388	0.352	0.424
1000	NaiveB	0.488	0.438	0.538
1000	NaiveB-sigmoid	0.495	0.441	0.548
1000	NaiveB-isotonic	0.413	0.364	0.462
5000	log-reg	0.508	0.465	0.551
5000	NaiveB	0.543	0.514	0.572
5000	NaiveB-sigmoid	0.546	0.519	0.573
5000	NaiveB-isotonic	0.510	0.474	0.545
10000	log-reg	0.548	0.528	0.568
10000	NaiveB	0.557	0.537	0.577
10000	NaiveB-sigmoid	0.548	0.528	0.569
10000	NaiveB-isotonic	0.541	0.511	0.570

Table B.4: Models performed on the generic corpus - bigrams

Bibliography

[1] L. Arbuckle K. El Emam. *Anonymizing Health Data*. O'Reilly Media, 2013. ISBN 9781449363079.

[2] Ryan Mitchell. *Web Scraping with Python*. O'Reilly Media, June 2015. ISBN 9781491910290.

[3] P. Meibner S. Munzert, C. Rubba and D. Nyhuis. *Automated Data Collection With R*. John Wiley & Sons, 2015.

[4] Dunson D. Blei D., Carin L. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27:55–65, November 2010.

[5] Megan R. Brett. Topic modeling: A basic introduction. *Journal of Digital Humanities*, 2, 2012. http://journalofdigitalhumanities.org/2-1/topic-modeling-a-basic-introduction-by-megan-r-brett/.

[6] S.Buntine W.Xie L Mehrotra, R. Sanner. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] Fred Damerau Nitin Indurkhya. *Handbook of Natural Language Processing, second edition*. Chapman & Hall CRC, 2010. ISBN 1420085921.

[8] Christopher Manning and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999. ISBN 9780262133609.

[9] Joakim Nivre. Dependency grammar and dependency parsing. In *Introduction to Data-Driven Dependency Parsing*. European Summer School in Logic Language and Information 2007, 2007.

[10] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944968.

[11] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3. URL http://dl.acm.org/citation.cfm?id=645526.657137.

[12] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 889–892, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484166. URL http://doi.acm.org/10.1145/2484028.2484166.

[13] S. T. Dumais D. Ramage and D. J. Liebling. Characterizing microblogs with topic models. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, 2010. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1528.

[14] Jordan Boyd-Graber, Jonathan Chang, Sean Gerrish, Chong Wang, and David Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems (NIPS)*, 2009.

[15] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eight International Conference on Web Search and Data Mining, Shanghai, February 2-6*, 2015. URL http://svn.aksw.org/papers/2015/WSDM_Topic_Evaluation/public.pdf.

[16] DARIA-DEH. Visualizing topic models. URL https://de.dariah.eu/tatom/topic_model_visualization.html.

[17] Ben Mabey. Visualizing topic models. Data Science Summit & Dato Conference, 2015. URL https://www.youtube.com/watch?v=tGxW2BzC_DU.

[18] V. Pandey and C.V.Krishnakumar Iyer. Sentiment analysis of microblogs.

[19] J.K. Ahkter and S. Soria. Sentiment analysis: Facebook status messages.

[20] M. Polignano C. Musto, G. Semeraro. A comparison of lexicon-based approaches for sentiment analysis of microblog posts.

[21] L. Gatti, M. Guerini, and M. Turchi. Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, PP(99):1–1, 2015. ISSN 1949-3045. doi: 10.1109/TAFFC.2015.2476456.

[22] Olga Kolchyna, Thársis T. P. Souza, Philip C. Treleaven, and Tomaso Aste. Twitter sentiment analysis. *CoRR*, abs/1507.00955, 2015. URL http://arxiv.org/abs/1507.00955.

[23] E. Frank F. B.-Marquez and B. Pfahringer. Positive, negative, or neutral: Learning an expanded opinion lexicon from emoticon-annotated tweets. *Knowl.-Based Syst.*, 108:65–78, 2016.

[24] R. Bhayani A. Go and L. Huang. Twitter sentiment classification using distant supervision, 2009.

[25] M. Marchetti-Bowick and N. Chambers. Learning for microblogs with distant supervision: Political forecasting with twitter.

[26] Ramon Xuriguera. Using twitter as a source of information for time series prediction. 2012.

[27] Jan Hendrik Metzen. Probability calibration. 2015. URL https://jmetzen.github.io/2015-04-14/calibration.html.

[28] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press. ISBN 978-0-9749039-5-8. URL http://dl.acm.org/citation.cfm?id=1795114.1795118.

[29] Reda Alhajj and Jon Rokne. *Encyclopedia of Social Network Analysis and Mining*. Springer Publishing Company, Incorporated, 2014. ISBN 1461461715, 9781461461715.

[30] Amitava Das and Sivaji Bandyopadhyay. Theme detection an exploration of opinion subjectivity. In *ACII*, pages 1–6. IEEE Computer Society, 2009.