Sentiment Analysis and Topic Modelling on Brand-Related Social Media Data

Maria Beatriz Ribeiro

Dissertação de Mestrado

Orientador na FEUP: Prof. Ana Cristina Costa Aguiar

Orientador na NOS: Carlos Miguel Silva Couto Pereira

Mestrado Integrado em Engenharia e Gestão Industrial

2020-06-29

À minha família, que tudo faz para me ver feliz.

Resumo

As opiniões são tão centrais à natureza humana que a importância de as compreender, seja num contexto pessoal ou comercial, não é uma surpresa. Desde *posts* no Twitter sobre um produto a *threads* em fóruns sobre atendimento ao cliente, as empresas desejam ter acesso a estes dados em tempo real. É neste ponto que entram as ferramentas de *Social Listening*. Na grande globalidade integrando ferramentas de *scraping* e análise de sentimento, o vasto mundo de benefícios que estas ferramentas oferecem reflete-se nos objetivos e na estrutura desta dissertação. Três objetivos principais foram definidos de forma a potenciar a construção de uma plataforma que motive as empresas a agir conformemente, perante a apresentação destes dados de sentimento valiosos sobre os seus clientes: (1) Criação de uma ferramenta de análise de sentimento escalável, capaz de classificar como negativos, neutros ou positivos comentários extraídos das mídias sociais relacionados com a marca. (2) Criação de uma ferramenta de modelação de tópicos que complemente esta análise, vinculando sentimentos a um alvo e incentivando a empresa a agir em conformidade com o aconselhado. (3) Extração de informações valiosas acerca dos resultados obtidos. Esta dissertação foi projetada para se ajustar à metodologia CRISP-DM.

Ao longo do desenvolvimento desta história, o leitor tomará conhecimento dos conceitos mais importantes relativos à Análise de Sentimento e juntar-se-á ao autor numa jornada que descreve as etapas cronológicas necessárias ao desenvolvimento da ferramenta. Os modelos de classificação ocupam o centro do palco nesta análise. Logistic Regression, Naïve Bayes, Decision Trees, Support Vector Machines, Random Forest and Multi-layer Perceptron foram implementados para prever o sentimento dos comentários nas mídias sociais. Dois modelos de modelação de tópicos foram ainda implementados: LDA e GSDMM. Os resultados fornecidos por estas metodologias são analisados e *insights* importantes são extraídos por meio de ferramentas de visualização. Com base na AUC-ROC, o melhor desempenho de classificação (74.84%) foi alcançado quando aplicado o modelo MLP com os seguintes parâmetros {hidden_layer_sizes: (100,), activation: relu, alpha: 0.05} num set de teste lematizado e desprovido de emojis. Este modelo é posteriormente comparado com uma ferramenta externa de análise de sentimento. O primeiro atingiu uma AUC-ROC de 65.25% e o segundo atingiu uma AUC-ROC de 66.77%. Quando comparando ferramentas desenvolvidas interna e externamente, outros valores devem ser tidos em causa, já que ferramentas internas proporcionam uma maior flexibilidade e transparência. Em relação à modelação de tópicos, tanto na *accuracy* (45.89%) quanto na *UMass topic coherence* (-2,62), o modelo GSDMM foi superior.

Sentiment Analysis and Topic Modelling on Brand-Related Social Media Data

Abstract

Opinions are so central to human nature that the importance of studying them, whether in a personal or business setting, is no surprise. From tweet posts about a product to forum threads about customer care, companies desire to access this data in real time. In come social listening tools. Typically integrating scraping and sentiment analysis, the vast world of benefits these tools provide has been reflected into this article's goals and structure. Three intents were set in order to create a platform which motivates companies to act according to sentiment data regarding its customers: (1) Create a scalable sentiment analysis tool capable of classifying brand-related comments, extracted from Social Media, as negative, neutral or positive. (2) Create a topic modelling tool that complements this analysis, by binding sentiment to a target, and encourages the company to make changes where necessary. (3) Provide valuable insights on the results acquired. This dissertation was designed to fit the CRISP-DM methodology.

Throughout the development of this story, the reader will be introduced to the most important Sentiment Analysis ideas and will join the author on an implementation journey that describes the chronological steps required in the tool development stage. The Classification models take centre stage in this analysis. Logistic Regression, Naïve Bayes, Decision Trees, Support Vector Machines, Random Forest and Multi-layer Perceptron were implemented in order to predict the sentiment of Social Media comments. Topic Modelling allows to associate a target with the predicted sentiment and enables companies to know when and where to take action. Two models were implemented to achieve this purpose: Latent Dirichlet Allocation (LDA) and Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM). Results provided by these methodologies are analysed and important insights are extracted through visualization tools. According to computed AUC-ROC, the best sentiment classification performance (74.84%) was achieved by when applying the MLP model with the following optimal parameters - {hidden_layer_sizes: (100,), activation: relu, alpha: 0.05} – on a lemmatized test set stripped of emojis. This model is, later on, put head to head with an external sentiment classification tool, the first achieving an AUC-ROC of 65.25% while the latter achieved an AUC-ROC of 66.77%. One must note that, an in-house tool can also provide multiple benefits to a company such as increase flexibility and transparency. These benefits must also be taken into account when deciding between internal and third-party tools. Regarding topic modelling, in both accuracy (45.89%) and UMass topic coherence (-2,62), the GSDMM model was superior.

Acknowledgements

"Kindness is a language which the deaf can hear and the blind can see."

Mark Twain

To my mentors, Carlos Pereira and Ana Aguiar, thank you. For shinning a light on a subject which was, at times, dark and unknown. For the technical support and kind words. For the time spent clearing my doubts. For introducing me to unknown territory and forcing me to find tools and motivation to always aim higher.

I cannot express enough gratitude towards NOS as a company and, above all, as a family, for making me feel so welcome and at ease to share a laugh or a concern. And for offering me the opportunity to learn with those who have so much to offer.

Shall I find this opportunity to thank everyone involved in my education. To my teachers and professors that, with every word, managed to enlighten me on the trivial, complex, amusing and, even, boring matters of the world.

To my family, thank you for being the reason I smile and stand here today, proud of what I have accomplished. Nothing more needs to be said.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Project Background and Motivation

"Most people are other people. Their thoughts are someone else's opinions, their lives a mimicry, their passions a quotation." - ***Oscar Wilde, De Profundis***

There is more than one story to tell. The world is a place where different perspectives can coexist, converge and diverge. Nowhere in this world is there someone who doesn't opine. In fact, opinions are a mark of human thought and reasoning. They are a form of communication and calibration to the people around us. Ideally, opinions are formed as the product of individual thought, with influence coming from experience, research, and interaction with others. They are so central to human nature that the importance of understanding other's opinions, whether in a personal or business setting, does not come as a surprise.

The search for opinions and sentiment

We usually search for external input when making decisions, whether it is in an individual setting, looking for reviews on a product, or in a business setting, searching for consumer feedback. To do so, for long, asking personal acquaintances or conducting surveys and focus groups was considered standard.

"But fragmenting media and changing consumer behaviour have crippled traditional monitoring methods. Tactics [of the traditional sort] such as clipping services, field agents, and ad hoc research simply can't keep pace." (Kim, 2006)

Recent times have dictated a paradigm shift in opinion acquisition methods, with the radical growth of social media platforms and usage. Using social media contents to aid decision-making has surpassed convenience and has become a necessity. Indeed, "the increased exposure of the average citizen and customer to polarised content from various sources has been of significant consequence for companies and governmental organisations" (Kazmaier and van Vuuren, 2020).There is a need for companies to pay more attention to these opinions and use them as valuable input to their marketing efforts, through social media monitoring.

"As major companies are increasingly coming to realize, these consumer voices can wield enormous influence in shaping the opinions of other consumers — and, ultimately, their brand loyalties, their purchase decisions, and their own brand advocacy... Companies can respond to the consumer insights they generate through social media monitoring and analysis by modifying their marketing messages, brand positioning, product development, and other activities accordingly" (Zabin and Jefferies, 2008)

However, opening a door to new possibilities also opens a door to new challenges which must be overcome in order to take proper advantage of the benefits of these methods. The high

number of websites and its diversity in types and contents makes it notably difficult for the average user to extract relevant information. (Horrigan, 2008) states that 58% of American users report that online information was missing, impossible to find, confusing, and/or overwhelming.

Due to the vast nature of such data, manual approaches to this problem are no longer feasible (Kazmaier and van Vuuren, 2020). Thus, with the growth of social media and review platforms, born were the automated sentiment analysis systems.

1.2 The Project within the Company

1.2.1 NOS and the Telecommunications Sector

NOS is one of the biggest players in the Portuguese telecommunications and entertainment sectors. Alongside Altice, Vodafone and Nowo, together they share this important market. According to (Oliveira, 2019), Portuguese telecommunication companies have been increasingly closing the gap between each other in terms of the service and price provided, and, therefore, have started to resort to competitive advantages appertaining to quality of the product as well as effectiveness of the customer relationship departments. This has led to the realization that customer understanding and satisfaction are of utmost importance in this sector. Brands must take advantage of Business Intelligence tools in order to attend to their customer's wants and needs.

1.2.2 Social Listening - the platform to a successful marketing strategy

From tweet posts about a product and Reddit threads about customer care, to competitors announcing price drops on Facebook, within the span of a second, millions of consumers are talking about brands on social media. Companies desire to access this data in real time. In come social listening tools.

Social listening is the process of tracking conversations and trends, through mentions of certain words, phrases, or complex queries commonly related to a brand or industry, across social media and the web, followed by an analysis of the data to aid marketing decision-making. (*Social listening: what it is, why it matters, and how to do it*, no date)

Social listening can also, and sometimes wrongly, be referred to as: buzz analysis, social media measurement, brand monitoring, social media intelligence and social media monitoring (*Social listening: what it is, why it matters, and how to do it*, no date). Although these terms have been used interchangeably, there are fundamental differences between Social Listening and Social Monitoring (*Social Media Listening: What You Need to Know to Get Started*, no date):

Social monitoring: Caring for your customers by monitoring social media for messages directly related to your brand and responding to those messages appropriately.

Social listening: Understanding your audience and improving campaign strategy by accessing the full spectrum of conversation around your industry, brand, and any topics relevant to your brand.

In essence, monitoring addresses the symptoms and listening reveals the root cause (*Social Media Listening: What You Need to Know to Get Started*, no date).

1.2.3 Social Listening is essential on a Brand's Social Strategy Toolkit

Taking Consumer Generated Media seriously via active social listening is critical for all B2C companies that consider themselves socially devoted. By doing so, businesses can answer important customer, market, and competition-related questions without having to ask the actual questions (*Social listening: what it is, why it matters, and how to do it*, no date). This tool is pertinent to a multitude of departments and applications, such as the following:

1. **Reputation Management**: Businesses can monitor mentions of their brand and products to track brand health and react to changes in volume of mentions and sentiment early, to prevent reputation crises.
2. **Performance Measurement**: Social listening also grants access to quantitative metrics (e.g. Volume of Conversation) and qualitative metrics (e.g. Sentiment of Conversation), which provides valuable insights to determine your social performance for Above the Line (ATL), Below the Line (BTL) and integrated campaigns (*7 Reasons Why Social Listening Is Important*, no date)
3. **Competitor Analysis**: Aid every step of competitor analysis: measuring share of voice and brand health metrics; benchmarking; learning about their customer's opinions; discovering influencers and publishers they partner with.
4. **Product Feedback and Messaging Strategy**: Monitoring sentiment on products serves as an important lesson on how customers react to product changes and what they love and believe is missing from them. Moreover, by monitoring their opinions, brands gain a profound understanding of their audience's needs in order to enhance their messaging and social media strategy. Focused listening will guide brands on their social network choices in such a way that maximizes their reach.
5. **Customer Service**: By only monitoring mentions which include their handles, brands are missing on 70% of the conversations about their business. Knowing that 68% of customers leave a company because of its unhelpful customer service, ignoring those 70% has quite expensive consequences.
6. **Public Relations**: Social listening lets you monitor when press releases and articles mentioning your company get published and allows to track mentions of competitors and industry keywords across the online media to find new platforms to get coverage on and journalists to partner with.
7. **Influencer Marketing**: Calculate the impact, or reach, of a brand's mentions and search for the most influential people in a specific niche to create valuable influencer partnerships and improve Word of Mouth.
8. **Research**: Social listening also permits monitoring sentiment on any phenomenon online that might be relevant to the company.

In conclusion, Social Listening tools helps brands understand why, where and how conversations about them are happening, and what people think, so that they can improve their promotion strategy, outpace competition and build better relationships with partners and customers (*Social Media Listening: What You Need to Know to Get Started*, no date).

1.2.4 In-house or third-party software development – the eternal debate

Although brands can manually perform basic monitoring tasks, an extensive social listening strategy requires a comprehensive in-house or third-party tool to analyse large volumes of data and provide structured and fruitful information. In other words: "While you can look at trees one by one at the ground level, you need a helicopter to scan the whole forest" (*Social*

Media Monitoring vs. Social Media Listening, no date). Ready-made automated listening tools can offer actionable data as meaningful as other customizable tools. Nonetheless, true value can be provided by advanced listening solutions that perform a well-rounded topic-based sentiment analysis to uncover trends and patterns and aid strategic decision-making. They sit comfortably within a company's larger social strategy by providing assistance on the way to reaching its clearly defined goals and achieving resounding success.

Whether companies choose an in-house or third-party approach, decisions must be made on a case to case basis, after weighing the pros and cons of each strategy.

There are multiple third-party tools available, offering different capabilities at different price points. Withal, one in particular is of most importance to this dissertation and has been set both as a starting point and a comparative reference for the project. To protect sensitive information, this application will be referred to as **ToolX**. **ToolX** is the company's current choice for a social listening tool.

1.2.5 ToolX: Square One

ToolX is a complete Social Business Intelligence platform that performs tasks on several domains: from Social Listening, Customer Relationship Management, Analytics, News and Ads, to Influencer Marketing. It is capable of:

- evaluating brand presence and performance on the most important social media platforms,
- monitoring what is said about the brand and competitors in the digital spaces,
- managing relationships with customers,
- creating personalized reports and feed real-time dashboards,
- providing assistance on Digital Media Investments.

One of its core functions is monitoring customer sentiment on a brand and respective competitors, by assigning an appropriate **sentiment** and **topic** to each extracted comment and providing a **visualization** platform to communicate the results.

1.3 Project Objectives

The heart of the writer is set on developing a tool that is relevant for the company and opens doors to new possibilities. Listening is, undoubtedly, an ability that connects companies to people, which must be cherished at all costs. To walk side by side with the customer is to understand his/hers needs and attend to them when the time is right.

This dissertation finds its purpose in aiding the company on its path to acquiring a better understanding of its customers, through a tool capable of analysing the sentiment expressed on social media which is, directly or indirectly, reflected on the company's products and services. The following objectives summarize these intents and provide a clear view of the necessary steps to take in order to fulfil the main purpose:

1. Create a scalable sentiment analysis tool that is capable of classifying brand-related comments, extracted from Social Media, as negative, neutral or positive.
2. Create a topic modelling tool that complements this analysis, by binding sentiment to a target, and encourages the company to make changes where necessary.
3. Provide valuable insights on the results acquired.

1.4 Methodology

The evolution of this project was formalized according to the main objectives of the dissertation.

An initial time window was set for required tools to be acquired such as Python Language Knowledge and specific Text Mining and Sentiment Analysis concepts. The three following main tasks relate to Sentiment Analysis, Topic Modelling and Visualization, respectively. The organic structure of the Natural Language Processing tasks is reflected in the pre-established project subtasks. The following Gantt Chart describes the work structure and corresponding timing.

Table 1 - Project tasks, subtasks and respective deadlines

TASK	SUBTASK	DEADLINE
Sentiment Analysis	Preprocessing and Feature Extraction	30th March
	Model Creation and Selection	15th April
	Necessary Adjustments	30th April
Topic Modelling	Preprocessing and Feature Extraction	20th May
	Model Creation and Selection	30th May
	Necessary Adjustments	5th June
Visualization	Dashboard Planning	10th June
	Dashboard Production	15th June

1.5 Dissertation Structure

Every story should start by providing the reader with enough context and information to develop a genuine interest in the plotline. This dissertation is no exception. On Chapter 2 (State of the Art) the reader will be introduced to the most important Sentiment Analysis ideas, both from a conceptual and technical point of view.

From definition and terminology, to applications and challenges (Section 2.1), landing on the Fundamentals and Approaches (Section 2.2), multiple tools will be provided to understand the Problem and respective Solution, presented in Chapter 3 (Problem and Solution Description). On Chapter 4 (Implementation), the reader is guided through the chronological steps required in the tool development stage. From resources to the reasoning behind specific choices, everything is detailed in this section.

Chapter 5 (Results) represents the culmination of all the acquired knowledge in previous chapters. Results are analysed and important insights are extracted through visualization tools.

The journey ends in Chapter 6, where the key takeaways from this dissertation are highlighted and the reader is invited to join the author on a reflection about the challenges overcome and possible expansion ideas which can elevate the functionality of the developed tool.

2 State of the Art

2.1 Sentiment Analysis, a necessary problem

2.1.1 Definition and Terminology

According to (Liu, 2012), what has been referred to as opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining, among others, has been aggregated under the umbrella of sentiment analysis or opinion mining. Both terms have been conceptualized, in the same book, as follows: a field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes.

Primordial research on the subject refers back to the 90s. The beginning of the millennium pinpoints, however, the official beginning of sentiment analysis research, having the terms sentiment analysis and opinion mining appeared shortly after, in (Nasukawa and Yi, 2003) and (Dave, Lawrence and Pennock, 2003) respectively. Since then, as stated in (Pang and Lee, 2008), there has been a subsequent growth of the field, due to the rise of machine learning methods in natural language processing, the availability of training datasets, due to the growth of social media and the realization of the fascinating intellectual challenges and applications this area offers.

2.1.2 Applications

The potential applications of sentiment analysis are vast and powerful (Kazmaier and van Vuuren, 2020)**.** In the past decade, this field has seen incredible growth, sustained by its clear versality, and has become one of the most active research areas in NLP, growing beyond the domains of computer science into management sciences and almost every possible industrial domain, from consumer products and services to healthcare, financial services, social events and political elections. Many big corporations (e.g., Microsoft, Google, Hewlett-Packard, SAP, SAS) have built in-house sentiment analysis capabilities (Liu, 2012) and shifts in sentiment on social media have been shown to correlate with shifts in the stock market (Kazmaier and van Vuuren, 2020). Applications can be aggregated as follows (Pang and Lee, 2008):

1. **Review-Related Websites:** encompasses review-oriented search engines and automated review and opinion-aggregation websites. Topics can vary from product reviews to opinions on political issues.
2. **As a Sub-component of Technology:** sentiment analysis can serve as an enabling technology for other systems with multiple purposes (e.g., recommendation systems, hate language detection, ad personalization).

3. **Business and Government Intelligence**: Sentiment analysis can be used in reputation management, public relations, trend prediction and government intelligence. These technologies allow to answer questions such as "Why aren't consumers buying our laptop?"(Lee, 2003), because they are able to create a digest of overall consensus points of the reviews related to the product.
4. **Across different domains:** From politics and legal matters to sociology and biology, sentiment analysis technologies have been able to find a place in the good graces of a remarkably diverse set of fields.

2.1.3 The challenges of sentiment analysis

Sentiment analysis systems require overcoming multiple challenges, which I have presently divided into two categories: those which are a consequence of conceptual complexity and those which derive from linguistic subtleties.

Conceptual Complexity:

The development of a complete review or opinion-search application might involve attacking each of the following problems (Pang and Lee, 2008):

- Determining which documents are topically relevant to an opinion-oriented query. This may or may not be a difficult problem in and of itself: perhaps queries of this type will tend to contain indicator terms like "review", "reviews" or "opinions".
- Determining which documents or portions of documents contain review-like or opinionated material. These can vary quite widely in content, style, presentation, and even level of grammaticality.
- Identifying the overall sentiment expressed by the fetched documents and/or the specific opinions regarding particular features or aspects of the items or topics in question presents its difficulties. Free-form text can be much harder for computers to analyse, and indeed can pose additional challenges.
- Finally, the system needs to present the sentiment information it has garnered in some reasonable summary fashion, textually or through visualization. This can involve some or all of the following actions, among others:
  - Aggregation of "votes" that may be registered on different scales.
  - Selective highlighting of some opinions.
  - Representation of points of disagreement and points of consensus.
  - Identification of communities of opinion holders.
  - Accounting for different levels of authority among opinion holders.

Sentiment Lexicon and NLP Challenges

Sentiment Analysis has to deal with several challenges which emerge as a direct consequence of the subtleties of language. *Sentiment words* or *opinion words* are words commonly used to express positive sentiments, *good* and *amazing*, or negative sentiments, such as *bad* and *horrible.* These sentiments can also be described by phrases such as *cost someone an arm and a leg*. A list of such words and phrases is called a *sentiment lexicon*, which is instrumental to sentiment analysis for obvious reasons, but not sufficient (Liu, 2012).

In addition, coming up with the right set of keywords is far from easy. A study by (Pang, Lee and Vaithyanathan, 2002), compared the accuracy achieved by hand-picked keyword lists to the one achieved by statistically created word lists of the same size. The former achieved about 60% accuracy, opposed to the latter, which achieved almost 70% accuracy.

The following issues also present major challenges to sentiment analysis (Liu, 2012):

- A positive or negative sentiment word may have opposite orientations in different application domains. For example, "suck" usually indicates negative sentiment, e.g., "*This camera sucks*," but it can also imply positive sentiment, e.g., "*This vacuum cleaner really sucks*.". Note that even the exact same expression can indicate different sentiment in different domains. For example, "go read the book" most likely indicates positive sentiment for book reviews, but negative sentiment for movie reviews (Pang and Lee, 2008).
- A sentence containing sentiment words may not express any sentiment. Interrogative sentences and conditional sentences are two important types, e.g., "*Can you tell me which Sony camera is good*?" and "*If I can find a good camera in the shop, I will buy it*." Both these sentences contain the sentiment word "*good*", but neither expresses a positive or negative opinion on any specific camera. However, not all conditional sentences or interrogative sentences express no sentiments, e.g., "*Does anyone know how to repair this terrible printer*" and "*If you are looking for a good car, get Toyota Camry."*
- Sarcastic sentences with or without sentiment words are hard to deal with, e.g., "*What a great car! It stopped working in two days.*"
- Many sentences without sentiment words can also imply opinions. Many of these sentences are actually objective sentences that are used to express some factual information. The sentence "*After sleeping on the mattress for two days, a valley has formed in the middle*" expresses a negative opinion about the mattress, although it has no sentiment words, and is objective as it states a fact.

(Pang and Lee, 2008) pertinently refers that, somewhat in contrast with topic-based text categorization, the order in which opinions are presented can be a more determinant factor than the frequency of the opinion. The following movie review serves as an example:

> *This film should be **brilliant**. It sounds like a **great** plot, the actors are **first grade**, and the supporting cast is **good** as well, and Stallone is attempting to deliver a **good** performance. However, it can't hold up.*

In this situation, although the positive words (in bold) dominate this excerpt, the overall sentiment is negative because of the crucial last sentence.

Moreover, one must not forget sentiment analysis falls under the NLP umbrella, which means it inherits some NLP challenges, such as coreference resolution, negation handling, and word sense disambiguation.

2.2 Fundamentals and Approaches

As discussed above, pervasive real-life applications are only part of the reason why sentiment analysis is a popular research problem. It is also highly challenging as an NLP research topic and covers many novel sub-problems as we will see later.

2.2.1 Levels of Analysis

In general, sentiment analysis has been investigated mainly at three levels:

Document level: By assuming each document expresses opinions on a single entity, the aim of this level of analysis is to classify whether a whole opinion document expresses a positive or negative sentiment on that entity (Turney, 2001; Pang, Lee and Vaithyanathan, 2002). Thus, it is not the best option if the goal is to study or compare multiple entities.

Sentence level: The aim is to determine whether each sentence expresses a positive, neutral or negative sentiment (neutral usually refers to lack of opinion). Note that objectivity does not imply lack of opinion, therefore, the analysis goes beyond *subjectivity classification*.

Entity and Aspect level: The previous levels of analysis lack a very frequently required functionality within the application contexts of sentiment analysis: to understand the target of people's opinions. According to (Liu, 2012) *aspect level* (or *feature level)* sentiment analysis solves this gap by realizing that an opinion consists of a *sentiment* (positive or negative) and a *target* (of opinion), and allows to create more informative and structured opinion summaries about entities and respective aspects/characteristics:

> *Thus, the goal of this level of analysis is to discover sentiments on entities and/or their aspects. For example, the sentence "The iPhone's call quality is good, but its battery life is short" evaluates two aspects [targets], call quality and battery life, of iPhone (entity). The sentiment on iPhone's call quality is positive, but the sentiment on its battery life is negative.*

2.2.2 The Sentiment Analysis Problem

The sentiment analysis problem can be perceived as a "rich set of inter-related sub-problems" (Liu, 2012). A single opinion is not enough to characterize the sentiment on an entity or aspect; thus, a collection of opinions must be analysed. Opinion can be expressed in several forms of formal or informal text such as news articles, social media posts (Twitter, Facebook, Instagram, etc.), forum discussions and blogs, which can vary in difficulty of analysis. Forum discussions (longer, varied and usually include interaction) are, typically, the hardest form to deal with, when compared with social media posts, which are shorter and straightforward. Moreover, opinions can be expressed on vastly different subjects, which also vary in difficulty of analysis. Social and political discussions are harder to analyse than opinions on products because complex expressions and sarcasm are more frequently found.

To fully understand the sentiment analysis problem, it is of utter importance to find a valid definition of ***opinion***.

Definition of Opinion (Hu and Liu, 2004; Liu, 2010): a quintuple, $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, where:

- $\mathbf{e_i}$ - name of entity
- $\mathbf{a_{ij}}$ - an aspect of entity $\mathbf{e_i}$ ($\mathbf{e_i}$ and $\mathbf{a_{ij}}$ together represent the opinion target)
- $\mathbf{s_{ijkl}}$ - the sentiment on aspect $\mathbf{a_{ij}}$ of entity $\mathbf{e_i}$ (positive, negative, or neutral, or expressed with different intensity levels)
- $\mathbf{h_k}$ - the opinion holder
- $\mathbf{t_l}$ - the time when the opinion is expressed by $\mathbf{h_k}$

Please note the following:

1. Missing any of components can be problematic.
2. An entity can be divided into parts, which can be divided into sub-parts, and so forth, each with its related attributes. This definition only considers the entity and its attributes, which, despite some limitations, encompasses most applications.
3. The different components of the definition can serve as attributes of a database schema, which transforms unstructured into structured data.
4. When dealing with comparative opinions, another definition is required.

2.2.3 Sentiment Analysis Tasks

This definition enlightens the framework which must be followed while dealing with an aspect-based sentiment analysis problem. The **goal of sentiment** analysis is to discover all opinion quintuples (ei, aij, sijkl, hk, tl) in a given opinion document d (Liu, 2010).

According to (Liu, 2012), given a set of opinion documents D, sentiment analysis consists of the following 6 main tasks, which can be extrapolated from the components of the quintuple:

1. **Entity extraction and categorization:** Extract all entity expressions in D and categorize or group synonymous entity expressions into entity clusters (or categories). Each entity expression cluster indicates a unique entity ei.
2. **Aspect extraction and categorization:** Extract all aspect expressions of the entities and categorize these aspect expressions into clusters. Each aspect expression cluster of entity ei represents a unique aspect aij.
3. **Opinion holder extraction and categorization:** Extract opinion holders from data and categorize them.
4. **Time extraction and standardization:** Extract the timing of opinions and standardize different time formats.
5. **Aspect sentiment classification:** Determine whether an opinion on an aspect aij is positive, negative or neutral, or assign a numeric sentiment rating to the aspect.
6. **Opinion quintuple generation:** Produce all opinion quintuples (ei, aij, sijkl, hk, tl) expressed in document d based on the results of the above tasks.

A seventh equally necessary task can be considered, when dealing with opinions from a large number of people: **opinion summarization**. This can take the form of a structured or short text summary and should always quantify results to express the general level of sentiment (e.g. 80% of positive opinions).

A common form of summary is called aspect-based opinion summary (or feature-based opinion summary) (Hu and Liu, 2004).

2.2.4 Topic Modelling as A Tool to Aid Aspect-Based Sentiment Analysis

Topic models and Fundamental Concepts

In topic modelling the word “topic” takes on the specific meaning of a probability distribution over words, while still alluding to the more general meaning of a theme or subject of discourse (Boyd-Graber, Hu and Mimno, 2017).

As stated by (Boyd-Graber, Hu and Mimno, 2017), topic modelling began with a linear algebra approach called Latent Semantic Analysis. However, the most agreed upon approaches are the ones with a probabilistic nature, which are “intuitive, work well, and allow for easy extensions”. These approaches encompass the Latent Dirichlet Allocation (LDA) and the Probabilistic Latent Semantic Analysis (pLSA).

Probability distributions can be considered the building blocks of topic modelling strategies because they pave the way for topic inference from available data. The distributions in Figure 1 are of great importance for this research domain.

Distribution	Density	Example Parameters	Example Draws
Discrete	$\prod_i \phi_i^{\mathbb{1}[w=i]}$	$\phi = \begin{bmatrix} 0.1 \\ 0.6 \\ 0.3 \end{bmatrix}$	$w = 2$
Dirichlet	$\frac{\prod_{i=1}^{K}\Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1}$	$\alpha = \begin{bmatrix} 1.1 \\ 0.1 \\ 0.1 \end{bmatrix}$	$\theta = \begin{bmatrix} 0.8 \\ 0.15 \\ 0.05 \end{bmatrix}$

Figure 1 - Important probability distributions for topic modelling

Discrete: Discrete distributions describe the connection between both (1) word and topics and (2) topics and documents (2), hence being the star player in topic modelling, according to (Boyd-Graber, Hu and Mimno, 2017)**.** Each topic distribution (distribution over words) assigns higher weights to some words more than others and the same happens between documents and topics. This is the tool that allows topics to be allocated to documents according to the words that compose them.

Dirichlet: Topic modelling often begins with Dirichlet distributions by producing probability vectors that can be used as the parameters of discrete distributions. They have parameters analogous to a mean and variance, which are often combined into a single measure for each dimension: $\alpha_k = \alpha_0\tau_k$:

- the "base measure" τ is the expected value of the Dirichlet distribution;
- the concentration parameter α_o controls the distance between individual draws and the base measure: higher concentration corresponds to smaller distances; lower concentrations mean higher sparsity.

A sparse distribution allocates high probabilities to a few specific values and low probabilities to all others, therefore, the sparsity of Dirichlet distributions is the probabilistic tool that encodes one's intuition to write about a specific set of topics and not an absurd range of subjects.

2.2.5 Literature Review on Frameworks

A number of frameworks for sentiment analysis have been proposed in the literature. Most articles either take a generic or a specific approach. The latter approaches usually aim to improve classification performance for a specific domain or language using preprocessing techniques and sentiment classification models. It is less common to find literature that explores all the generic concepts at a lower level of abstraction. Note that the specificity decreases the usability of the solutions proposed, as the same level of performance is not guaranteed in a different domain.

According to (Kazmaier and van Vuuren, 2020), whilst an accurate sentiment classification model is necessary to evaluate opinionated content, it is not sufficient to form an understanding of the overall sentiment present in the data. To properly examine customer feedback, it is also necessary to identify:

- which aspects of a product or service contributed to customer satisfaction or dissatisfaction;
- any trends that may indicate why certain customer segments are (dis)satisfied.

Generic Frameworks:

(Khan, Bashir and Qamar, 2014) proposed a high-level generic framework:

1. Retrieval and preprocessing of input data;
2. Information extraction;
3. Sentiment analysis (polarity classification);
4. Summarization of results and visualization through a *graphical user interface* (GUI).

It the same paper, a more specific framework was also proposed. The analysis was tailored to social media posts on Twitter: *Twitter Opinion Mining Framework* (TOM).

The lack of generalisability of most frameworks has typically been addressed by:

- designing frameworks in a modular, extendible manner;

- incorporating several different models into the framework.

Visualization-Focused Frameworks:

(Liu, Hu and Cheng, 2005) attended to this necessity by implementing a prototype system called Opinion Observer which follows a "novel framework for analysing and comparing consumer opinions of competing products". Given a set of products and URLs of Web pages that contain customer reviews, Opinion Observer works in two stages:

Stage 1: Extracting and analysing customer reviews in two steps:

- Step 1: Periodically download reviews from pages. All raw reviews are stored in a database.
- Step 2: Analyse all new reviews of every product, by identifying product features and opinion orientations from each review.

Stage 2: Visualization and comparison of opinions on competing products through a user interface that easily highlights the "strengths and weaknesses of each product in the minds of consumers in terms of various product features", with the help of a histogram.

As a visualization framework introduction, this article is considered a pinnacle of research, although, by using semi-structured customer reviews, it doesn't fully explore the sentiment classification tasks.

(Lucena, 2016), followed a similar approach to that of (Liu, Hu and Cheng, 2005) and generated meaningful textual summaries which transformed the mentioned features and sentiment polarities into **recommendations for action**. It proposes a knowledge management system which transforms the gathered knowledge into explicit ontologies and allows "to build tools with advanced reasoning capacities with the aim to support enterprises decision-making processes".

Multiple other specific frameworks have been proposed by different authors, some that incorporate structured data, others that analysed the effect of specific features on sentiment predictions.

Generic and Adaptable Framework

(Kazmaier and van Vuuren, 2020) searched for a framework which would be distinguished from the previous literature largely by the following characteristics:

1. The framework is *interactive* with a focus on facilitating rather than automating the evaluation of opinionated data by a user.
2. Rather than incorporating a specific model into the framework, the user is guided through the *model development* process, with a particular focus on the machine learning approach, where algorithm selection, parameter and hyperparameter tuning, as well as feature selection, are required.
3. Instead of merely presenting model results, the framework facilitates an *exploratory analysis* of these results, including an investigation of the relationship between sentiment and data attributes from supplementary, structured data sources.
4. The framework is designed with the objectives of generalisability and flexibility, further supporting its applicability to various problem domains.

2.2.6 The KDD Process

Data Mining and the Knowledge Discovery in Databases (KDD) process are often used interchangeably because the former is part of the latter (Gheware, Kejkar and Tondare, 2014). The KDD process is highly iterative and requires user input. In (Fayyad and Stolorz, 1997), its basic steps are broadly outlined:

1. Problem and Goal Definition
2. Data Selection
3. Data Cleaning and Preprocessing: removing noise, handling missing data, among other tasks;
4. Data Reduction and Projection: selection of most important features;
5. Match goals of the KDD process to a particular data-mining method;
6. Exploratory analysis and model and hypothesis selection: selecting the data mining algorithm(s) and method(s);
7. Data Mining;
8. Pattern Visualization and Interpretation: possible return to step 1 for further iteration;
9. Acting on the Discovered Knowledge: using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parties.

In the same article, the authors also raise awareness to the importance and clear impact of all steps on the success of the KDD application.

2.2.7 The Data Mining Step

The data mining component of the KDD process often involves repeated iterative application of particular methods to achieve specific goals divided according to the intended use of the system (Fayyad and Stolorz, 1997):

- Verification – aims to verify the user's hypothesis.
- Discovery – aims to autonomously find new patterns and encompasses:
  - *prediction,* where the system finds patterns for predicting the future behaviour of some entities,
  - *description,* where the system finds patterns for presentation to a user in a human-understandable form.

These goals can be achieved by performing a variety of data mining tasks (Gheware, Kejkar and Tondare, 2014):

1. **Summarization**: Abstraction of data to create a smaller set able to provide a general overview of that data. According to (Fayyad and Stolorz, 1997)**,** summarization techniques are often applied to interactive exploratory data analysis and automated report generation.
2. **Clustering**: Grouping a set of objects in such a way that minimizes differences within clusters and maximizes differences between clusters. Objects are, then, labelled and common features within clusters are summarized to create class descriptions. E.g. a bank may cluster its customers into several groups based on their similarities, to allow customization of its services.

3. **Classification**: Learning rules that can be applied to new data and will typically include the following steps: pre-processing of data, designing modelling, learning/feature selection and validation/evaluation. A set of objects is given as training set in which every object is represented by a vector of attributes along with its class. By analysing the relationship between attributes and class of the objects in the training set, classification model can be constructed. Such classification model can be used to classify future objects and develop a better understanding of the classes of the objects in the data base.

   According to (Lucena, 2016), some common application domains in which the classification problem arises, are the following: Customer Target Marketing, Medical Disease Diagnosis, Supervised Event Detection, Multimedia Data Analysis, Document Categorization and Filtering, Social Network Analysis.

4. **Regression**: Finding the best function (that minimizes error) to model the data. It is widely used for **prediction** and **forecasting** and allows to explore relationships between independent and dependent variable. In this context, time series data can be used to identify temporal trends in sales and resource costs or changes in key drivers of demand (Lucena, 2016).

5. **Association**: Looking for relationship between variables or objects. It aims to extract interesting association, correlations or casual structures among the objects. Association rules can be very useful in marketing and advertising contexts, as they allow to formulate affirmations such as: "A customer who buys products x_1 and x_2 will also buy product y with probability p" (Lucena, 2016).

In (Lucena, 2016), visualization is presented as an additional data mining task that, used in conjunction with other data mining models, can provide a clearer understanding of the discovered patterns or relationships.

Indeed, Sentiment Analysis falls into the scope of Supervised Classification problems, along with a multitude of Machine Learning algorithms. Some have, however, shown to perform better than others. In (Sousa, 2019), the focal point of the research was automatic hate speech detection in text, which is a specific sentiment analysis problem. He summarized the frequency of algorithms used on this type of problem. Deep Learning approaches lead the ranking of most used alternatives, followed by SVM, Ensemble Learning and Logistic Regression.

2.2.8 Preprocessing

To enhance the Sentiment Analysis and Topic Modelling performance, a solid preparation of the Dataset is unquestionably imperative. Transforming text into something an algorithm can digest is, indeed, a complicated but necessary process, which can be divided into:

Cleaning: remove superfluous parts of text through stopword removal, lowercasing, among others.

Annotation: Annotations may include structural markup and part-of-speech tagging.

Normalization: Linguistic reductions through Stemming, Lemmatization and other forms of standardization.

Tokenization: Dividing text into smaller components (sentences, words or characters).

Considering our target is social media, more specifically social networks such as Twitter, there's a big linguistic diversity in the content we may find in the platforms. Whether we focus on English, Portuguese or any other language, the amount of noise in the data is substantial due to the comments' shortness and informality, usually containing useless or

unknown characters, emoticons, among other things. In any machine learning problem, it is important to have clean data in order to maximize the efficiency of the algorithms used in the classification processes.

According to (Sousa, 2019), the following techniques can be applied in this context:

- Tokenization
- Lowercasing
- Punctuation and irrelevant character removal
- Emoji removal or transformation
- Stemming and Lemmatization: Both are text normalization techniques used to prepare text, words, and documents for further processing. The first transforms words into their root form, while the second takes morphological and language-specific information into account, such as Part of Speech, and, therefore, can be considered more informative. Stemming techniques transform 'studies' into 'studi' and 'studying' into 'study', in opposition to Lemmatization techniques, which transform both words into study.
- Stopword removal: remove commonly used which don't provide any beneficial information.
- Part-of-Speech process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context (Godayal Divya, no date)
- Spell checker

2.2.9 Feature Extraction and Selection

Feature extraction plays a prominent role in sentiment analysis. In fact, extracting informative and essential features greatly enhances the performance of machine learning models and reduces the computational complexity (Avinash and Sivasankar, 2019). Most machine learning algorithms require numeric input data. One can easily come to the conclusion that the textual input data we are in possession of doesn't fit this criterion and must, therefore, be vectorized, i.e. each token must be represented numerically (Sousa, 2019).

Vectorization Strategies:

Bag of Words: Based on BoW, each element of the feature vector can be the word occurrence (absence or presence), word frequency, or TF-IDF score. Due to the high dimensionality of the vocabulary incorporated into the textual contents, a BoW vector is noticeably sparse (Zhang, Wang and Liu, 2018). Additional remarks dwell on the word order that is inconveniently neglected, which means semantics cannot be encoded and, as long as two documents use the same vocabulary, they will also share the same BoW representation (Zhang, Wang and Liu, 2018).

N-Grams (from Bag-of-N-Grams): Works as an extension of BoW. N-grams considers sets of words or characters (if n is higher than 1), as tokens, instead of only taking single words into account (Sousa, 2019). Each element of the feature vector maintains the same meaning as in BoW. N-Grams can consider the word order in a short context, but also suffers from data sparsity and high dimensionality (Zhang, Wang and Liu, 2018).

TF-IDF (term frequency-inverse document frequency): The TF-IDF considers the frequency of each token according to its inverse frequency in the corpus. This means that tokens with less occurrences have a weighted frequency value higher than those with high occurrences **(Sousa, 2020).** It is largely used in information retrieval and text mining and is obtained by multiplying the Term Frequency (TF) and the Inverse Document Frequency (IDF).

- **TF** measures the number of times a particular term occurs in a document and divides it by the number of terms in that particular document. Frequency increases when the term has occurred multiple times.
- **IDF** is used to elevate the most important terms and detract from word such as stopwords which, although frail in relevance, generally occur multiple times in several documents. It is computed as the natural logarithm of the ratio between the total number of documents and the number of documents that include the term (Avinash and Sivasankar, 2019).

Word Embeddings: According to (Zhang, Wang and Liu, 2018), word embedding techniques based on neural networks were proposed to tackle the shortcomings of BoW and Ngram methods, by generating low-dimensional vectors which are, to some extent, able to encode some semantic and syntactic properties of words. This technique can still be used for non-neural learning models.

Other Features

Besides features acquired through text vectorization, some other semantic features can aid the classification procedure. Some linguistic and semantic information can be of service, such as document length, average word length, number of punctuation marks and number of capital letters. For product reviews, it can be beneficial to model sentiment with some additional information, such as user and product information (Zhang, Wang and Liu, 2018).

Feature Selection

Feature selection is, also, important for text classification because it allows to determine and select the features that are most relevant to the classification process (Aggarwal, 2014). Multiple methods have been designed to perform this task. A commonly used strategy in NLP is to define a minimum and maximum token frequency to be accepted, when generating document representations. TF-IDF values can also serve as a criterium to be compared with a predefined threshold.

2.2.10 Algorithms

Classification Algorithms

Most methods for quantitative data can be used directly on text, after modelling it as quantitative data. Word attributes are, however, sparse, highly dimensional and not frequent (Aggarwal, 2014). Still, a wide variety of techniques can be applied to text classification problems, from which the following have been highlighted:

Logistic regression: A (predictive) regression analysis which estimates the parameters of a logistic model, a statistical model that uses a logistic function to model a binary dependant variable (Sousa, 2019). It distinguishes itself from linear regressions by using a different hypothesis that predicts the probability that a given example belongs to class 0 or 1 (Avinash and Sivasankar, 2019). The model is simple and easy to interpret, but it is rigid when modelling more complex non-linear relations.

Naive Bayes Classifier: Bayesian learners are probabilistic models, based on the Bayes Theorem, with a strong naive independence assumption between the features (Sousa, 2019). The idea is to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the documents (Aggarwal, 2014)**.** Of course these assumptions of independence are rarely true, which may explain why some have referred to the model as the "Idiot Bayes" model, but in practice Naive Bayes models have performed surprisingly well, even on complex tasks where it is clear that the strong independence assumptions are false (Russell and Norvig, 2003). For multiclass classification problems it is common to use a Multinomial Naive Bayes classifier, which is a specific

instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features.

Decision Trees: Performs classification by using yes or no conditions (Avinash and Sivasankar, 2019)**.** A decision tree is constructed in an iterative way with the use of a hierarchical division of the underlying data space designed in order to create class partitions that are more skewed in terms of their class distribution (Aggarwal, 2014). In each step, the learning algorithm chooses one feature and creates a new branch for each of its possible values. Therefore, each inner node corresponds to a feature; the edges represent decisions for one of the feature's possible values. A leaf represents the predicated value of the target variable (Lucena, 2016). When the unlabelled data samples are to be classified, they pass through series of test nodes finally leading to the decision node with a class label to which the sample can be assigned (Avinash and Sivasankar, 2019). Its advantages include interpretability, speed and good performance on large datasets, although they are prone to overfitting (Sousa, 2019)**.**

Support Vector Machines: SVM Classifiers attempt to partition the data space with the use of linear or non-linear delineations between the different classes, known as hyperplanes. The key in such classifiers is to determine the optimal boundaries between the different classes and use them for the purposes of classification (Aggarwal, 2014). SVMs are widely used in classification problems. In fact, in 2017, SVMs held the best results for text classification tasks, having been later surpassed by Deep Learning in 2018 (Sousa, 2019).

Ensemble Methods:

Meta-algorithms play an important role in classification strategies because of their ability to enhance the accuracy of existing classification algorithms by combining them or making a general change in the different algorithms to achieve a specific goal (Aggarwal, 2014). The most commonly used ensemble methods are:

Random Forest: Bagging methods are generally designed to reduce the model overfitting error that arises during the learning process. Their main goal is to reduce the variance component of the underlying classifier (Aggarwal, 2014). The Random Forest classifier specifically combines decision tree predictors in order to yield the final result. It implicitly performs feature selection, requiring very little preparation. It has a quick and, overall, good performance (Sousa, 2019).

Gradient Boosting: Contrary to bagging methods, training models in a boosting method are not constructed independently, but sequentially. Specifically, after *i* classifiers are constructed, the *(i+1)* th classifier is constructed on those parts of the training data that the first *i* classifiers are unable to accurately classify. The results of these different classifiers are combined together carefully, where the weight of each classifier is typically a function of its error rate (Aggarwal, 2014). Such boosting algorithms usually consist of an ensemble of weak prediction models, typically decision trees (that's why it may also be called gradient boosted trees) (Sousa, 2019).

Deep Learning:

Learning mechanisms are at the heart of how the brain processes information (Lucena, 2016). (Rolls, 2000) states that useful neuronal information processors for most brain functions are built by modifying the synaptic connection strengths (or weights) between neurons. Learning requires this neuronal process, which, according to (Patterson, Nestor and Rogers, 2007), can be facilitated by manipulating emotions through rewards and punishments.

This biological knowledge has inspired a revolution in the Data Mining domain. Deep learning's popularity has been increasing significantly over the recent years, especially in text

classification (Sousa, 2019). In fact, the study and computer modelling of the learning process in their multiple manifestations has been the most challenging and fascinating goal in Artificial Intelligence (Carbonell, Michalski and Mitchell, 1983). As stated in (Lucena, 2016), techniques, such as artificial neural networks increase a system's machine intelligence quotient (ability to represent and deal with knowledge). The disclosure of this architecture has made it possible and easier to tune the parameters and, consequently, produce better results, outperforming baseline algorithms (Sousa, 2019).

Artificial Neural Networks (ANN) attempt to simulate biological systems, corresponding to the human brain. In the human brain, neurons are connected to one another via points, which are referred to as synapses. In biological systems, learning is performed by changing the strength of the synaptic connections, in response to impulses (Aggarwal, 2014). They can learn from existing data and experience even when humans find it difficult to identify rules and are able to adapt when facing new data. Note that, if the ANN is implemented as a 'black box', then any information acquired by the network during the training is unavailable. Previous research developed design techniques that allow network operation to be decoded after training, facilitating the employment of user's feedback to adapt the ANN (Lucena, 2016)**.**

The main artificial neural networks' architectures are described below (Sousa, 2019):

Multilayer Perceptron (MLP): a class of feedforward ANN consisting of at least three layers of nodes: an input layer, a hidden layer and an output layer (*Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning*, no date). It is one of the most traditional types of Deep Learning architectures, where every element of a previous layer, is connected to every element of the next layer. The transformation is encoded by matrixes. MLP utilizes a supervised learning technique called backpropagation for training.

Convolutional neural networks (CNN): A type of feed-forward artificial neural networks that consists of an input and output layer and multiple hidden layers: convolutional layers, pooling layers and fully connected layers.

Recurrent neural networks (RNN): A class of artificial neural networks that, unlike CNNs, are able to handle sequential data, allowing to produce temporal dynamic behaviours according to a time sequence. RNN's have feedback loops in the recurrent layer, which act as a memory mechanism, although long-term dependencies can still post some challenges. As a development of RNN, other architectures were created:

- **Long Short-Term Memory** (LSTM) neural networks include a memory cell able to keep information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten.
- **Gated Recurrent Unit** (GRU) neural networks are similar to LSTM's, but their structure is slightly simpler. Although they also use a set of gates to control the flow of information, these are fewer when compared to LSTM's.

Classification Performance Evaluation

As stated in (Hossin and Sulaiman, 2015), the evaluation metric can be categorized into three types, (1) threshold, (2) probability and (3) ranking metric, the first two being the most common. Besides, these metrics can be employed with three different evaluation purposes:

1. To evaluate the generalization ability of the trained classifier.
2. To determine the best classifier among different types of trained classifiers which focus on the best future performance (optimal model) when tested with unseen data.

3. To select the optimal solution among all generated solutions during the classification training.

Taking into account the fact that the present implementation is dealing with a multiclass classification problem which can be highly benefited by the application of performance metrics, the following measures can be applied to this context, as reported by (Hossin and Sulaiman, 2015):

Averaged Accuracy: Ratio of correct predictions over total number of instances evaluated.

Averaged Error Rate: Ratio of incorrect predictions over the total number of instances evaluated.

Averaged Precision: Correct predictions over the total predicted patterns in a positive class.

Averaged Recall: Fraction of positive patterns that are correctly classified

Averaged F-Measure: Harmonic mean between recall and precision values.

Mean Square Error: Difference between the predicted solutions and desired solutions.

AUC: AUC (Area Under the Curve), also known as AUROC (Area Under the ROC Curve), is one of the popular ranking type metrics. A ROC curve (receiver operating characteristic curve) displays the performance of a classification model at all thresholds, according to its True Positive Rate or Recall (True Positives / all Positive observations) and False Positive Rate (False Positives / all Negative observations). AUC measures the entire two-dimensional area underneath the entire ROC curve, therefore, providing an aggregate measure of performance across all possible classification thresholds. Unlike the previous metrics, the AUC value reflects the overall ranking performance of a classifier.

Topic Modelling Algorithms

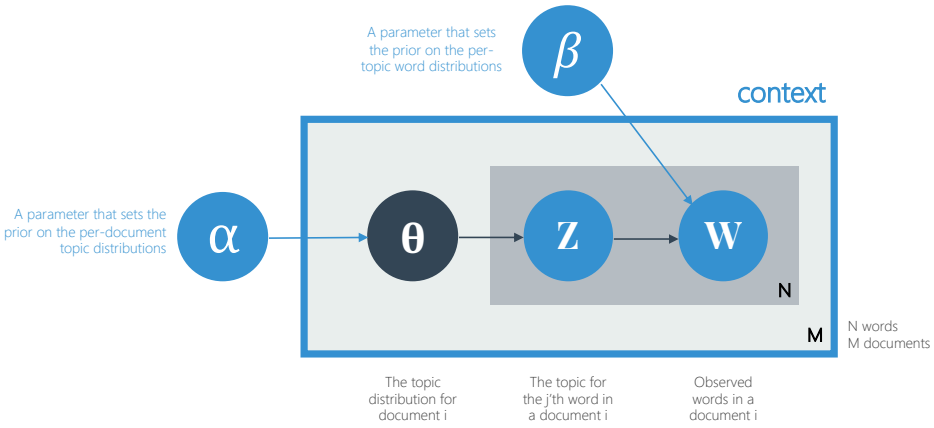


Figure 2 - Conceptual representation of the LDA Problem

A) Latent Dirichlet Allocation (LDA)

In LDA, each document can be considered a composition of topics and each topic, a composition of words. This is similar to the standard bag of words model assumption and makes the individual words exchangeable. It is a generalization of the pLSA model, but the topic distribution is assumed to have a sparse Dirichlet prior, which is able to encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently. Each topic has probabilities of generating various words. A lexical word

may occur in several topics with a different probability, however, with a different typical set of neighbouring words in each topic (*Latent Dirichlet allocation - Wikipedia*, no date).

Latent Dirichlet Allocation works over a "generative process" conception which recreates the story of how the data came to be, i.e. how topics are generated and used to create diverse documents. To infer the topics of a given corpus, the process must, then, be reverse engineered. The following conceptions form the basis of the most popular methodology for topic modelling:

Generating Topics: LDA takes into consideration a user-specified number of distinct topics (K). Each topic is modelled by a Dirichlet distribution over all the words in the vocabulary: $\emptyset_k \sim Dir(\beta)$, where β is the concentration parameter which sets the prior on the per-topic word distributions and is typically sparse.

Document Allocations: Each document is modelled by a Dirichlet distribution over topics: $\theta_i \sim Dir(\alpha)$ where α is a symmetric, typically sparse, concentration parameter which sets the prior on the per-document topic distributions.

It might be helpful to describe θ and ϕ as matrixes created by decomposing the original corpus matrix. In θ, rows are defined by documents and columns are defined by topics. In ϕ, rows represent topics and columns represent words.

Words in Context: As previously stated, this process reflects an inverted image of what we usually want to achieve when applying such method in a practical manner. This means that, after creating both Dirichlet functions, the process will try to generate a document which fits the desired probability distributions. With that being said, for each word *j* in the document *i*, the algorithm will, firstly, assign the topic to a token according to a discrete distribution: $z_{i,j} \sim Discrete(\theta_i)$. After discovering which of the k topics the word token is from, it is necessary to assign a specific word to the token, which is made according to another Discrete distribution: $w_{i,j} \sim Discrete(\emptyset_{z_{i,j}})$. The topic assignment tells you what the word is about, and then selects which distribution over words we use to generate the word.

Inference: The process of discovering the hidden probability distributions given a generative model and a corpus is a problem of statistical inference, which, according to (Boyd-Graber, Hu and Mimno, 2017)**,** can be solved with the help of multiple different algorithms: message passing, variational inference, gradient descent, and Gibbs sampling, the latter being the most frequently used.

B) Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM):

(Yin and Wang, 2014) proposed a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model with the intention of improving short text clustering tasks, which have posed some challenges due to its sparse, high-dimensional, and large-volume characteristics. This algorithm can "infer the number of clusters automatically with a good balance between the completeness and homogeneity of the clustering results and is fast to converge". Their extensive experimental study shows that GSDMM can achieve significantly better performance than three other clustering models. In their paper, the authors base their explanation of the algorithm on a simple conceptual model called the **Movie Group Process**:

Imagine that a professor of a film discussion course plans to assign the students to different tables according to their cinematic taste. In the beginning, the students are randomly assigned to one of the k tables. Each student quickly writes down a personal list of favourite films, which means each student can be represented by that same list. (Yin and Wang, 2014) formalizes the problem as follows:

"The input is D students (documents) and each student (document) is represented by a short list of movies (words). The goal is to cluster the students (documents) into several groups, so that students (documents) in the same group are similar and students (documents) in different groups are dissimilar. We define the number of distinct movies (words) as V. The sparse characteristic of short text means that V is really large (often larger than10^5), while the average number of words (L) in each short text is small (often less than 10^2)."

On the second iteration, the students must select a new table to sit on. It is expected that they choose the table according to the following rules:

- Rule 1: The new table has more students than the current table.
- Rule 2: The new table has students with similar lists of films.

We can expect that the students eventually arrive at an "optimal" table configuration: only a part of the tables will still have students and the students in each table will share similar interests.

Dirichlet Multinomial Mixture: DMM is a probabilistic generative model which follows two assumptions: (1) the documents are generated by a mixture model, and (2) there is a one-to-one correspondence between mixture components and clusters.

In a short text clustering problem, the mixture component (cluster) z for each document d needs to be estimated.

Collapsed Gibbs Sampling for DMM: GSDMM is a soft clustering model since we can get the probability of each document belonging to each cluster. The algorithm runs through the following stages:

1. In the initialization stage, the documents are randomly assigned to k clusters and the following information is recorded: z (cluster labels of each document), m_z (number of documents in cluster z), n_z (number of words in cluster z), and n_z^w(number of occurrences of word w in cluster z);
2. Then we traverse the documents for I iterations (in the paper it is mentioned that GSDMM can achieve good and stable performance when I equals five). In each iteration, a new cluster is re-assigned to each document d according to the conditional distribution $p(z_d = z \mid \vec{z}_{\neg d}, \vec{d})$, where $\neg d$ means the cluster label of document d is removed from $\vec{z}$; and the recorded information is updated
3. Finally, only a part of the initial K clusters will remain nonempty. the number of non-empty clusters can be near the true number of groups as long as K is larger than the true number.

We can derive $p(z_d = z \mid \vec{z}_{\neg d}, \vec{d})$, from the Dirichlet Multinomial Mixture (DMM) model, and find that it conforms to the two rules of MGP previously introduced.

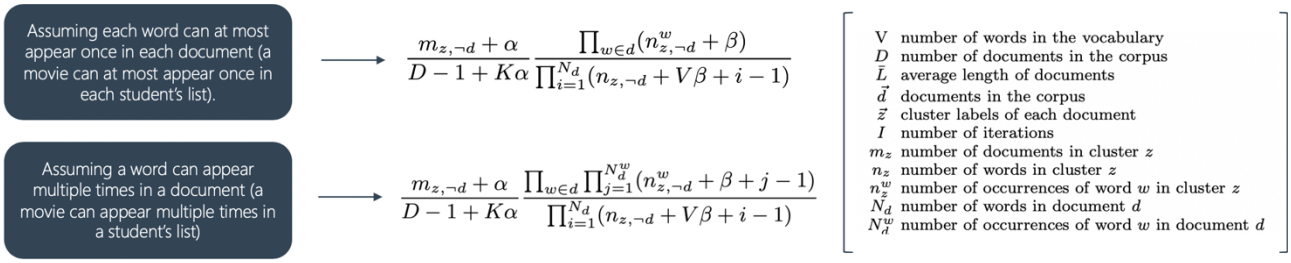


Figure 3 - DMM Equations

Parameters: The GSDMM parameters follow the moulds of LDA:

- α relates to the prior probability of a student (document) choosing a table (cluster), which concerns the first rule of MGP. When α increases, the probability of a student choosing an empty table also increases.
- β relates to the prior probability of a table (topic) choosing a certain film (word), which concerns the second rule of MGP (the new table has students with a similar taste to the student who is choosing.). Higher values of β assign lower importance to a film (word) and vice-versa. Note that some words should be considered more important than others. A film watched by every student (a word that appears in several documents) does not provide any valuable information and can actually mislead the clustering algorithm, therefore, less emphasis should be given to it.

In the end, the main difference between LDA and GSDMM resides in the fact that the latter assumes one topic per document and uses that information to cluster documents together, in opposition to what happens with LDA. Therefore, GSDMM should be more suitable to model short text topics.

Topic Modelling Performance Evaluation

Probabilistic topic modelling tools, such as LDA, although very popular, follow a longstanding assumption that the latent space discovered by them is meaningful and useful. However, because they exist under the umbrella of unsupervised learning, it becomes challenging to assess the veracity of these assumptions as well as compare methods against each other. Topic Models are usually evaluated on some secondary task, such as document classification (Extrinsic Evaluation Metrics) or using Intrinsic Evaluation Metrics (Wallach, Murray and Mimno, 2009)**.** When in possession of ground truth topic annotations, the previously referred classification metrics can be applied to this context, as Extrinsic Evaluation Metrics.

Intrinsic Evaluation Metrics:

Likelihood: Measures how well a model fits the observed data. Increases in likelihood are a consequence of more appropriate models. (Griffiths and Steyvers, 2004) estimated likelihood through the harmonic mean of the log likelihoods in the Gibbs sampling iterations after a certain number of "burn-in" iterations.

Perplexity: Describes how well a model predicts a sample, i.e. how "perplexed" or surprised the model is by unseen data. The goal is to minimize this measure. It is calculated as the normalized log-likelihood: $e^{-L/N}$ where L is the log-likelihood and N is the number of words in the data.

Both Scikit-learn and Gensim have implemented methods to estimate these measures.

However, recent studies, such as (Wallach, Murray and Mimno, 2009), have raised concerns about its accuracy. In fact, some have shown that human judgement and likelihood (or equivalently, perplexity) might be slightly anti-correlated, which means optimizing for perplexity may not yield human interpretable topics.[24] Nevertheless, one must note that, for comparison purposes, this method is enough because it correctly ranks models according to quality.

Topic Coherence: This measure surfaced as a response to above limitations, by combining multiple measures to estimate the degree of semantic similarity between high scoring words in each topic and, thus, distinguish between topics that are semantically interpretable and topics that are artifacts of statistical inference. [24] Multiple coherence measures have been defined. According to (Stevens *et al.*, 2012), two have shown to match well with human judgements of topic quality: (1) UCI and (2) UMass, both computing the coherence of a topic as the sum of pairwise distributional similarity scores over the set of topic words. As reported by (Röder, Both and Hinneburg, 2015):

- **UMass**: Asymmetrical confirmation measure between top word pairs (smoothed conditional log-probability). Its summation accounts for the ordering among the top words of a topic. Word probabilities are estimated based on document frequencies of the original documents used for learning the topics.
- **UCI**: Based on pointwise mutual information (PMI). Probabilities are estimated based on word co-occurrence counts derived from documents that are constructed by a sliding window that moves over the corpus**.**

Other measures include:

- **NPMI:** improved version of UCI using a Normalized Pointwise Mutual Information (NPMI) to define context vectors (determined using context windows that contain all words located ±5 tokens around the word);
- **V**: based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity;
- **P:** based on a sliding window, one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence;
- **A:** based on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity.

After calculating individual topic coherence, the model requires an aggregate measure representative of the overall quality of the model. Two aggregate methods can be considered: (1) average coherence and (2) entropy of the coherence (Stevens *et al.*, 2012).

Through the '*coherencemodel'* package, Gensim has provided implementations for the following measures: 'u_mass', 'c_v', 'c_uci', 'c_npmi'.

3 Problem and Solution Description

3.1 Available data and information

Before anything else, it is important to understand the data provided by ToolX as a source of information. Indeed, two different services (and datasets) are provided by this tool: one encompasses manually annotated observations, which shall be used to train and test the classification and topic models; the other comprises automatically annotated comments which will serve as the ultimate comparison between performances of ToolX and the model developed hereafter. The latter includes comments from January and February of 2020, from which 377 comments were impartially annotated, therefore, allowing to compare model and platform performances.

The former includes scraped comments from the beginning of 2019 until the end of January 2020. Reports on sentiment are created every month, thus, the information is divided into 13 files, one for each month. Each file is comprised of the following 17 attributes: Brand, Marca, Date, Time, Content, Sentiment, Origem do Buzz, Temas, Tags, Topics, Author_followers, Author_followings, Author_gender, Author_location, Service, Tipo de Análise, Triggers. Not all are relevant and will be considered for this analysis. Some conceptual clarification is, also, in order so that one may be on a par with the terminology adopted henceforth. Table 2 summarizes the selected attributes along with their newly assigned terminology.

Table 2 - Attribute summarization and respective description.

ATTRIBUTE	TYPE	BRIEF DESCRIPTION	NEW NAME
DATE	date	Date of the content (YYYY-MM-DD).	DATE
BRAND	text	Page from which the content was obtained.	PAGE
SERVICE	text (cat)	Type of media associated with the page: forums, facebook, instagram, twitter.	SERVICE
CONTENT	text	Text of comment or post.	CONTENT
MARCA (brand)	Text (cat)	Brand referred in the content: LIXO (scrap), MEO, NOS, Vodafone.	BRAND
SENTIMENT	Text (cat)	Sentiment associated with the content: NEGATIVE, NEUTRAL, POSITIVE.	SENTIMENT
TEMAS (themes)	Text (cat)	Topic associated with the comment: Apoio ao Cliente (Customer Service), Ativações (Activations), Campanhas e Comunicação (Campaigns and Communication), Fixo (Bundles), Institucional (Institutional), Móvel (Mobile)	TOPIC

Evidently, the most important attributes for this analysis, highlighted in bold, are the **content**, the associated **sentiment** (categorical variable with 3 levels) and the underlying **topic** (categorical variable with 6 levels), given that the software aims to predict sentiment and topic, based on the content.

The combined dataset is composed of 199109 observations divided as follows (Appendix A):

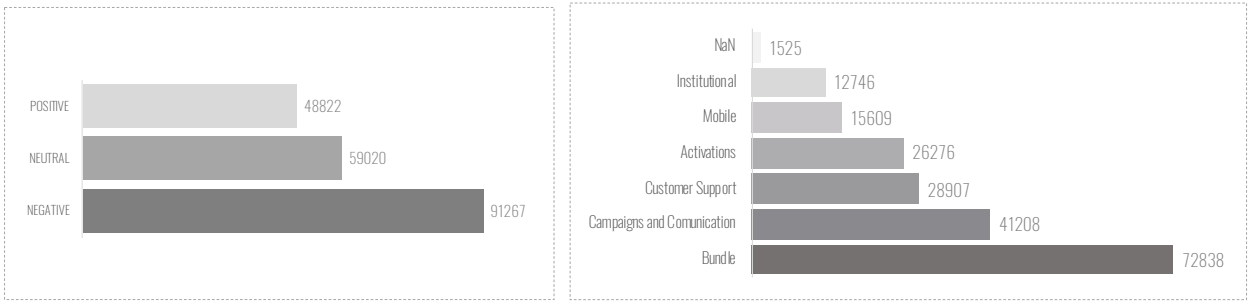


Figure 4 - Distribution of comments according to Sentiment (on the left) and Topic (on the right)

3.1.1 Factor Analysis

To further understand the dataset and underlying information susceptible to be extracted from it, a comprehensive set of questions were defined, and answers were sought through a, purely observational, factor analysis which takes into account the 4 categorical variables (brand, service, topics and sentiment). Defined questions are as follows:

Does each topic reveal a more positive or negative sentiment?

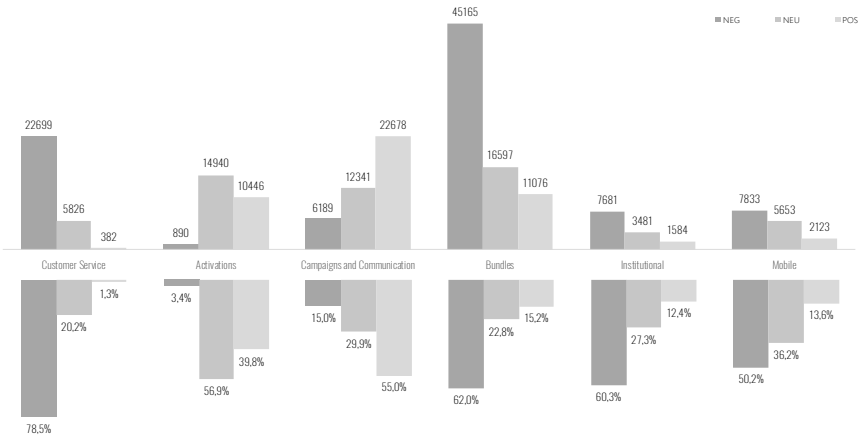


Figure 5 - Sentiment according to topic.

Customer Support: 79% negative

Activations: 57% neutral

Campaigns and Communication: 55% positive

Bundles: 62% negative

Institutional: 60% negative

Mobile: 52% negative

When disregarding neutral opinions, most themes are quite polarizing. Additionally, **Activations** is the only topic that has more neutral comments than any other sentiment. When talking about **Customer Support,** people are exceedingly negative. The same happens to **Bundles**, **Institutional** and **Mobile** (almost to the same extent).

Both **Activations** and **Campaigns and communication** have significantly more positive/neutral than negative comments, which could be expected, as people tend to complain about services, especially customer support, but react positively to campaigns (mostly) and neutrally or positively to activations (at least, recently activated services are usually wanted and the act of activating them isn't a source of polar sentiment).

Are there visible differences between users of different services? Do they tend to be more or less negative?

A quick glance at the plot (Appendix B) allows to conclude that social media and forum users tend to be more negative than positive in their comments, when referring to a telecommunications brand. Instagram is the clear exception. This can be due to the fact that Instagram is not the preferred platform to complain or express a negative feeling towards a brand.

If brand reputation can be measured in percentage of positive comments, how does it vary from brand to brand?

- NOS: 52% negative
- MEO: 44% negative
- Vodafone: 34% neutral

For both **NOS** and **MEO**, most comments are of negative nature. **Vodafone** positively stands out from the other two brands for its lack of polarity in sentiment, in particular, lack of negative sentiment.

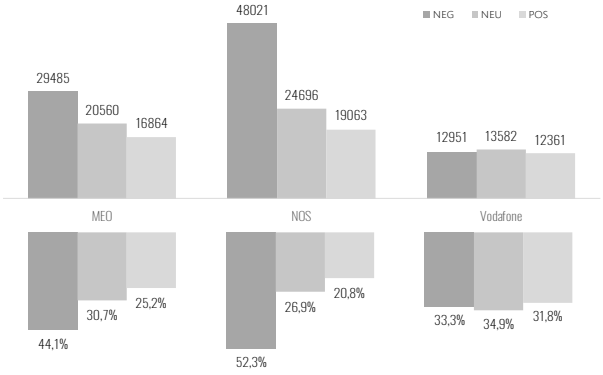
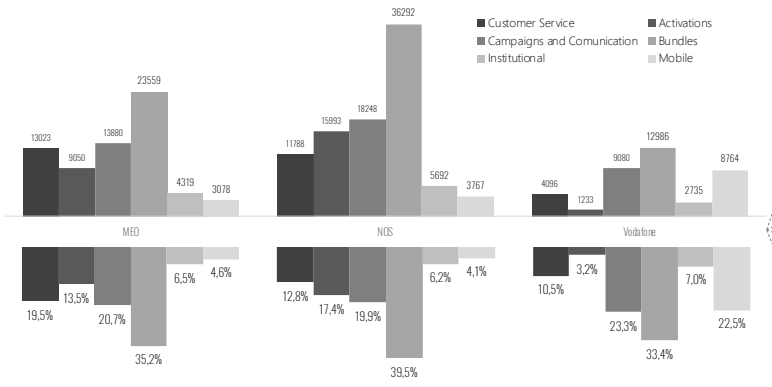


Figure 6 - Sentiment according to Brand

However, is this disparity being indirectly influenced by discrepancies between users of different services? Most comments extracted from Forums are, indeed, **Vodafone**-related (Appendix C). If people tend to be more positive in this type of platform, this could be influencing the percentage of positive comments. The plot interpreted in the previous question (Appendix B), however, annuls that possibility by making clear that forum users tend to make more negative than positive regards. It is, therefore, more acceptable to consider that Vodafone does receive more positive comments than other brands. *So, from which service is this positivity coming from?* Remarkably, all other services, apart from Forums, provide more positive than negative Vodafone-related comments (Appendix D).

What topics are more commonly associated with each brand? And, in those circumstances, which type of sentiment do customers usually project?



BRAND > TOPIC > SENTIMENT					
Brand	Topics	TOT (%)	NEG (%)	NEU (%)	POS (%)
MEO	Customer Service	19%	88%	11%	1%
	Activations	14%	3%	45%	52%
	Campaigns and Comunication	21%	20%	46%	34%
	Bundles	35%	44%	28%	27%
	Institutional	6%	72%	20%	7%
	Mobile	5%	42%	39%	19%
NOS	Customer Service	13%	70%	29%	1%
	Activations	17%	4%	66%	31%
	Campaigns and Comunication	20%	14%	22%	64%
	Bundles	40%	83%	13%	5%
	Institutional	6%	69%	23%	9%
	Mobile	4%	71%	26%	3%
Vodafone	Customer Service	11%	71%	25%	4%
	Activations	3%	1%	32%	67%
	Campaigns and Comunication	23%	10%	22%	68%
	Bundles	33%	36%	41%	23%
	Institutional	7%	24%	48%	28%
	Mobile	23%	44%	40%	16%

Figure 7 - Topics according to Brand

Topics' proportions and respective sentiment can be examined as follows:

- Customer Service: mentioned in relatively similar proportions; strongly negative (88%, 70%, 71%) in all of them.

- Activations: mentioned less frequently in Vodafone-related comments (3%); mostly positive and neutral.
- Campaigns and Communication: similar proportions; mostly positive and neutral comments.
- Bundles: similar proportions; noticeably more negative in NOS (83%), but overall negative tendency.
- Institutional: Similar proportions; strongly negative in MEO and NOS but more neutral and positive in Vodafone.
- Mobile: higher proportion of Vodafone-related comments; overall negative tendency but more so in NOS.

Bundles is, quite visibly, the most relevant topic for all brands, with an overall propensity to the negative sentiment, although the proportion of neutral **Vodafone**-related comments surpasses the negative.

3.2 The Problem - the platform doesn't fulfil the company's needs

It must not come as a surprise that what has been previously adored and criticized about third-party tools entirely applies to ToolX. For the company, this has, recently, meant facing unrequited problems owing to a lack of **flexibility**, **control**, **transparency** and, even, **performance**, of which they do not intend to abdicate.

From a distant, conceptual perspective, two steps are required to obtain a sentiment classification on text:

1. Information cannot be classified if there isn't any in in our possession, in the first place. Thus, the first step is to *scrape the web* for comments that follow specific requirements, in order to store them in a file or database.
2. The second module of the software should have an overall aim to classify the data according to sentiment. This task will be further deconstructed, later on.

In this context, NOS has decided to seize the opportunity of improvement and attempt to increase the performance of the sentiment analysis tasks (the second step). The data described in the previous section will, therefore, serve its purpose as a replacement for the scraping software which will not be developed in the context of this dissertation.

3.3 The Solution - Aspect-based Sentiment Analysis Tool

The project can be, straightforwardly, described as the following:

> *Develop a tool that is capable of: predicting the **Sentiment** and **Topic** associated with a comment or post; achieving a better sentiment classification performance; displaying the results through a **Visualization** platform.*

With that being said, two important concepts arise from the project statement:

- Predicting Sentiment implies a **Sentiment Analysis Problem** (a **classification** problem).
- Predicting Topic implies a **Topic Modelling Problem**, (a **clustering** problem).

System Architecture

A system architecture allows to establish a shared understanding on the system design and supports the evolution and development of an application. The following figure schematizes the proposed architecture.

ToolX's Monthly Excel Files serve as input to the "Initial Preparation of Dataset" python script (described in section 4.1.1) that produces a clean dataset, which will, then, serve as the starting point for the Sentiment Analysis and Topic Modelling tasks. The centrepiece of this tool consists of two Python scripts, one for each of these Data Mining challenges. These scripts can be divided into four essential modules which meet the core KDD process assignments: Preprocessing, Feature Extraction and Selection, Algorithms and Performance Evaluation. These scripts output annotated files, which feed the Power BI platform in order to produce a Dashboard that allows for result visualization and interpretation.

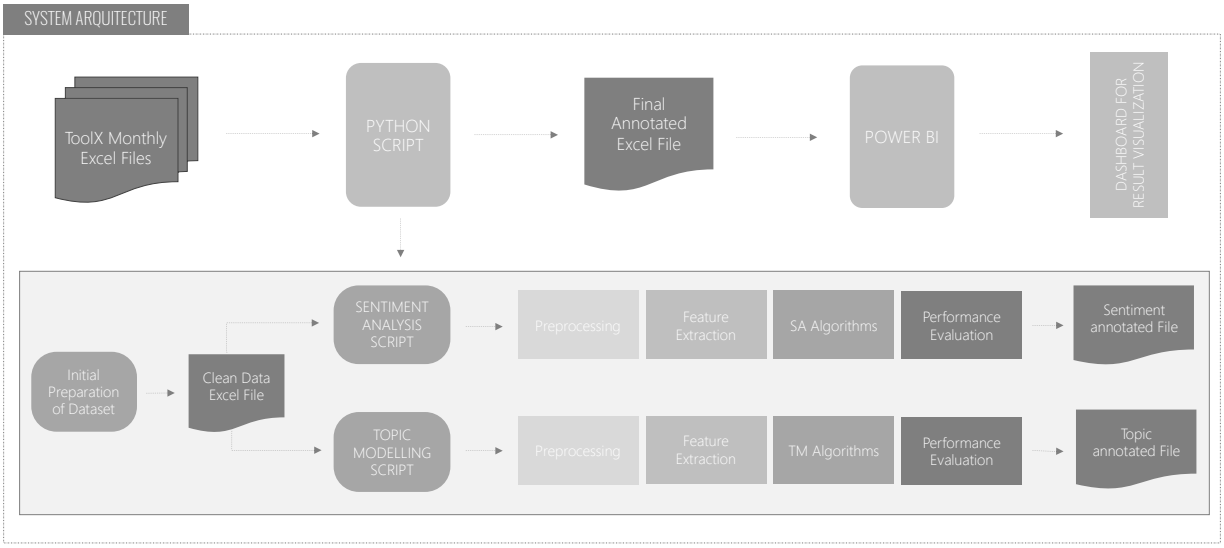


Figure 8 - System Architecture

3.3.1 Methodology

This dissertation was designed to fit the CRISP-DM methodology (Cross-Industry Standard Process for Data Mining) which provides a framework for carrying out data mining projects which is independent from both industry sector and the technology used (Wirth and Hipp, 2000). The methodology tells the story of the data through the following phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment, the latter not being addressed in the present dissertation.

The first step of the CRISP_DM methodology has been tackled in Chapter 1, Section 1.2 "The Project within the company". The second step has been addressed in the previous section (3.1) under the name of "Available data and information. The remaining phases will be explored in the subsequent chapter (4) "Implementation".

4 Implementation

4.1 Data Preparation

4.1.1 Initial Preparation of Dataset

The implementation of simple changes in the dataset, such as changes in attribute names, variable types and category names, addition of new attributes and removal of unnecessary observations, allows to reduce the pointless complexity of the raw data. These changes do not include the pre-processing steps of the 'content' attribute, referring to the extracted comments, because those changes will be separately addressed in the subsequent section. The following steps were performed so that one may obtain the clean document which serves as a foundation for posterior analysis:

Textual Changes:

- Column names were changed according to the information displayed in Table 2.
- All textual attributes (except 'content') were lowercased.
- Part of the 'page' attribute text (bolded in the following example) was shared by all observations and, therefore, superfluous: e.g. **noselife2016gma**_marcas_jovens. Thus, only the second part was selected after splitting the text by the first underscore.
- All accents and atypical characters were removed from categorical variables, using the *unidecode* method.

Data Type Changes:

- Categorical variables were categorized ('brand', 'topics', 'sentiment', 'service').
- The 'date' attribute data type was changed to date and only the YYYY-MM-DD information was kept. The dataset was sorted according to date of post or comment publication.

Row elimination:

- Observations with 'content' equal to NaN were removed, as they are irrelevant for this analysis.
- There were several duplicated comments, mostly due to retweet extraction. Although these are not irrelevant, because someone who retweeted a post is, usually, in agreement with its author, they increase the amount of data that needs to be analysed. In order to keep the information but reduce the number of dataset rows, a new attribute with the number of retweets of each tweet was created:
  - The 'RT' indicator in the beginning of retweet comments was removed;
  - A new array was created with all duplicated 'content';
  - Duplicates were counted and a new dataframe - 'checked' - with the following columns was created: 'comment', 'duplicates_count', 'checked';

- The original dataframe was updated according to the information provided by the 'checked' dataframe. All original comments were kept and associated with the corresponding 'duplicate_count'. All repeated comments were eliminated.

4.1.2 Text Preprocessing

After obtaining the backbone dataset of this tool, some important acknowledgements must be addressed regarding the utmost importance of the next few steps in the overall picture of the analysis.

According to (Gurusamy and Kannan, 2014), preprocessing tasks are essential in NLP for the following reasons:

To reduce indexing (or data) file size of the Text documents:

- Stop words accounts 20-30% of total word counts in a particular text document.
- Stemming may reduce indexing size as much as 40-50%.

To improve the efficiency and effectiveness of the NLP tools:

- Stop words are not useful for Text mining and may confuse the algorithms
- Stemming is used for matching the similar words in a text document and, thereby, reducing the dimensionality of the data

Furthermore, one may consider the importance of removing additional noise in the form of unnecessary characters, URLs, html tags or even specific social media constructs. Certainly, this noise won't provide any supplementary benefit to the analysis and, instead, may bedevil the algorithms by incorrectly skewing the results. Besides, most Data Mining models do not accept text as input and, consequently, some preprocessing step must be implemented to map unorganized text into a series of tokens which will later be converted to a numerical format.

That being said, it is difficult to predict how a certain preprocessing step will influence the model performance, hence, multiple combinations have been considered and tested, although some steps were applied to every possibility. As a final regard, please note that two separate pipeline functions have been created: the first gathers all text cleaning tasks, from striping html tags to removing extra whitespace; the second performs all annotation and normalization tasks, from tokenization to lemmatization.

The first function requires the user to decide if hashtags, punctuation and emojis should be removed, which means combinations of these possibilities will appear. The other steps were set to always be applied because they were not considered to have a possible improvement effect on the performance of the sentiment analysis models, i.e. in this context, only hashtags, punctuation and emojis can provide fruitful information on the sentiment of the post. The following pseudo-code describes the cleaning stage of the selected preprocessing pipeline:

For each line in the 'content' column:

- **Remove html tags and other unnecessary characters:** Some extracted responses from a Vodafone-related forum had an 'Re' indicator in the beginning of the comment which had to be removed. Multiple observations also had html tags (in a '<tag>' format) which presented no benefit to this analysis.
- **Remove mentions**: in this particular analysis, mentions will not be considered, although some utility could be found in acquiring information on specific users who have relevant patterns of behaviour in social media.
- **If 'hashtags' is set to True, remove hashtags**:

- **Remove URLs:** there is no added value in considering an URL for classifying a comment on its sentiment.
- **If 'punctuation' is set to True, remove Punctuation:** there is a strong correlation between the use of certain punctuation marks and sentiment which can aid sentiment classification. Besides, it is also common to use multiple punctuation marks in a row to elevate the strength of an opinion.
- **Remove Numbers**: Numbers have little to no relevance in terms of sentiment classification.
- **If 'emojis' is set to True, remove emojis;**
- **If 'emojis' is set to False**, convert emojis to text: Using the *demojize* package
- **Lowercase text:** Lowercasing text reduces dimensionality. However, it may also be helpful in sentiment classification, as full capitalized words or sentences are commonly associated with strong sentiment.
- **Remove extra new lines, and whitespaces**

All previous steps involved custom-made functions to fit the dataset requirements.

After going through with these steps, the resulting text must be annotated and normalized according to what is considered appropriate. To do so, there are several well-rounded open libraries, such as NLTK and SpaCy. For multiple reasons, the second was deemed as a better fit for this problem. According to (Kakarla Swaathi, no date),"while NLTK provides access to many algorithms to get something done, SpaCy provides the best way to do it. It provides the fastest and most accurate syntactic analysis of any NLP library released to date. It also offers access to larger word vectors that are easier to customize."

After the text cleaning stages, SpaCy docs must be created in order to extract the desired information. For this purpose, SpaCy offers pretrained statistical models for a multitude of languages. The Portuguese library applied in this analysis is referred to as *'pt_core_news_sm'.* In the second step of the preprocessing pipeline, the following tasks were completed:

- Tokenization
- Stopword removal: after analysing the proposed stopword list, the word 'não' (Portuguese word for 'not') was removed from this list, as negations are of utter importance for sentiment analysis, for being able to completely change the meaning of a sentence.
- Lemmatization: a new column was created with the lemmatized tokens. Note that SpaCy does not provide the possibility of using stemming techniques. Nevertheless, lemmatization was the preferred method because it withholds important morphological information that stemming methods do not.
- Part-of-Speech Tagging: after tagging each token with the appropriate Part of Speech, tokens were filtered so that only Nouns, Verbs, Adjectives and Adverbs were kept for future analysis. A new column was also created for these lemmatized, filtered tokens.
- Named Entity Recognition

Preprocessing combinations can be found in Table 3.

Running times had very little to no fluctuation between preprocessing combinations, none having exceeded the 20-minute mark. The most time-consuming task was transforming SpaCy docs into lists to be used in the subsequent preprocessing techniques, namely lemmatization and part of speech tagging and selection.

With the intent of evaluating the impact of opting for each of the 8 text cleaning combinations above, the AUC-ROC of a Logistic Regression Model with default parameters was calculated.

After electing the best alternative, the same principle was followed to differentiate between the other 3 possibilities referring to Lemmatization and Part of Speech tagging. These three alternatives could be found in three different columns: the first having the original tokens, the second having the lemmatized tokens and the third having the lemmatized tokens filtered according to part of speech.

Table 3 - Preprocessing combinations

No	Remove emojis	Remove punctuation	Remove hashtags
1	No	No	No
2	Yes	No	No
3	No	Yes	No
4	No	No	Yes
5	Yes	Yes	No
6	Yes	No	Yes
7	No	Yes	Yes
8	Yes	Yes	Yes

No	Lemmatization	POS
1	No	No
2	Yes	No
3	Yes	Yes

Some occasional changes were made to this processing pipeline in order to adapt it to the topic modelling needs and requirements. The overall complexity introduced by the multiple preprocessing combinations was eliminated and only one preprocessing possibility was considered. Therefore, emojis, punctuation and hashtags were removed from all text, and tokens were lemmatized and filtered according to part of speech tags. Moreover, some NaN values were reported in the 'Topics' attribute. These instances referred to occasions where the 'Brand' attribute had the value 'lixo' (garbage in Portuguese). Some interest was manifested in enriching the model with the ability to identify comments with no significant value to the analysis. Therefore, the NaN instances were substituted by 'no_value' to allow the clustering techniques to treat them as another topic.

4.1.3 Feature Extraction and Selection

Feature extraction plays a prominent role in sentiment analysis. In fact, extracting informative and essential features greatly enhances the performance of machine learning models and reduces the computational complexity (Avinash and Sivasankar, 2019)**.** Most machine learning algorithms require numeric input data. One can easily come to the conclusion that the textual input data we are in possession of doesn't fit this criterion and must, therefore, be vectorized, i.e. each token must be represented numerically (Sousa, 2019).

Vectorization (Document Representation Strategy): Note that, because there are multiple ways to numerically represent a textual document, not all vectorization methods will achieve the same performance. The most popular alternatives are the **Bag of Words** and the **N-Gram** representations (Samuel and Coelho, 2013). Due to the high dimensionality of the vocabulary incorporated into the textual contents, a BoW vector is noticeably sparse (Zhang, Wang and Liu, 2018). Additional remarks dwell on the word order that is inconveniently neglected, which means semantics cannot be encoded and, as long as two documents use the same vocabulary, they will also share the same BoW representation. N-Grams can consider the

word order in a short context, but also suffers from data sparsity and high dimensionality (Zhang, Wang and Liu, 2018).

Taking all these clarifications into account, due to its popularity and simplicity, the n-grams methodology was selected in combination with TF-IDF calculations. N was set to range from 1 to 3 so than one may take advantage of some word context without overly inflating the dimensionality of the feature vectors. This was accomplished by applying the **TfidfVectorizer**, which performs both procedures at once.

Feature Selection: Sparsity can become a real obstacle to classification techniques by unnecessarily increasing running time. Besides, there is no use in keeping uncommon tokens that do not allow to distinguish between classes (Fortuna, 2017)**.** Concurrently to the vectorization, a study on the dimensionality of the dataset was performed. It became clear that we were dealing with highly dimensional data with low frequencies. In fact, when defining a minimum ngram frequency of 0.5%, only 267 features were selected from which all were unigrams. A maximum number of features was, instead, set to 1000 to keep the dimensionality under control

Dimensionality Analysis: When comparing the number of observations before and after the Initial Preparation of the Dataset, there was a reduction of 50304 (199109 to 148805) observations, which was further amplified to 80065 (199109 to 119044), after the Preprocessing tasks. This corresponds to a reduction of 40.2%. The dimensionality of the dataset was greatly reduced when the feature selection was performed. The initial number of features was reduced to 1000 because of the established maximum number of features in the *tf-idf vectorizer*.

4.1.4 Train, Validation and Test Set Division

In order to test the produced models, it is critical to divide the original dataset into a train and test set. A traditional 80/20 division was adopted, where 80% of the observations are allocated to the train set and the remaining 20% are allocated to the test set. This division aimed to preserve the chronological nature of the social media scraping, as comments that will be extracted in the future will be classified according to models trained with previous data. The training set resulted in a total of 119044 observations, from which 28366 (23%) refer to positive comments, 36940 (31%) to neutral and 53738 (45%) to negative. The test set is composed by 29761 observations, 6092 positive, 8544 neutral and 15125 negative. The train dataset was not considered imbalanced to the point where pre-model balancing techniques are required. Instead, besides including ensemble methods in the analysis, a cost sensitive learning approach was implemented by setting the models' parameter '*class_weight'* to '*balanced'*.

Another division is also mandatory in instances where parameters need to be tuned: the training set must be further split into a *training subset* and a *validation set*, so that the model can be trained on the training subset and the parameters can be chosen according to its performance on the validation set. Subsequently, the model is trained on the full training set using the chosen parameters, and the error on the test set is recorded (Cochrane Courtney, no date).

This division into train and validation sets can be accomplished according to various different methods. Cross-validation (CV) is a popular technique for tuning hyperparameters. Two of the most common types of cross-validation are *k*-fold cross-validation and hold-out cross-validation. However, when dealing with time series data, traditional cross-validation (like k-fold) should not be used because there is a need to simulate the "real world forecasting environment" (Tashman, 2000) and preserve temporal dependencies. A possible solution is to use a Time Series Split (Appendix E), which is a variation of the k-fold method provided by

Scikit-learn which, instead of always considering the complete train set to extract a train subset and validation set on each fold, in the kth split, it returns the first k folds as a train set and the (k+1)th fold as a validation set. Scikit-learn also notes that unlike standard cross-validation methods, successive training sets are supersets of those that come before them. To further elaborate on this procedure, consider k folds and n observations:

1. On the first iteration, the method selects the first n/k observations (first fold) to be the train subset and the second n/k observations (second fold) to be the validation set; the model is trained and tested along those lines.
2. For the second iteration, the previously selected train and validation sets (first and second folds) become the new train subset and the third fold is indicated to be the validation set.
3. This procedure is replicated until the last fold is reached.

For the stated reasons, a Time Series Split with 10 folds was considered appropriate for tuning hyperparameters of the selected models. The pipeline on Figure 9 summarizes the selected approach to deal with the original dataset and create train, validation and test sets.

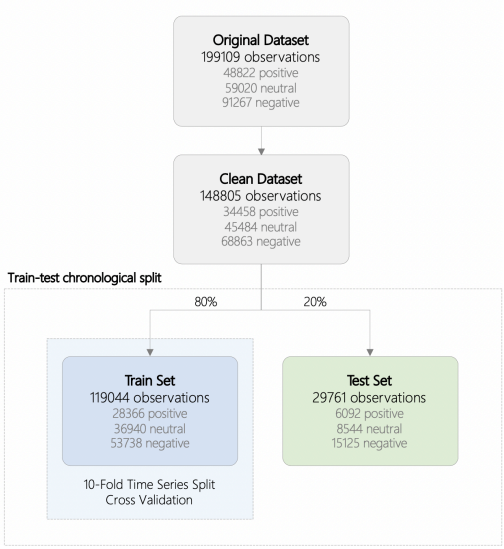


Figure 9 - Subdivision of Datasets

4.2 Modelling

4.2.1 Sentiment Analysis Algorithms

From the previously stated classifiers (State of the Art) the following were selected: Logistic Regression, Decision Tree, Naive Bayes, SVM, Random Forest, Gradient Boosting, MLP. All algorithms were obtained from Scikit-learn's free software machine learning library for Python, apart from the XGBClassifier, which was obtained from the *xgboost* package. This selection was based on interpretability, engine efficiency and ability to train large and highly dimensional datasets.

According to the defined methodology, hyperparameters must be tuned on the train and validation sets in order to maximize performance. This section aims to elucidate the reader on the elected combinations of parameters for each model. As previously mentioned, parameter tuning was performed using a Time Series Split methodology with GridSearch. The following parameter descriptions can be found in Scikit-learn and the XGBoost website. Only parameters implicated in the algorithm have been described Inside the curly braces which succeed the parameter name, the reader can find the option(s) used in the implementation. For all models, *random_state* was set to 22, in addition to *class_weight*, which was set to 'balanced' in all models that offer and require this feature to deal with unbalanced data. Neither Naïve Bayes nor Gradient Boosting need this feature, the first because of its conceptual nature, and the latter because of its implementation which deals with class imbalance by constructing successive training sets based on incorrectly classified examples.

Logistic Regression: *solver* {saga} - Algorithm to use in the optimization problem; *penalty* {'l1', 'l2', 'elasticnet'} - Used to specify the norm used in the penalization; *C* {0.01, 0.1, 0.5, 1, 5}- Inverse of regularization strength, smaller values specify stronger regularization.

Naïve Bayes: *alpha* {1, 0.5, 1e-1, 1e-2}- Additive (Laplace/Lidstone) smoothing parameter.

Decision Tree: *max_depth* {10, 20, 30, 50}- The maximum depth of the tree

SVM: *C* {0.1, 1, 10} - Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty; *kernel* {linear} - Specifies the kernel type to be used in the algorithm; *probability* {True} - enables probability estimates.

Random Forest: *max_depth* {10, 20, 30} - The maximum depth of each tree; *n_estimators* {100, 500, 1000} - The number of trees in the forest; *min_samples_leaf* {1, 5, 10} - minimum number of samples required to be at a leaf node.

Gradient Boosting: *max_depth* {10, 20, 30} - The maximum depth of each tree; *n_estimators* {100, 500} - The number of trees in the forest; *min_child_weight* {1, 5, 10} - Minimum sum of instance weight (hessian) needed in a child.

MLP: *hidden_layer_sizes* {(100, 50), (100,)} - The ith element represents the number of neurons in the ith hidden layer; *activation* {tanh, relu} - Activation function for the hidden layer; *alpha* {0.0001, 0.05} - L2 penalty (regularization term) parameter. According to (Huang, 2003), the optimal number of hidden nodes in the first hidden layer is $\sqrt{(m+2)N}+2\sqrt{N/(m+2)}$ and, in the second layer, it is $m\sqrt{N/(m+2)}$, where m is the number of inputs and N is the number of outputs. In this particular case, m is equal to 1000 (number of features) and N is equal to 3 (number of classes), therefore, besides the default setting of *hidden_layer_sizes*, another possibility was calculated according to this rule of thumb: (98, 42.43) approximately becomes (100, 50).

Table 4 - Summarization of Models and respective parameters considered for further analysis.

MODEL	PACKAGE	PARAMETER	VALUES
Logistic Regression	LogisticRegression	penalty	L1, L2, elasticnet
		C	0.01, 0.1, 0.5, 1, 5
Naïve Bayes	MultinomialNB	alpha	1, 0.5, 1e-1, 1e-2
Decision Tree	DecisionTreeClassifier	max_depth	10, 20, 30, 50
SVM	SVC	C	0.1, 1, 10
Random Forest	RandomForestClassifier	max_depth	10, 20, 30
		n_estimators	100, 500, 1000
		min_samples_leaf	1, 5, 10
Gradient Boosting	XGBClassifier	max_depth	10, 20, 30
		n_estimators	100, 500
		min_child_weight	1, 5, 10
MLP	MLPClassifier	hidden_layer_sizes	(100, 50), (100,)
		activation	tanh, relu
		alpha	0.0001, 0.05

Because most metrics for performance evaluation assume a balanced dataset, the ability to deal with this type of situation was seen as a criterion to choose an appropriate metric, which led to the decision of using AUC-ROC (Area Under the ROC Curve).

A ROC curve (receiver operating characteristic curve) displays the performance of a classification model at all thresholds, according to its True Positive Rate or Recall (True Positives / all Positive observations) and False Positive Rate (False Positives / all Negative observations). AUC measures the entire two-dimensional area underneath the entire ROC curve, therefore, providing an aggregate measure of performance across all possible classification thresholds. AUC is advantageous for the following reasons:

- It measures how well predictions are ranked, rather than their absolute values.
- It measures the quality of the model's predictions irrespective of what classification threshold is chosen.
- It reflects class imbalance in the sense that it takes into account falsely classified observations instead of only looking at the True Positives or True Negatives.

It is also important to remember that the present tool is dealing with a multi-class problem (3 classes) Thus, heuristics are used to split the multi-class classification problem into multiple binary classification problems. According to (Murphy, 2012) , the obvious choice is to use a one-versus-rest (OvR) approach (also called one-vs-all). A binary classifier is trained on each binary classification problem and predictions are made using the model that is the most confident. Also note that Scikit-learn models implement the OvR strategy by default when using these algorithms for multi-class classification.

In short, this dissertation uses the AUC-ROC in conjunction with an OvR methodology to assess model performance.

Because there is no preestablished method to visualize the average ROC Curve for multiclass classification problems, a short function was implemented to enable this visualization. The algorithm computed the ROC Curve and AUC for each class, which served as a foundation for computing the micro-average ROC Curve and AUC.

4.2.2 Topic Modelling Algorithms

To model the topics associated with each observation, two algorithms were put head to head: Latent Dirichlet Allocation and GSDMM. The first was provided by Gensim (LdaMulticore). The latter used a modified implementation of the algorithm found in (Rwalk, no date).

It is important to mention that, although a clustering approach was taken to obtain a set of topics and assign them to the observations, the original dataset already provides a set of manually assigned topics according to a defined list (see ‘Topics’ Attribute in Data Understanding). The reasoning behind choosing a clustering approach instead of classification is explained by the volatile nature of topics, which frequently change over time. An unsupervised clustering approach provides the tool with the necessary flexibility to deal with this volatility. It allows to automatically identify introduction of new brand-related topics on social media and forums. Nevertheless, the ‘Topics’ Attribute provides a foundation for calculating model performance as if we were dealing with a classification problem. It also creates a possibility to automatically assign topic names to the generic topics obtained by the LDA and GSDMM models, as will be subsequently explained.

Both were implemented in three different steps, each corresponding to a different customized function:

1. Train the model on the train set;
2. Assign a topic to each observation according to probability distribution;
3. Assign an appropriate name to each topic in a way that maximizes performance:

For each topic in the list of predicted topics:

- select all rows from the dataset which correspond to that unnamed predicted topic;
- calculate what is the most frequent 'true' topic ('Topics' attribute) to be assigned to those observations;
- change the name of the unnamed topic to the name of the most frequent 'true' topic.

Because of their conceptual similarities, they share the same parameters.

Performance Evaluation: As previously mentioned, topic modelling evaluation methods fall into these categories: Eye Balling Models, Intrinsic Evaluation Metrics, Human Judgements, Extrinsic Evaluation Metrics. The existence of a 'Topics' attribute opens the door to using Extrinsic Evaluation Metrics such as the ones used in classification problems. A simple accuracy score was used to compare LDA to GSDMM. Moreover, topic coherence, an intrinsic evaluation metric, was also nominated to quantitatively assess the quality of the clustering performed and to justify the model selection. UMass was used as a coherence measure. In regard to LDA, this metric was implemented by Gensim in the package '*coherencemodel'*, which uses arithmetic mean as the aggregation method. An equivalent function was created from scratch to calculate the same metric for the GSDMM model, using a word vector with the top 10 words for each topic.

Cohen's kappa score: Besides comparing the accuracy-based performance, a Cohen's kappa score computation (cohen_kappa_score on Scikit-Learn) allowed to infer how often these models agree on their topic classification. Cohen's kappa is a statistic that measures inter-annotator agreement, i.e. the agreement between two models who each classify N items into C mutually exclusive categories (*Cohen's kappa - Wikipedia*, no date). It is defined as $\kappa = (p_0 - p_e)/(1 - p_e)$ where p_0 is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and p_e is the expected agreement when both annotators assign labels randomly. p_e is estimated using a per-annotator empirical prior over the class labels. The result varies between -1 and 1. The maximum value is associated with a complete agreement between models, a score of 0 implies a random agreement and score lower than 0 means that there is less agreement than chance (*Cohen's Kappa - Towards Data Science*, no date).

Stacking: Aiming to improve the performance of the topic modelling task, a Stacking methodology using a Logistic Regression was implemented (Figure 10).

Stacking is a tool which befits the ensemble method sphere. In their traditional demeanour, ensemble methods are used to boost predictive accuracy by combining the individual predictions of a set of classifiers (typically by voting) (Džeroski and Ženko, 2004). Model stacking is an ensemble method which uses a second-layer learning algorithm that optimally combines the predictions of the first-layer algorithms. It produces a new and improved set of predictions and, therefore, offsets the weaknesses and biases of some with the strengths of others (*Why do stacked ensemble models win data science competitions? - The SAS Data Science Blog*, no date). Stacking can be considered a meta-learning approach. According to (Džeroski and Ženko, 2004), the following meta-learning tasks can be contemplated: learning to select an appropriate learner, learning to dynamically select an appropriate bias, and learning to combine predictions of base-level classifiers.

Figure 10 - Model stacking schema.

4.3 Visualization

One of the most important steps, when dealing with any type of data, is data visualization, which can be defined as the act of taking information and placing it into a visual context, such as a map or graph (*Data Visualization: What It Is, Why It's Important & How to Use It for SEO*, no date).

These visualizations can simplify what is sometimes too complex for the human brain to understand, besides opening doors to pattern and trend identification. "Good data visualizations should place meaning into complicated datasets so that their message is clear and concise."

Having that in mind, it was mandatory to select a platform which could deliver such possibilities. **Microsoft Power BI** is an intuitive business intelligence platform that combines tools for aggregating, analysing, visualizing and sharing data. In the company's context, it made sense to take advantage of this technology to fulfil the visualization needs of the sentiment analysis and topic modelling tasks performed.

5 Results

5.1 Sentiment Analysis

Aiming to evaluate how different preprocessing possibilities would affect the overall performance of the model, 8 different combinations were compared based on the AUC obtained through application of a Logistic Regression model with default parameters. Table 5 highlights these variations in performance. Note that, when emojis are not removed, they are still converted to text.

Table 5 - Performance Evaluation of Data Cleaning Combinations

No	Remove emojis	Remove punctuation	Remove hashtags	RUN TIME	AUC on CV
1	No	No	No	8.68 s	81.58%
2	Yes	No	No	8.66 s	81.70%
3	No	Yes	No	9.02 s	81.57%
4	No	No	Yes	8.69 s	81.58%
5	Yes	Yes	No	8.70 s	81.70%
6	Yes	No	Yes	8.61 s	81.69%
7	No	Yes	Yes	8.70 s	81.60%
8	Yes	Yes	Yes	8.73 s	81.68%

Two cleaning combinations achieved an equal average AUC performance level on the validation sets: 2 and 5. The former, however, was able to perform slightly faster than the latter, which was enough of a reason to choose a preprocessing combination where only emojis are removed from comments. When applying these preprocessing steps on the test set, an AUC of 73.91% as obtained.

Regarding Lemmatization and Part of Speech tagging, 3 different possibilities were considered. One must remember that, if POS tagging is selected, only tokens tagged as a Noun, Verb, Adjective or Adverb are kept for classification purposes. Once again, a Logistic Regression with default parameters was implemented in order to compare AUC performance results, which can be observed in Table 6.

Table 6 - Performance Evaluation of Annotation and Normalization Combinations

No	Lemmatization	POS	RUN TIME	AUC on CV
1	No	No	9.47 s	82.64%
2	Yes	No	9.59 s	83.23%
3	Yes	Yes	8.63 s	81.70%

A superior performance (83,23%) was achieved when tokens were lemmatized, but all parts of speech were considered. This technique allowed to achieve an AUC of 74.87% on the test set. In a nutshell, the preferred preprocessing techniques generated lemmatized tokens which do not include emojis, but keep all punctuation, hashtags and parts of speech.

All that was left to determine was the ultimate classification model and respective parameters to be used for the succeeding sentiment analysis necessities.

Table 7 - Performance Evaluation of Classification Models and respective optimal parameters

MODEL	BEST PARAMETER COMBINATION	RUN TIME	AUC on CV
Logistic Regression	C: 0.5, penalty: l2	896.64 s	83.28%
Naïve Bayes	alpha: 0.5	5.40 s	82.52%
Decision Tree	max_depth: 50	288.10 s	71.91%
SVM	C: 1	82413 s	83.10%
Random Forest	max_depth: 100, n_estimators: 1000, min_sample_leaf: 1	9591.62 s	82.36%
MLP	hidden_layer_sizes: (100,), activation: relu, alpha: 0.05	14346 s	83.41%

In accordance with the presented AUC results, the best average performance on the validation sets (83.41%) was obtained when applying the MLP model with the following parameters: {hidden_layer_sizes: (100,), activation: relu, alpha: 0.05}. The same model obtained a performance of 74.84% on the test set. The ROC Curve for each class, as well as the aggregated curve, were plotted for each model. Figure 11 portrays this visualization. The ROC Curve visually codifies the trade-off between sensitivity and specificity. It is evident that the class with the worst associated results is class 1, representative of the neutral sentiment. A better performance was achieved when classifying comments as negative (class 0). Lack of sentiment, i.e. neutrality, is, indeed, a difficult class to predict, as multiple linguistic subtleties need to be taken into account.

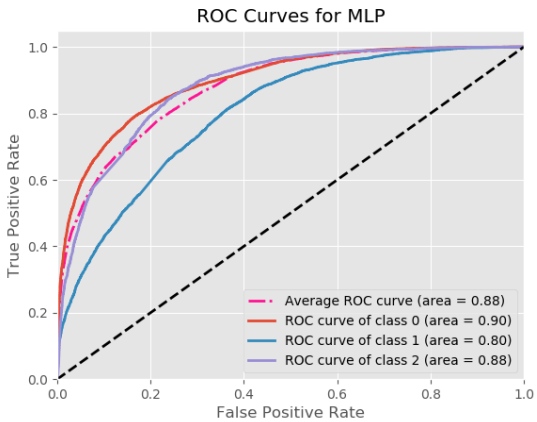


Figure 11 - ROC Curve for MLP

What has been previously analysed regards comparisons between model-predicted sentiment labels and manual annotations provided by ToolX.

As mentioned in Section 3.1 – Available Data and Information (p.24), to establish a comparison between the produced MLP model and the automatic tool provided by the same platform (ToolX), another dataset was made available. When juxtaposing these sentiment classifications on the 377 manually annotated comments, it was concluded that ToolX's AUC-ROC performance was 68.07% and MLP's performance was 60.43%. However, after removing 91 worthless observations which added no value to the analysis, MLP's performance increased to 65.25% and ToolX's performance decreased to 66.77%. It can be deduced that MLP's performance increased with the quality of data and, therefore, the scraping task is of utmost importance, being able to greatly influence the results.

These performances are similar in figure. Nonetheless, an in-house tool presents multiple advantages which must be considered in this context. In fact, more transparency and flexibility can be achieved with an internally developed solution that has so much space to grow in complexity and purpose.

5.2 Topic Modelling

The performances of LDA and GSDMM have been compared according to accuracy score and 'UMass' topic coherence and can be contemplated in the Table 8.

Table 8 - Performance Evaluation of Topic Modelling Techniques

MODEL	PARAMETERS	ACCURACY ON TEST	COHERENCE
LDA	passes: 10; num_topics: 10; alpha: 0.01; beta: 0.01	28.00%	-3.79
GSDMM	passes: 10; num_topics: 10; alpha: 0.01; beta: 0.01	45.89%	-2,62
Stacking	Default Logistic Regression Parameters	38.33%	---

The best performance on the train set (45.89%) was obtained by GSDMM, as expected, given that it this model has been optimized for Short Text Topic Modelling problems. Regarding the UMass Topic Coherence measure, this method also outperformed LDA (-2,62).

The computation of the Cohen's kappa score allowed to infer how often these models agree on their topic classification. Train agreement was 0.095 and Test agreement was 0.146. Because this agreement was particularly low, a stacking technique could improve the results of both models by learning from each model's strengths. The result, however, reveals that this did not happen.

Aiming to gain a deeper understanding on the classification performed by the above topic models, the following confusion matrixes were created (Table 9 and Table 10). Highlighted in green are the true topics which had the right maximum prediction equivalent. In orange, one can find topics which were mostly wrongly predicted.

According to LDA (Table 9), no comments were attributed to the Activations, Institutional and No_value topics. Most comments were associated with Campaigns and communications, Customer Service and Bundles, which is and anticipated behaviour. In reality, there is a high concentration of comments regarding these topics because they are usually reflected in rather polarizing sentiment.

It is also important to highlight that these particular topics were mostly well predicted. Activations, Institutional and No_value (garbage) were all associated with Campaigns and Communication. The first was also commonly associated with Bundles, which reflects the interchangeable property between these topics.

Table 9 - Confusion Matrix for the LDA Model

LDA		PREDICTIONS						
		Mobile	Bundles	Campaigns and communication	Customer Service	Activations	Institutional	No_value
TRUE TOPICS	Mobile	398	547	432	776	0	0	0
	Bundles	238	2511	2444	2042	0	0	0
	Campaigns and communication	887	1674	4193	1194	0	0	0
	Customer Service	403	474	895	5139	0	0	0
	Activations	94	1097	1283	207	0	0	0
	Institutional	104	697	1086	669	0	0	0
	No_value	4	94	134	45	0	0	0

Table 10 – Confusion Matrix for the GSDMM Model

GSDMM		PREDICTIONS						
		Mobile	Bundles	Campaigns and communication	Customer Service	Activations	Institutional	No_value
TRUE TOPICS	Mobile	39	1416	387	288	23	0	0
	Bundles	52	4030	1503	947	703	0	0
	Campaigns and communication	92	2274	4461	705	416	0	0
	Customer Service	32	1805	837	4202	35	0	0
	Activations	18	240	1416	71	933	0	0
	Institutional	63	1362	522	546	63	0	0
	No_value	3	63	163	27	21	0	0

According to GSDMM (Table 10), no comments were associated with Institutional and No_value topics. This model was, indeed, better at predicting activation-related comments (933), although most of them were still associated with Campaigns and Communication. The same happened with No_value comments. Institutional, however, was mostly associated with Bundles. Mobile also has a very high interaction with the Bundles topic. In fact, both topics are usually associated with each other because multiple customers have Mobile expenses associated with Bundles.

The overall performance of this model was superior. Nonetheless, LDA was able to predict Mobile-related comments visibly better (398 vs 39) and Customer Service-related comments slightly better (5139 vs 4202).

5.3 Visualization of Results in Power BI

The following Dashboard (Figure 12) was produced in order to properly visualize the results of the Sentiment Analysis and Topic Modelling tasks. It is highly interactive and allows to filter results according to Sentiment, Brand, Themes and Time Horizon. Permanent plots range from two bar plots: the first allows to visualize number of comments of each sentiment according to Brand; the second allows to analyse the number of comments of each sentiment per Topic. The tree diagram on the right summarizes the total number of comments of each sentiment and the chronological plot displays the evolution of sentiment throughout time.

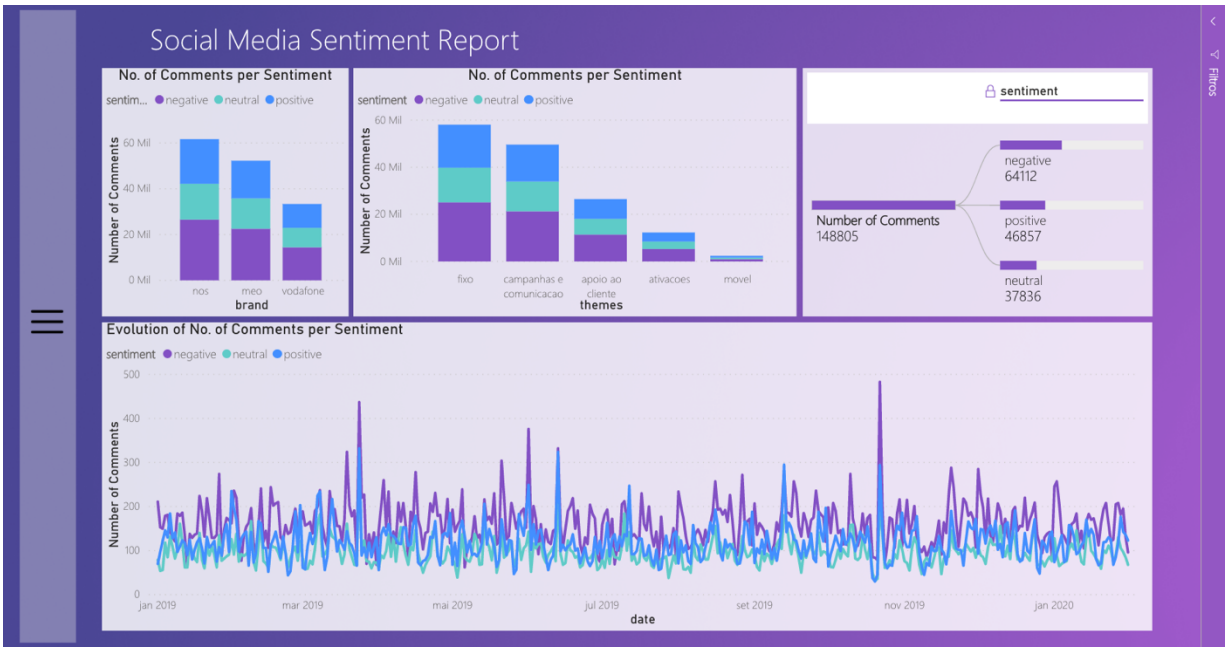


Figure 12 - Power BI Dashboard

6 Conclusion

As an overall Sentiment Analysis tool, the present project has potential to fulfil the initially defined objective. As with any project worth working for, multiple challenges had to be overcome. Ranging from conceptual to technical trials, text mining requires a broad understanding of the data and its capabilities. Indeed, with no prior knowledge of the Python language and a superficial and limited knowledge of Text Mining notions, a great deal of new concepts and techniques had to be acquired in order to achieve what was initially intended. The author tried to follow a methodical approach when deciding between preprocessing, feature extraction and modelling alternatives. By applying classification and clustering techniques, the dissertation's variety and complexity increased and so did the challenges. However, this created an opportunity to expand personal knowledge on an area many consider to be essential in the modern day of a company, whatever sector it belongs to. In a nutshell, in regard to the sentiment analysis classifiers, an AUC-ROC of 74,84% was achieved when applying the MLP model with the following optimal parameters {hidden_layer_sizes: (100,), activation: relu, alpha: 0.05} on a lemmatized test set stripped of emojis. Neutrality was the most difficult sentiment to model. The best topic models were created when employing a GSDMM model, as could be expected due to its superior ability to deal with short text. Its performance on the test set was 45.89% and UMass Topic Coherence was -2,62. Although a stacking approach was considered promising because of the low agreement between models, results did not improve accordingly. Further research should be carried out along these lines, in order to improve the accuracy and coherence of these alternatives. The AUC-ROC achieved by ToolX's automatic sentiment classification (66.77%) and the one achieved by MLP (65.25%), were, indeed, rather comparable. The latter, however, provides more flexibility and transparency to the company.

The project comes to an end. The author, however, hopes that this analysis will give place to an array of future possibilities, which work on expanding the capacities of the developed tool.

On a technical context, a scraping software would be of immense benefit to the company and would allow to complete the necessary tasks that collectively represent a Social Listening Tool. Multiple other models could be implemented, from which Deep Learning models should take centre stage. The truth is the Machine Learning world is in full exponential expansion and every day is an opportunity to achieve better performances. Other features could be taken into account, such as length of comments, Named Entity Recognition results, or, even more importantly, additional user or product-related information. From a birds eye-view, an immensely alluring possibility would be to dabble into the Decision Support Systems realm and enrich the software with abilities that make it the perfect weapon to assist companies on its journey to improve customer satisfaction. From prediction capabilities to pattern identification, true value can be provided by advanced listening solutions, which sit comfortably within a company's larger social strategy. Integration of supervised event detection tools can help fulfil these opportunities. Product development can also largely benefit from potential future iterations of this tool, as its functionalities could help identify the next big innovation or, at least, understand what is lacking from the company's product line.

As many colloquially say, the world is your oyster when it comes to Sentiment Analysis solutions. With so many advantages and forward-looking abilities, one cannot just risk falling behind.

Bibliography

7 Reasons Why Social Listening Is Important (no date). Available at: https://www.theodysseyonline.com/why-prom-is-important.

Aggarwal, C. C. (2014) *Data Classification Algorithms and Applications*. Chapman & Hall/CRC.

Avinash, M. and Sivasankar, E. (2019) 'A study of feature extraction techniques for sentiment analysis', *Advances in Intelligent Systems and Computing*, 814. doi: 10.1007/978-981-13-1501-5_41.

Boyd-Graber, J., Hu, Y. and Mimno, D. (2017) 'Applications of Topic Models', *Foundations and Trends® in Information Retrieval*, 11(2–3), pp. 143–296. doi: 10.1561/1500000030.

Carbonell, J. G., Michalski, R. S. and Mitchell, T. M. (1983) 'An Overview of Machine Learning BT - Machine Learning: An Artificial Intelligence Approach', in Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (eds). Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–23. doi: 10.1007/978-3-662-12405-5_1.

Cochrane Courtney (no date) *Time Series Nested Cross-Validation - Towards Data Science*. Available at: https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9 (Accessed: 30 June 2020).

Cohen's Kappa - Towards Data Science (no date). Available at: https://towardsdatascience.com/cohens-kappa-9786ceceab58 (Accessed: 30 June 2020).

Cohen's kappa - Wikipedia (no date). Available at: https://en.wikipedia.org/wiki/Cohen%27s_kappa (Accessed: 30 June 2020).

Data Visualization: What It Is, Why It's Important & How to Use It for SEO (no date). Available at: https://www.searchenginejournal.com/what-is-data-visualization-why-important-seo/288127/#close (Accessed: 30 June 2020).

Dave, K., Lawrence, S. and Pennock, D. M. (2003) 'Mining the peanut gallery: Opinion extraction and semantic classification of product reviews', *Proceedings of the 12th International Conference on World Wide Web, WWW 2003*, (October 2003), pp. 519–528. doi: 10.1145/775152.775226.

Džeroski, S. and Ženko, B. (2004) 'Is combining classifiers with stacking better than selecting the best one?', *Machine Learning*, 54(3), pp. 255–273. doi: 10.1023/B:MACH.0000015881.36452.6e.

Fayyad, U. and Stolorz, P. (1997) 'Data mining and KDD: Promise and challenges', *Future Generation Computer Systems*, 13(2), pp. 99–115. doi: https://doi.org/10.1016/S0167-739X(97)00015-0.

Fortuna, P. (2017) 'Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes', p. 109. Available at: https://repositorio-aberto.up.pt/handle/10216/106028.

Gheware, S., Kejkar, A. S. and Tondare, S. M. (2014) 'Data Mining Task Tools Techniques and Applications', *Ijarcce*, (November 2015), pp. 8095–8098. doi: 10.17148/ijarcce.2014.31003.

Godayal Divya (no date) *An introduction to part-of-speech tagging and the Hidden Markov Model*. Available at: https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/ (Accessed: 30 June 2020).

Griffiths, T. and Steyvers, M. (2004) 'Finding Scientific Topics', *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl, pp. 5228–5235. doi:

10.1073/pnas.0307752101.

Gurusamy, V. and Kannan, S. (2014) 'Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining', 5(October 2014).

Horrigan, J. B. (2008) 'Online Shopping. Internet users like the convenience but worry about the security of their financial information', *Pew internet & american life project*, p. 32. doi: citeulike-article-id:4207811.

Hossin, M. and Sulaiman, M. N. (2015) 'A Review on Evaluation Metrics for Data Classification Evaluations', *International Journal of Data Mining & Knowledge Management Process*, 5(2), pp. 01–11. doi: 10.5121/ijdkp.2015.5201.

Hu, M. and Liu, B. (2004) 'Mining and Summarizing Customer Reviews', *Engineering Applications of Artificial Intelligence*, 65, pp. 361–374. doi: 10.1016/j.engappai.2017.08.006.

Huang, G.-B. (2003) 'Huang, G.: Learning Capability and Storage Capacity of Two-Hidden-Layer Feedforward Networks. IEEE Trans. on Neural Networks 14(2), 274-281', *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 14, pp. 274–281. doi: 10.1109/TNN.2003.809401.

Kakarla Swaathi (no date) *Natural Language Processing: NLTK vs spaCy | ActiveState*. Available at: https://www.activestate.com/blog/natural-language-processing-nltk-vs-spacy/ (Accessed: 30 June 2020).

Kazmaier, J. and van Vuuren, J. H. (2020) 'A generic framework for sentiment analysis: Leveraging opinion-bearing data to inform decision making', *Decision Support Systems*. Elsevier, (April), p. 113304. doi: 10.1016/j.dss.2020.113304.

Khan, F., Bashir, S. and Qamar, U. (2014) 'TOM: Twitter opinion mining framework using hybrid classification scheme', *Decision Support Systems*. doi: 10.1016/j.dss.2013.09.004.

Kim, P. (2006) 'The {Forrester Wave: Brand} monitoring, {Q3} 2006'.

Latent Dirichlet allocation - Wikipedia (no date). Available at: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation (Accessed: 30 June 2020).

Lee, L. (2003) '"I'm sorry Dave, I'm afraid I can't do that": Linguistics, Statistics, and Natural Language Processing circa 2001', pp. 1–6. Available at: http://arxiv.org/abs/cs/0304027.

Liu, B. (2010) 'Sentiment Analysis and Subjectivity', *ISCAIE 2019 - 2019 IEEE Symposium on Computer Applications and Industrial Electronics*, pp. 272–277. doi: 10.1109/ISCAIE.2019.8743799.

Liu, B. (2012) 'Sentiment analysis: Mining opinions, sentiments, and emotions', *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, (May), pp. 1–367. doi: 10.1017/CBO9781139084789.

Liu, B., Hu, M. and Cheng, J. (2005) 'Opinion Observer: Analyzing and Comparing Opinions on the Web', *Proceedings of the 14th International Conference on World Wide Web*, pp. 342–351. doi: 10.1145/1060745.1060797.

Lucena, C. I. M. de (2016) *Framework for collaborative knowledge management in organizations*, *PQDT - Global*. Available at: https://search.proquest.com/docview/1985979339?accountid=41307.

Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning (no date). Available at: https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1 (Accessed: 30 June 2020).

Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Nasukawa, T. and Yi, J. (2003) 'Sentiment analysis: Capturing favorability using natural language processing', *Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP 2003*, (March), pp. 70–77. doi: 10.1145/945645.945658.

Oliveira, P. B. De (2019) 'Improving Customer Experience - Predictive Model of Billing Service Requests'.

Pang, B. and Lee, L. (2008) 'Open-domain question-answering', *Foundations and Trends in Information Retrieval*, 1(2), pp. 91–233. doi: 10.1561/1500000001.

Pang, B., Lee, L. and Vaithyanathan, S. (2002) 'Thumbs up? Sentiment Classification using Machine Learning Techniques', *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*, 2018-Janua, pp. 1–5. doi: 10.1109/ICIIECS.2017.8276047.

Patterson, K., Nestor, P. J. and Rogers, T. T. (2007) 'Where do you know what you know? The representation of semantic knowledge in the human brain', *Nature Reviews Neuroscience*, 8(12), pp. 976–987. doi: 10.1038/nrn2277.

Röder, M., Both, A. and Hinneburg, A. (2015) 'Exploring the space of topic coherence measures', *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pp. 399–408. doi: 10.1145/2684822.2685324.

Rolls, E. T. (2000) 'Memory Systems in the Brain', *Annual Review of Psychology*. Annual Reviews, 51(1), pp. 599–630. doi: 10.1146/annurev.psych.51.1.599.

Russell, S. and Norvig, P. (2003) *Artificial Intelligence A Modern Approach Third Edition*, *Pearson*. doi: 10.1017/S0269888900007724.

Rwalk (no date) *GitHub - rwalk/gsdmm: GSDMM: Short text clustering*. Available at: https://github.com/rwalk/gsdmm (Accessed: 30 June 2020).

Samuel, P. and Coelho, A. (2013) 'Multi-Topic Sentiment Analysis'.

Social listening: what it is, why it matters, and how to do it (no date). Available at: https://marketingland.com/social-listening-267175 (Accessed: 30 June 2020).

Social Media Listening: What You Need to Know to Get Started (no date). Available at: https://sproutsocial.com/social-listening/ (Accessed: 30 June 2020).

Social Media Monitoring vs. Social Media Listening (no date). Available at: https://sproutsocial.com/insights/listening-vs-monitoring/ (Accessed: 30 June 2020).

Sousa, J. G. R. de (2019) 'Feature extraction and selection for automatic hate speech detection on Twitter', pp. 1–77. Available at: https://repositorio-aberto.up.pt/bitstream/10216/119511/2/326963.pdf.

Stevens, K. *et al.* (2012) 'Exploring topic coherence over many models and many topics', *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*, (December), pp. 952–961.

Tashman, L. (2000) 'Out-of-sample tests of forecasting accuracy: An analysis and review', *International Journal of Forecasting*, 16, pp. 437–450. doi: 10.1016/S0169-2070(00)00065-0.

Turney, P. D. (2001) 'Thumbs up or thumbs down?', (December 2002), p. 417. doi: 10.3115/1073083.1073153.

Wallach, H. M., Murray, I. and Mimno, D. (2009) 'Evaluation Methods for Topic Models', *Proceedings of the 26th International Confer- ence on Machine Learning*. doi: 10.1007/BF00457859.

Why do stacked ensemble models win data science competitions? - The SAS Data Science Blog (no date). Available at: https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/ (Accessed: 30 June 2020).

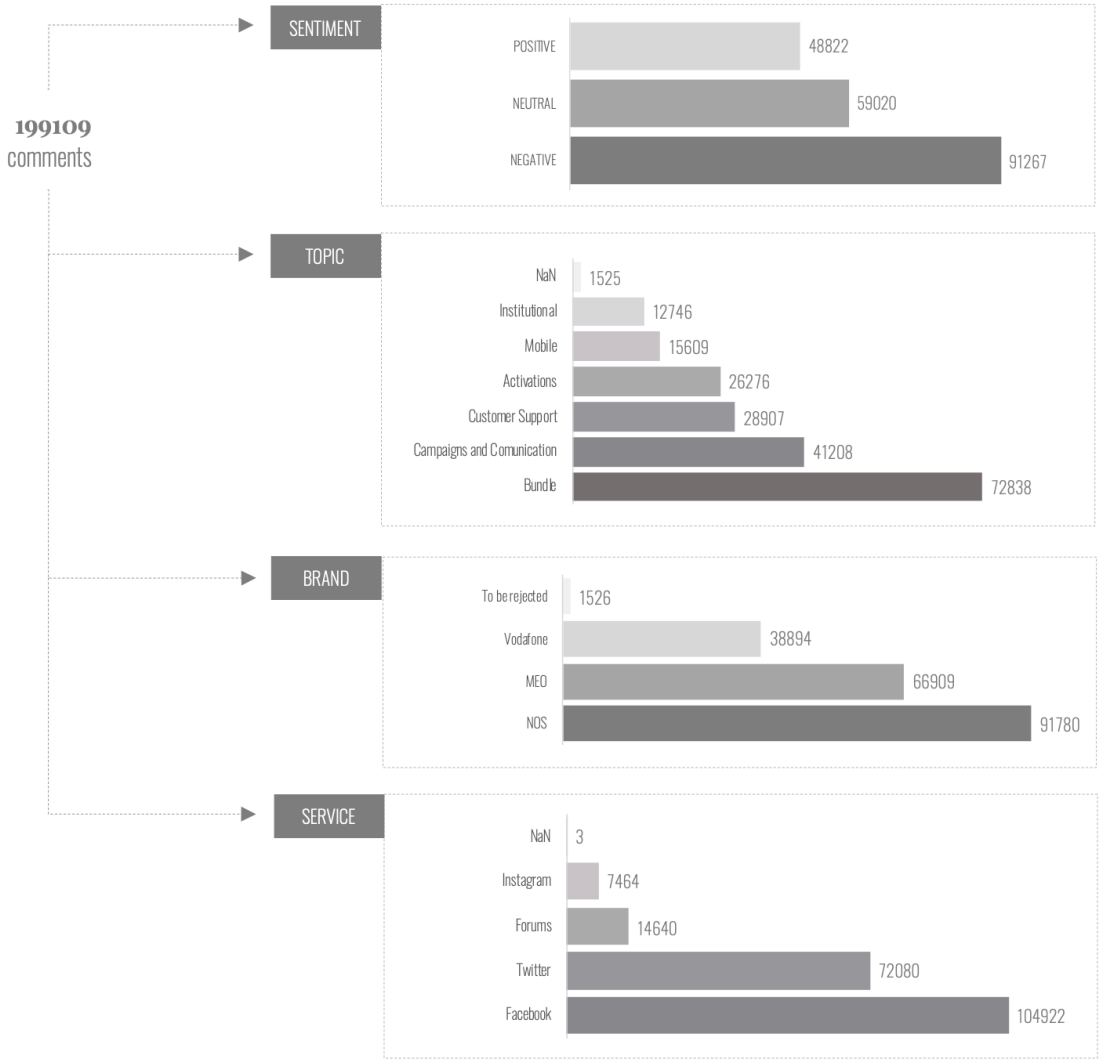
Wirth, R. and Hipp, J. (2000) 'CRISP-DM : Towards a Standard Process Model for Data Mining', *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, (24959), pp. 29–39. doi: 10.1.1.198.5133.

Yin, J. and Wang, J. (2014) 'A Dirichlet multinomial mixture model-based approach for short text clustering', *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 233–242. doi: 10.1145/2623330.2623715.

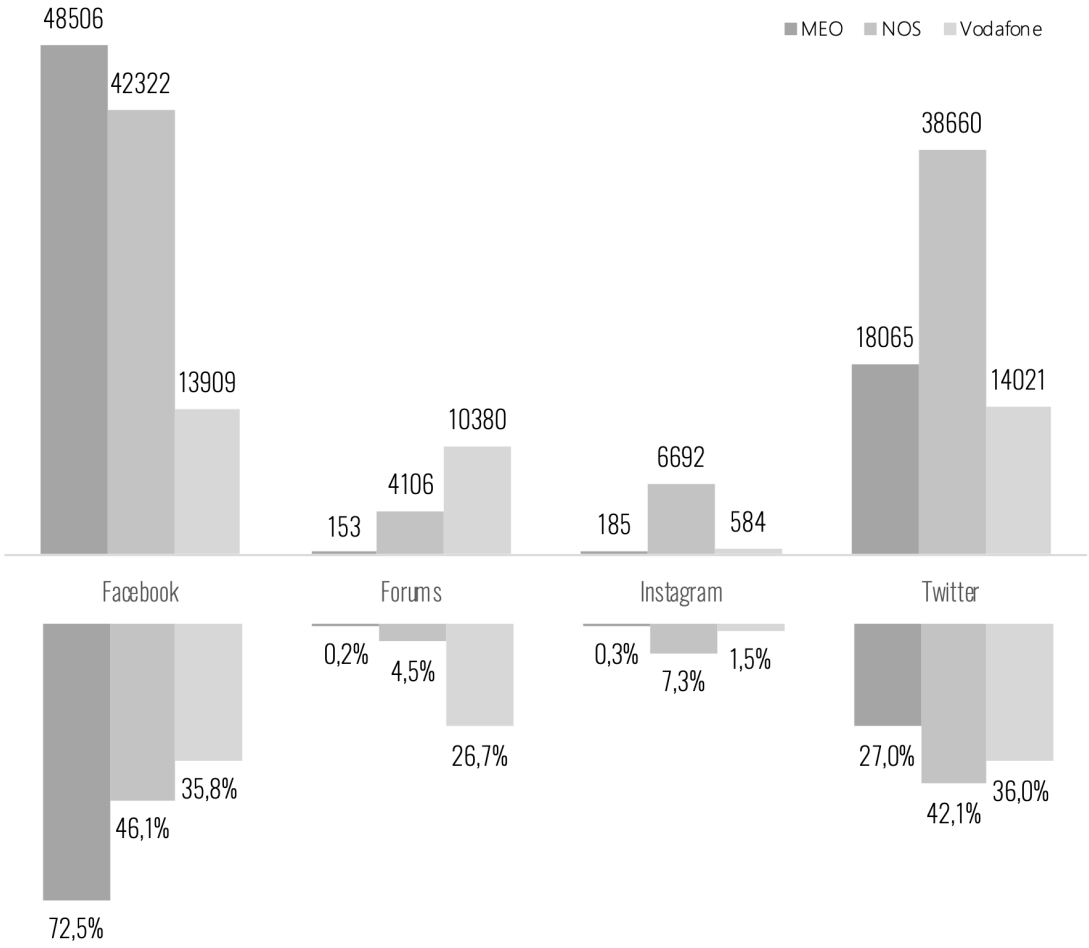
Zabin, J. and Jefferies, A. (2008) 'Social Media Monitoring and Analysis: Generating Consumer Insights from Online Conversation'.

Zhang, L., Wang, S. and Liu, B. (2018) 'Deep learning for sentiment analysis: A survey', *WIREs Data Mining and Knowledge Discovery*, 8(4), p. e1253. doi: 10.1002/widm.1253.

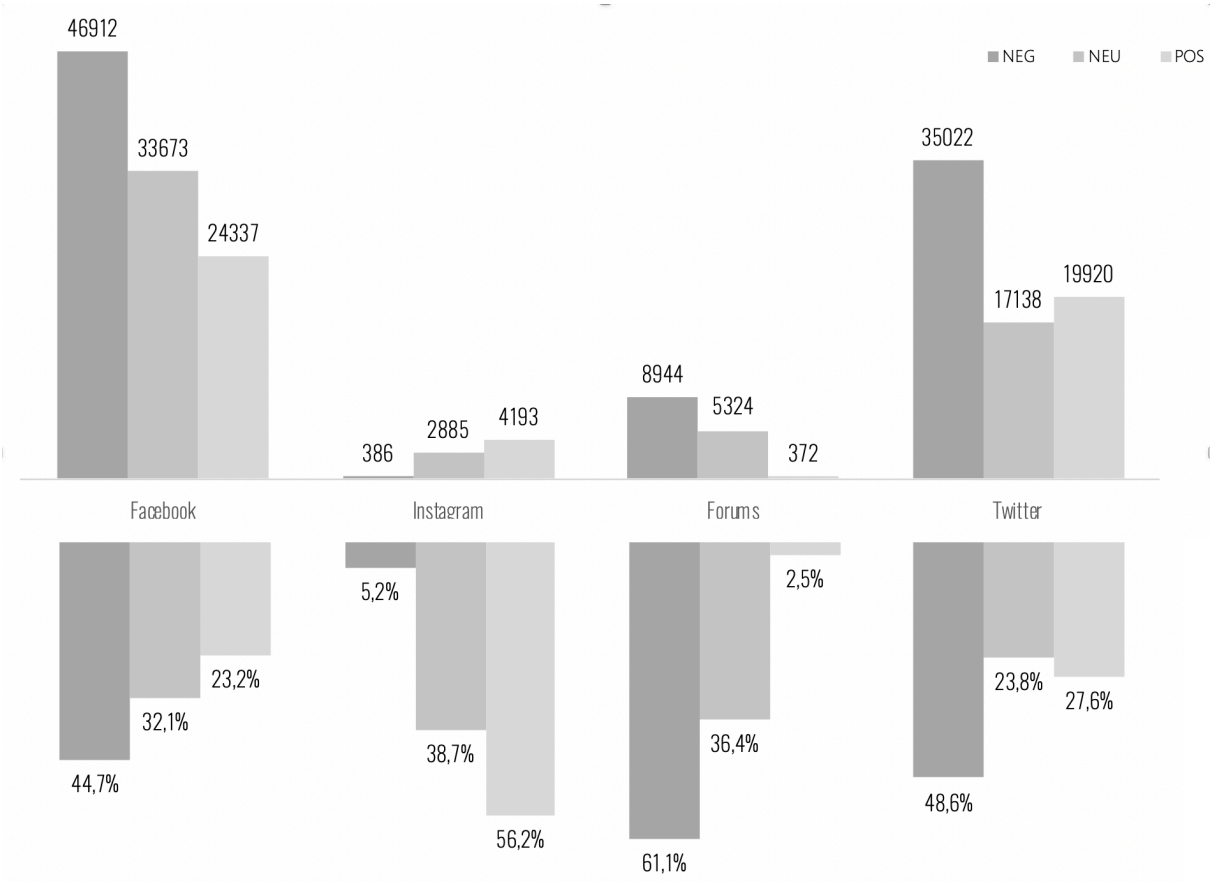
APPENDIX A: Distribution of comments according to Sentiment, Topic, Brand and Service



APPENDIX B: Number of comments of each Brand per Service



APPENDIX C: Number of comments of each Sentiment per Service



APPENDIX D: Sentiment distribution of comments for each Topic and Brand

BRAND > SERVICE > SENTIMENT					
Brand	Topics	TOT (%)	NEG (%)	NEU (%)	POS (%)
MEO	Facebook	72%	49%	34%	17%
	Forums	0%	22%	72%	7%
	Instagram	0%	0%	5%	95%
	Twitter	27%	32%	22%	46%
NOS	Facebook	46%	50%	27%	23%
	Forums	4%	47%	52%	1%
	Instagram	7%	6%	43%	52%
	Twitter	42%	63%	21%	16%
Vodafone	Facebook	36%	13%	41%	46%
	Forums	27%	67%	30%	3%
	Instagram	2%	0%	3%	97%
	Twitter	36%	30%	34%	36%

APPENDIX E: Time Series Split Cross Fold Validation