UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ANALYSIS AND DESIGN OF A TOPIC CLUSTERING SYSTEM BASED ON GENETIC ALGORITHMS FOR ANALYZING CUSTOMER VOICE IN TRANSPORT SERVICES

ALEJANDRO MORENO GARCÍA
JUNIO 2021

TRABAJO DE FIN DE GRADO

Título:	Análisis y Diseño de un Sistema de Agrupamiento de Temas basado en Algorítmos Genéticos para el estudio de la Atención al Cliente en los Servicios de Transporte
Título (inglés):	Analysis and Design of a Topic clustering system based on Genetic Algorithms for analyzing Customer Voice in Transport Services
Autor:	Alejandro Moreno García
Tutor:	Carlos Ángel Iglesias Fernández
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	—
Vocal:	—
Secretario:	—
Suplente:	—

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

ANALYSIS AND DESIGN OF A TOPIC CLUSTERING SYSTEM BASED ON GENETIC ALGORITHMS FOR ANALYZING CUSTOMER VOICE IN TRANSPORT SERVICES

Alejandro Moreno García

Junio 2021

Resumen

Durante la última década, la cantidad de información pública que existe en las redes sociales está en constante crecimiento. Esto se debe mayoritariamente a que el número de usuarios que utilizan estas plataformas está aumentado, y con ello la cantidad de datos e información relevante sobre los mismos. Además, el uso de las redes sociales ha revolucionado en todo el mundo la manera que tenemos los seres humanos de comunicarnos. Debido a este auge, muchas empresas de movilidad han decido modificar su modelo de atención al cliente, el cual se basaba en formularios a papel y llamadas telefónicas, para adaptarse a esta nueva forma de comunicación pública entre usuarios y empresas.

Este proyecto tiene el propósito de comprobar si es posible la disección de un servicio de atención al cliente en la red social de Twitter. En concreto, se centrará en análisis de las preguntas y preocupaciones que tienen los usuarios del servicio de transporte Uber. Para ello, se ha creado un conjunto de datos con el objetivo de evaluar los tweets correspondientes al idioma inglés durante el año 2020 en la plataforma @Uber_Support.

En primer lugar, se ha realizado un análisis global de la información recogida para comprender el contexto de los servicios de atención al cliente relacionados con el transporte. Posteriormente, y con el objetivo de obtener la estructura subyacente que alberga este conjunto de datos, se ha realizado un Sistema de Modelado de Temas combinado con un Sistema de Agrupamiento basado en un modelo híbrido de Algoritmos Genéticos. El resultado de ambos sistemas nos ha proporcionado la agrupación de los tweets en siete grupos que se refieren a los temas más discutidos en el conjunto de datos. Por último, el uso combinado de estos sistemas nos ha permitido además caracterizar la polaridad asociada a tema, y con ello conocer en detalle la postura de los clientes en cada tema descrito.

En definitiva, consideramos que el uso de los sistemas descritos en el proyecto para análisis periódico de la información pueden ser de gran utilidad para las empresas que necesiten conocer las opiniones del público para establecer las estrategias de marketing.

Palabras clave: Uber, Servicio al Cliente, Big Data, Twitter, Python, PLN, Modelado de Temas, Aprendizaje Automático, Algorítmos Genéticos, Análisis de Sentimientos

Abstract

During the last decade, social media information has experienced constant growth. This fact is largely due to the increment of users using these platforms and the amount of data and relevant information about them. Moreover, social media has completely changed the way people communicate and interact with the rest of the world. This impact has directly affected customer engagement. Some of the most important companies have decided to modify their customer service model based on phone calls and paper forms to be adapted to this new way of communicating between users and companies.

Keeping this in mind, this project aims to determine if it is possible to dissect a customer service on the Twitter social network. Specifically, we will focus on analysing the questions and concerns that users from the Uber transport service have. To do this, we have created a dataset containing tweets from the English speakers addressing the @Uber_Support platform during the year 2020.

Firstly, we performed a global analysis of the collected data. This first approach helped us to understand how users publish information in the context of both transport and customer services. Secondly, to characterize the dataset's underlying structure, we implemented a Topic Modelling System combined with a Topic Clustering one based on a hybridized Genetic Algorithm. The performance of both systems described a result of forming seven groups of topics in order to cluster the different tweets from the dataset efficiently. Finally, the combined implementation of both systems also allowed us to characterize the polarity associated with each topic, giving us the complete stance of users towards the specific issues described in the platform.

To conclude, we believe that the systems created in the project can be efficiently used for the periodic analysis of the information on social media. Thus, companies who require public opinions to establish marketing approaches can easily handle this high user-generated content by using the systems described.

Keywords: Uber, Customer service, Big Data, Twitter, Python, NLP, Topic Modelling, Machine Learning, Genetic Algorithms, Opinion mining

Agradecimientos

Con la entrega de esta memoria termina uno de los proyectos que sin lugar a dudas ha sido de los que más he aprendido en toda la carrera. No sin antes agradecer a todas las personas que sin su apoyo no habría sido posible llegar hasta aquí.

En primer lugar, quiero agradecer a mi hermano Enrique todas las cosas que he aprendido gracias a él y la paciencia que tiene conmigo, especialmente durante este último año. Además, me gustaría agradecer a mi madre Nieves y a mi padre Enrique todo el cariño y apoyo que nos dan tanto a mi hermano como a mí desde siempre, lo cual ha hecho posible que nos enfrentemos a todo lo que nos proponemos. Gracias a los tres por ser parte de mi familia y un apoyo fundamental en mi vida.

En segundo lugar, me gustaría expresar mi gratitud a los compañeros que la ETSIT me ha permitido conocer durante estos cuatro años de carrera, los cuales muchos se han convertido en amigos muy importantes para mi. Muchas gracias por hacer que la ingeniería sea mas amena, por todas las risas que hemos tenido, y también por las que estoy seguro que quedan.

Por último, me gustaría dar las gracias a mi tutor Carlos Ángel, cuyo apoyo durante este proyecto me ha permitido mejorar tanto profesionalmente como personalmente en muchísimos aspectos de la vida.

Muchas gracias a todos.

Contents

Resumen **I**

Abstract **III**

Agradecimientos **V**

Contents **VII**

List of Figures **XI**

List of Tables **XIII**

Acronyms **XV**

1 Introduction **1**

1.1 Context 1

1.2 Project goals 2

1.3 Structure of this document 4

2 Enabling Technologies **5**

2.1 Introduction 5

2.2 Natural Language Processing technologies 6

2.2.1 Natural Language Toolkit 6

2.2.2 SpaCy 6

2.2.3 Gensim 7

2.2.4 Senpy 7

2.3 Machine learning technologies 8

2.3.1 Scikit-learn 8

2.3.2 DEAP 9

2.3.3 Scipy 9

2.4 Data modelling technologies 9

2.4.1 Twitter Intelligence Tool 9

2.4.2 Numpy 10

2.4.3 Pandas 10

2.5 Data visualization technologies 11

2.5.1 Matplotlib 11

2.5.2 PyLDAvis 11

2.5.3 Yellowbrick 11

2.5.4 Scattertext 12

3 Dataset Description 13

3.1 Introduction 13

3.2 Dataset collection and analysis 14

3.2.1 Dataset collection 14

3.2.2 Dataset analysis 15

3.2.2.1 Monthly volume of tweets 15

3.2.2.2 Most frequent words 16

3.2.2.3 Hashtags analysis 16

3.2.2.4 Sentiment analysis 17

4 Machine Learning Model and Evaluation 19

4.1 Introduction 19

4.2 Data pre-processing 20
4.3 Topic Modelling System 22
4.3.1 Latent Dirichlet Allocation 22
4.3.2 Determining T optimal number of topics 23
4.3.3 Topic Modelling Results 26
4.4 Topic Clustering System 27
4.4.1 Introduction 27
4.4.2 Evaluation metrics 27
4.4.3 K-Means clustering algorithm 28
4.4.3.1 Introduction to K-means 29
4.4.3.2 Evaluation of K-Means 29
4.4.3.3 K-Means results 31
4.5 Genetic Algorithms 33
4.5.1 Introduction to Genetic Algorithms 33
4.5.1.1 Selection operator 34
4.5.1.2 Crossover operator 35
4.5.1.3 Mutation operator 35
4.5.2 Genetic Algorithm case of study 36
4.5.2.1 Initial Population Description 36
4.5.2.2 Fitness Function 37
4.5.3 First Case of Study 38
4.5.4 Second Case of Study 39
4.5.4.1 Description of operators 39
4.5.4.2 Results 40
4.5.4.3 Conclusion 42
4.6 Sentiment analysis 44

4.6.1 Sentiment of words and phrases . . . 44

4.6.2 Sentiment of topics . . . 45

5 Conclusions and future work 47

5.1 Conclusions . . . 47

5.2 Future work . . . 49

Appendix A Impact of this project i

A.1 Social Impact . . . i

A.2 Economic Impact . . . ii

A.3 Environmental Impact . . . ii

A.4 Ethical Implications . . . iii

Appendix B Economic budget v

B.1 Introduction . . . v

B.2 Physical resources . . . v

B.3 Human resources . . . vi

B.4 Software and licences . . . vi

B.5 Taxes . . . vii

Appendix C Sample of tweets distribution ix

Appendix D Detailed explanation of the first Genetic Algorithm study case xi

D.1 Description of operators . . . xii

D.2 Results . . . xiii

D.3 Conclusions . . . xv

Appendix E Genetic Algorithm Iterations xvii

Bibliography xix

List of Figures

2.1 Senpy architecture 7

3.1 Monthly volume of tweets. 15

3.2 Most repeated words tweeted by customers. 16

3.3 Most frequent hashtags in the dataset. 17

3.4 Global sentiment of tweets 18

4.1 Pre-processing process. 21

4.2 Topic Modelling Process. 22

4.3 Coherence score from LDA models with respect the number of topics. . . . 23

4.4 PyLDAvis model representations 25

4.5 Evaluation of the optimal number of clusters with the Elbow Method. . . . 30

4.6 Silhouette score. 31

4.7 Silhouette diagram (K=7). 31

4.8 K-Means cluster-centroids. 31

4.9 K-Means distribution of documents. 32

4.10 Genetic Algorithm process. 34

4.11 Crossover methodologies. 35

4.12 Individual representing a solution based on 7 cluster-centroids. 36

4.13 Genetic Algorithm convergence. 41

4.14 SLSQP algorithm performance. 42

4.15 Genetic Algorithm second study case results 43

4.16 Frequency of words for both Positive and Negative categories 44

4.17 Sentiment associated to topics using the Genetic Algorithm for clustering. . 45

D.1 Mutation pseudocode first GA xiii

D.2 Genetic Algorithm convergence. xiii

D.3 Cluster-centroids solution in the first study case xiv

List of Tables

4.1 Description of topics and most frequent words. 26

4.2 Genetic operators in the second case of study. 40

4.3 Evaluation of K-means and Genetic Algorithm of the second case of study . 41

4.4 Evaluation of K-means, Genetic Algorithm and Hybridized GA 42

C.1 Sample tweets in each topic. x

D.1 Genetic operators in the first case of study. xii

D.2 Evaluation of K-means and Genetic Algorithm from the first study case . . xiv

D.3 Tweets in each cluster xiv

E.1 Iterations of the genetic algorithm from the first study case. xviii

E.2 Iterations of the genetic algorithm from the second study case. xviii

Acronyms and Abbreviations

COVID-19: Coronavirus disease 2019

GA: Genetic Algorithm

GSI: Intelligent Systems Group

LDA: Latent Dirichlet Allocation

ML: Machine Learning

MSE: Mean Squared Error

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

POS: Part of Speech

SLSQP: Sequential Least Squares Programming

SSE: Sum of the Squared Error

TWINT: Twitter Intelligence Tool

CHAPTER 1

Introduction

1.1 Context

During the last decade, social networks have become one of the main communication tools in society where users can share their daily activities and give their opinions about any theme. These platforms have increased their number of users in recent years and it is estimated that at the end of 2021 there will be 3.78 billion of active users[1]. Furthermore, social media has completely changed the way people communicate and interact with the rest of the world [1, 2].

This impact has directly affected to customer engagement. Some of the most important companies have decided to modify their customer service model based on phone calls and paper forms to be adapted to this new way of communicating. This led companies to create official accounts on the most popular social media networks in order to help customers with their concerns, questions, and opinions.

Customers can freely express their satisfaction with a brand and this could directly affect the brand popularity, since millions of users can read this public information. Moreover, companies need to analyze their competitors to obtain a competitive advantage [3]. For

[1] www.statista.com

these reasons, the periodic capture and analysis of this large amount of user-generated content will be critical for understanding the public opinion about brands and the provision of higher quality responses that satisfy their needs.

Keeping this in mind, the aim of this project is to analyse a customer service from Twitter. This social network is one of the most popular platforms referring to customer services. Its simplicity of publishing short messages by using smart devices has tailored perfectly with customer needs of communication with brands.

Our first intention is to determine if it is possible to analyse the most discussed topics by users in a customer service platform. This first approach will give us a complete vision of the common questions and concerns related to the transport service.

Moreover, we are interested in determining the sentiments of users they tweet to these services. The combination of these two analysis will give us the complete stance of a user towards an specific topic, which can be deterministic to evaluate the expectations and satisfaction of the users with a brand.

We decided to analyze Uber Customer Service (@Uber_Support[2]) due to its fast growth in the ride-sharing sector [4] and its robust customer service. Furthermore, we are particularly interested in customer replies to the platform. We consider these tweets as the most valuable ones for companies to understand the real necessities of customers when communicating with brands.

However, the individual analysis of these tweets may not represent the opinions and concerns of the majority of customers. For this reason, we will need to apply techniques that provide an efficient analysis on these large amount of tweets.

1.2 Project goals

The main objective of this project is to create a computational data mining model capable of ensuring an efficient analysis of a large amount of data-generated content from a Customer Service platform from the Twitter domain.

Specifically, tweets will be collected from the @Uber_Support platform since this account addresses the Official Customer Service of Uber on Twitter.

We will focus the study on the replies of users towards this service. Furthermore, the replies captured will focus on the English speaker users.

[2] https://twitter.com/Uber_Support

In order to achieve these objectives, the following tasks have been carried out during the project:

- Study the state of art about Natural Language Processing, Machine Learning Algorithms and Genetic Algorithms.
- Perform the collection of tweets from the user @Uber_Support customer service from the Twitter domain in order to create a dataset to analyze.
- Perform a global analysis of the most characteristic features from the dataset (this includes the monthly volume of tweets, most frequent words, hashtags analysis and sentiment analysis) to understand the context of customer and transport services.
- Design and analyze a Topic Modelling System based on machine learning to obtain the *latent* topics from the dataset following the next steps:
  - Perform an individual and detailed pre-processing of the raw tweets collected. This process will allow the model to retain the most relevant characteristics and reduce the dimensionality of corpora to feed the machine learning algorithm efficiently.
  - Interpretate the results provided by the machine learning algorithm.
- Design and analyze a Topic Clustering System to group the collected tweets into the predefined topics described efficiently. This process includes:
  - Develop a machine learning algorithm for clustering. This algorithm will help us establish a basis and check the model's viability before performing the genetic algorithm.
  - Develop a Genetic Algorithm for clustering aiming to optimize the current clustering technologies.
- Analyze the results provided by both Topic Modelling and Clustering Systems to obtain each topic's polarity.

1.3 Structure of this document

To achieve the different objectives described in the previous section, this project will be divided into the following chapters and appendices:

Introduction. Chapter 1: This chapter introduces the context of the project by analyzing the impact of Social Media in society and its effect on customer engagement. Moreover, it describes the project goals to achieve and the structure of the complete document.

Enabling Technologies. Chapter 2: This chapter provides the reader with the principal technologies used in the project.

Dataset Description. Chapter 3: This chapter describes the process of the dataset formation. Moreover, it provides several general analysis from the data collected, which will be useful in future chapters.

Machine Learning Model and Evaluation. Chapter 4: This chapter describes the processes implemented for creating the Topic Modelling System and the Topic Clustering System, as well as the results obtained for each model. Finally, it introduces the reader to the different sentiments towards each topic.

Conclusion and future work. Chapter 5: This chapter describes the conclusions and the future outcomes that can be performed in the project.

Impact of this project. Appendix A: This appendix describes the social, economic, environmental and ethical implications of the project.

Economic budget. Appendix B: This appendix describes in detail the economic budget needed to create the project.

Sample of tweets distribution. Appendix C: This appendix gives the reader several samples of tweets and their distribution through the topics as well as their polarity once finished the project.

Detailed explanation of the First Genetic Algorithm study case. Appendix D: This appendix describes in detail the first study case of genetic algorithms implemented in the project.

Genetic Algorithm Iterations. Appendix E: This appendix describes the different iterations of the genetic algorithms implemented in the project.

CHAPTER 2

Enabling Technologies

2.1 Introduction

This chapter provides an overview of all the libraries, frameworks, and visualization technologies implemented throughout the project. Firstly we will introduce the scientific fields this project is based on. This includes the concepts of Natural Language Processing (NLP), Machine Learning, and Genetic Algorithms.

At first instance, we intend to analyze a customer service based on messages from the Twitter domain. For this reason, we will need several to use tools that make possible the transformation of plain texts (here tweets) into numerical expressions that computers can process. For this reason, the use of Natural Language Processing is required. NLP is a subfield of computer science that makes possible the interaction between computers and human language to analyze and interpret large amounts of texts through several syntactical and semantical analyses.

Once these large amounts of natural language data are transformed into numerical values, there will be two major steps this project will deal with. These steps include machine learning and optimization, which will be done using Genetic Algorithms.

Machine learning is composed of a set of algorithms that studies data input for making predictions that will improve over time. Machine learning can be divided into three main fields: Unsupervised, Supervised, and Semi-Supervised Learning.

As we mentioned in the introduction (Section 1.2), our goal is to create a Topic Modelling and Clustering System based on messages from Twitter. Hence, both systems will be carried out through unsupervised machine learning algorithms since the collected tweets do not present any previous reference nor label to classify them. For this reason, the machine learning techniques implemented will help us to understand the underlying structure of our dataset.

As the last step, the genetic algorithms [5] will be performed in this project aiming to optimize the clustering solution provided in the machine learning algorithms. These algorithms are adaptative heuristic search and optimization techniques that provide high-quality solutions based on the natural selection process and genetics.

The following sections of this chapter will introduce the aforementioned technologies implemented in the project.

2.2 Natural Language Processing technologies

2.2.1 Natural Language Toolkit

Natural Language Toolkit (NLTK) [6] is an extended set of Python libraries that works with human language data. These libraries were written by Steven Bird, Edward Loper, and Ewan Klein in 2001 for development and educational purposes.

NLTK provides more than 50 corpora and lexical resources that allow researchers the capacity for classifying, tokenizing, stemming, tagging, and parsing their documents.

This library has been implemented in our project for tokenizing, removing stopwords, and stemming to train our machine learning algorithms efficiently.

2.2.2 SpaCy

SpaCy [7] is an advanced open-source Python library that provides high-performance tools in the NLP field. It was initially developed by Matt Honnibal in 2015 and nowadays it supports more than 60 languages.

We have used this library in our project for implementing the lemmatization and POS-Tagging on our corpus due to its good results in the English domain.

2.2.3 Gensim

Gensim [8] is a Python library written by Radim Řehůřek in 2009 whose main objective is to represent documents as vectors to be processed by computers throughout several algorithms. This library can use several unsupervised machine learning algorithms for performing topic modelling approaches such as Latent Dirichlet Allocation (LDA), Latent Semantic Indexing (LSI), etc. Moreover, the Gensim package provides their users with many different tools for transforming their documents into a dictionary of words and a set of vector representations (doc2bow and bag-of-words) to be analyzed efficiently by computers. Furthermore, the Gensim library allows the execution of the algorithm selected with multi-threads which, is essential for Big Data problems where computational time is deterministic.

We decided to implement the Gensim library in our model to extract the most discussed topics from our dataset using the LDA package.

2.2.4 Senpy

Senpy is a framework for sentiment and emotion analysis services [9] created by the Intelligent Systems Group (GSI) at ETSIT-UPM.

Figure 2.1 shows the most relevant chatacteristics of the Senpy architecture. We can appreciate that this framework is composed of two main elements. The first component is named as the *Core*, and it gives the framework the main functionalities to operate. The second component are the different *plugins* implemented. Each plugin will be composed of different machine learning algorithms for analyzing the data provided.

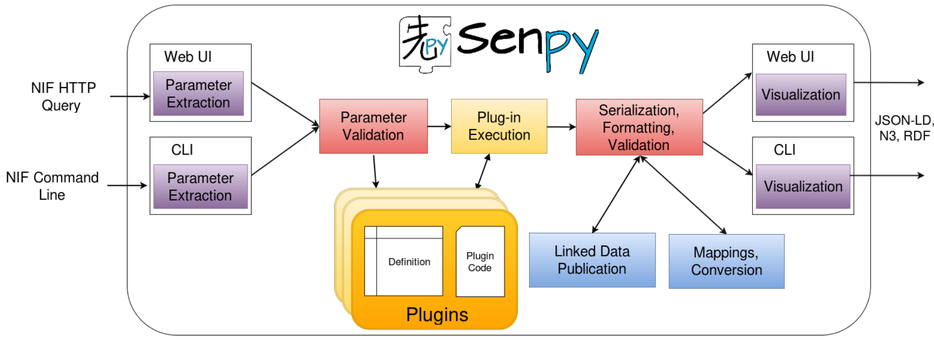


Figure 2.1: Senpy architecture

The main advantage of Senpy among other online services from this field is its simplicity for merging many different sentiment and emotion algorithms in a unique and scalable platform.

The principal algorithm used for the extraction of sentiments in the project was the Sentiment140 commercial service.

2.3 Machine learning technologies

2.3.1 Scikit-learn

Scikit-learn [10] is an open-source Python package for machine learning developed by David Cournapeau in 2007 during a project of the Google Summer of Code (GSoC).

This library is built on the top of Numpy [11], Scipy [12], and matplotlib [13] libraries. Scikit-learn [10] provides a wide range of highly optimized packages for supervised and unsupervised machine learning algorithms such as:

- Clustering: Unsupervised algorithms used for groping objects that presents similarities into different sets.
- Classification: Supervised algorithms used for identifying the category of objects.
- Regression: Supervised algorithms used for predicting a continuous-valued attribute referred to an object
- Dimensionality Reduction: Algorithms used for reducing the number of features (variables) to process.
- Model selection: Algorithms used for comparing, validating and choosing parameters and models.
- Preprocessing: Packages used for feature extraction and normalization

Scikit-learn clustering package was implemented in this project before the development of the genetic algorithm to check the validity of the machine learning algorithms for clustering the tweets.

2.3.2 DEAP

Distributed Evolutionary Algorithms in Python (DEAP) [14] is an open-source computational framework developed at Université Laval in 2009 that provides researchers high performance in prototyping and experimenting with evolutionary algorithms.

This package allows researchers to implement several evolutionary algorithms for efficiently solving NP-Hard computational problems using Genetic Algorithms, Genetic Programming, Multi-Objective Evolutionary Algorithms (such as NSGA-II and SPEA2), and many others. Furthermore, DEAP is compatible with several parallelization schemas such as SCOOP and thread-multiprocessing, which will significantly reduce computational times in evaluating the fitness function.

DEAP was used in our project to create all the necessary operators for performing the Genetic Algorithms.

2.3.3 Scipy

SciPy [12] is a community-driven project implemented as a Python library. It was first developed by Travis Oliphant, Pearu Peterson, Eric Jones in 2001.

This library contains a wide variety of mathematical algorithms that gives researchers the capacity of performing many different engineering tasks such as Linear Algebra Operations, Interpolation, Optimization and fit, Fast Fourier transforms, Signal Processing, Image manipulation, etc.

In this project, we have used Scipy's optimization package to perform a local convergence on the result provided by the genetic algorithm. Particularly, we used the SLSQP module.

2.4 Data modelling technologies

2.4.1 Twitter Intelligence Tool

Twitter Intelligence Tool (TWINT) [1] is an easy-to-use Python library that allows researchers to collect tweets from any profile from Twitter social media.

This project was mainly created by Francesco Poldi and it has become one of the primary data collection sources for many researching projects in recent years. Moreover, this OSINT

[1] https://github.com/twintproject/twint

tool provides several advantages in comparison with Twitter scrapping tools using Twitter API.

- **No scrapping limitations:** Twitter API has a limitation for collecting past tweets. It only allows researchers to collect tweets up to 7 days ago. Moreover, Twitter API only allows us to collect 18000 tweets in intervals of 15 minutes. However, TWINT has no limitations for collecting tweets, and we can easily obtain past tweets for its analysis.

- **Fast results:** TWINT internal optimization packages have made it possible to collect high volumes of tweets in a few period of time.

We decided to use the TWINT scrapping tool to collect all the tweets needed for our project because we decided to collect a complete year of tweets from the Uber Customer Service platform.

2.4.2 Numpy

Numpy is an open-source Python library created by Travis Oliphant in 2006 [11]. It precedes from *Numeric* and it has become the main package used for scientific computing in any Python project.

This library allows us to work with N-dimensional arrays and matrices. The main advantage of Numpy is its wide range of mathematical functions to operate with these arrays and its efficiency in time-computing.

Numpy library has been used in this project for reducing time processing in the mathematical operations used in clustering.

2.4.3 Pandas

Pandas is an open-source library written in Python language and build by Wes Mckinney in 2008 [15].

It has become a flexible and widely used tool that provides high-performance data structures and data analysis. The main advantage of Pandas over traditional methods is its simplicity for cleaning, grouping, merging, and efficiently querying data structures. Moreover, this library provides versatile compatibility for reading and writing in a wide range of file formats such as comma-separated-values (CSVs), JSONs, SQLs, and Microsoft Excel files.

This tool was used during the entire project for cleaning, grouping, reshaping, and analyzing the different features of the tweets from the datasets created.

2.5 Data visualization technologies

2.5.1 Matplotlib

Matplotlib [13] is one of the most popular Python libraries for plotting arrays and lists in 2-D graphical visualization. This comprehensive library was created by John Hunter in 2003, and it is inspired by MATLAB [16] visualization tools. Moreover, it contains many internal resources that allow researchers to plot very different visualization graphs.

In our case, we decided to use this library for plotting bar charts, pie charts, area charts and line plots in this project.

2.5.2 PyLDAvis

PyLDAvis [17] is a Python open-source library developed by Carson Sievert and Kenneth Shirle that allows researchers to visualize and interpret topics created by machine learning models.

This package plots the user's predefined topics in a 2-dimensional representation composed of circles (containing different distributions of words) and a series of histograms representing the comparison of the overall frequency of a term and its specific frequency on the topic selected.

In this project, pyLDAvis was implemented in Section 4.3 for understanding the distribution of words given by LDA through the topics plotted by using an interactive HTML file.

2.5.3 Yellowbrick

Yellowbrick [18] is an open-source Python library created to simplify the visualizations and analysis provided from the scikit-learn [10] machine learning algorithms. This library supports a wide variety of visualizations from many different machine learning models such as: Clustering, Regression, Classification, Feature Extraction, etc.

We used the Yellowbrick library in our project for plotting several clustering visualizations in Section 4.4. Specifically, we focused this library on studying several metrics used in the K-means algorithm to determine the optimal number of clusters in the dataset.

2.5.4 Scattertext

Scattertext is an open-source Python package created by Jason S. Kessler in 2017 [19].

The main purpose of this library is to visualize the frequency of words and phrases for a given categories in a 2-D scatter plot by performing an initial pre-processing using Spacy [7]. The results provided by this package allow us to interpret results in an interactive HTML file.

We decided to use this tool for plotting the frequency of polarity for each word and phrase from our corpus. We divided the categories required in the library into *Positive* and *Negative* sentiment as explained in Section 4.6.1.

CHAPTER 3

Dataset Description

3.1 Introduction

This chapter describes the collection and analysis of the dataset created for the project. This dataset has been obtained from the Twitter domain to analyze the opinions and concerns of users who use ride-hailing services. Specifically, we focused our study on those users and drivers that uses the Uber Transport Service.

Uber Technologies, Inc.[1] is a transport company created by Garrett Camp and Travis Kalanick in 2009 whose vision is to offer their users a technological platform to connect with drivers subscribed to the company. Uber operates in 69 different countries worldwide, where it is estimated that this platform has almost 90 million active users [2] each month. Uber's priority service is the ride-hailing service. However, as this company gained popularity in the sector, they adapted their business model to their customer's needs. For this reason, Uber started to offer in recent years many other services such as food delivery (Uber Eats), freight transportation, package delivery, rental of motorized scooters and bicycles, etc.

[1] https://www.uber.com

[2] https://www.statista.com/statistics/833743/us-users-ride-sharing-services/

This technological platform has many different channels where their customers can solve their questions and concerns about the services provided by the company. One of the most important channel for contacting Uber is located on Twitter with the username @Uber_Support. This Customer Service platform was created in 2014 to help their customers and drivers solve their issues online by easily posting short and public messages. @Uber_Support is used worldwide by their customers since this platform allows them to write in many languages such as English, Spanish, Italian, Arabic, etc., for contacting Uber.

We selected the @Uber_Support Twitter account for making our study since Uber customers usually post their questions and concerns on this official customer service platform.

3.2 Dataset collection and analysis

3.2.1 Dataset collection

Data was collected from tweets that contain the keyword "@Uber_Support" to focus on the Official Uber Customer Service. Additionally, we decided to capture tweets between January 01 2020, and December 31, 2020 to focus on the complete year 2020. This collection of tweets was compiled through the use of TWINT (Section 2.4.1). Then, tweets were filtered through the following criteria:

- They must be written in English since our goal is to address to the English speakers community.
- Tweets posted by the customer service "@Uber_Support" were eliminated since our achievement is to analyse customers' demands and issues with the brand.
- Duplicated tweets were eliminated.
- We have analysed the users who posted the most to detect spam. To do this, we selected the first 20 user accounts that posted the most in the dataset. Then, if one of those users was a spammer, the whole account was eliminated from the dataset. This analysis was performed to reduce the creation of false topics in the Topic Modelling approach (Section 4.3).

Following the steps mentioned above, the final capture resulted in 215387 raw tweets.

3.2.2 Dataset analysis

Before performing the pre-processing of tweets (Section 4.2), we made several analysis on this dataset created to understand some relevant features, which are unique in the dataset. This analysis will focus on the monthly volume of tweets, the most repeated words in the dataset, the most relevant hashtags posted by their users and the global analysis of sentiments from this dataset. This deep analysis will give us a first approach that will be useful to understand public opinion towards Uber Service. Moreover, in future chapters, this initial dataset description will help us to understand the results given by the machine learning models.

3.2.2.1 Monthly volume of tweets

As we mentioned before, we collected a dataset focused on the capture of an entire year (2020) from the Official Customer Service @Uber_Support platform on the Twitter domain. These individual tweets contained the attribute “Date”, which gave us the hour and day this tweet was posted on the platform. Hence, we can have the daily volume of tweets posted by users. We merged this high volume of daily tweets into new groups divided by months.

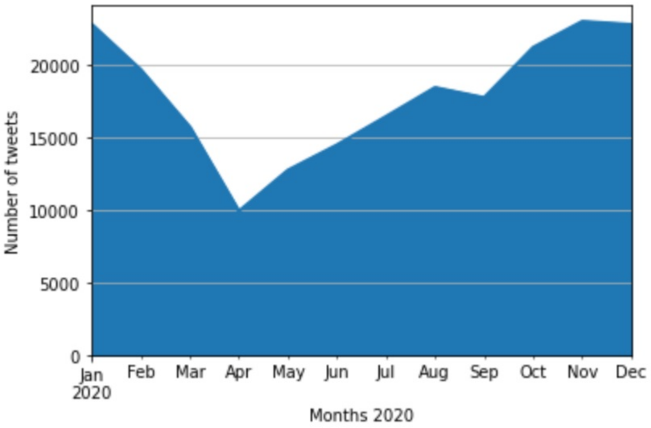


Figure 3.1: Monthly volume of tweets.

As a result, we obtained the monthly volume of tweets posted by customers and drivers to this platform in Figure 3.1. We can observe a significant decrease (43%) in the daily tweet volume during March and April. This decrement occurred with high probability due to the restrictions and confinements suffered during those months throughout the world population due to the pandemic caused by Coronavirus disease (COVID-19). After restrictions and confinements were decreasing, we observe that people started to use again transport platforms to recover their daily activities. This means that people also started to tweet their problems and opinions on the different transport platforms again. For this reason, from June to December, tweets volume started to increase until it reached its previous monthly values resulting in approximately 25000 tweets per month.

3.2.2.2 Most frequent words

We also collected the most frequent words of customers and drivers to check the most common and relevant issues these users post on the platform. This process was held through the FreqDist package from NLTK [6] which provided us a dictionary of the different words from the corpus with its frequency. We sorted these words according to their frequency in the dataset in order to show in Figure 3.2 a word cloud with these most repeated words.

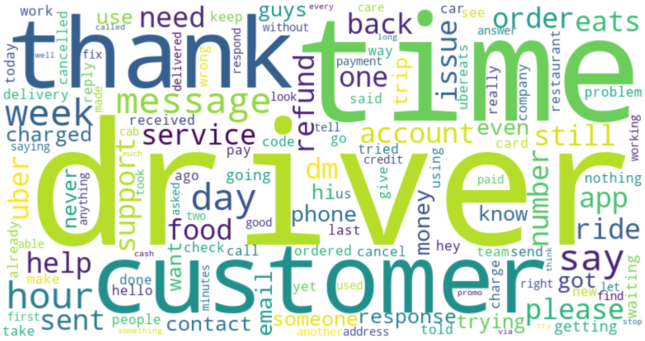


Figure 3.2: Most repeated words tweeted by customers.

This word cloud provides each word with a different size according to the frequency on the corpus. For example, words like “driver”, “customer”, “time”, and “thank” are the most frequent words. These word frequencies show that people commonly tweet to @Uber_Support for discussing problems or recommendations related to drivers and their customer service. In section 4.3.3 we will describe in detail the latent topics obtained, but we can ensure that these words are going to be highly relevant in the Topic Modelling analysis since they are the most repeated in the tweets collected.

3.2.2.3 Hashtags analysis

Hashtags are keywords commonly used on social networks for classifying the messages posted by their users into different categories. The messages containing a hashtag can be easily searched by other users from the platform since they are preceded by the characteristic token “#”. In our project, we created a dictionary of the most 20 frequent hashtags from the dataset in order to analyze the most common categories that Uber users post to summarize their tweets. To collect these hashtags, we tokenized the tweets from the dataset (more details in Section 4.2) and we searched the tokens that contained the keyword "#".

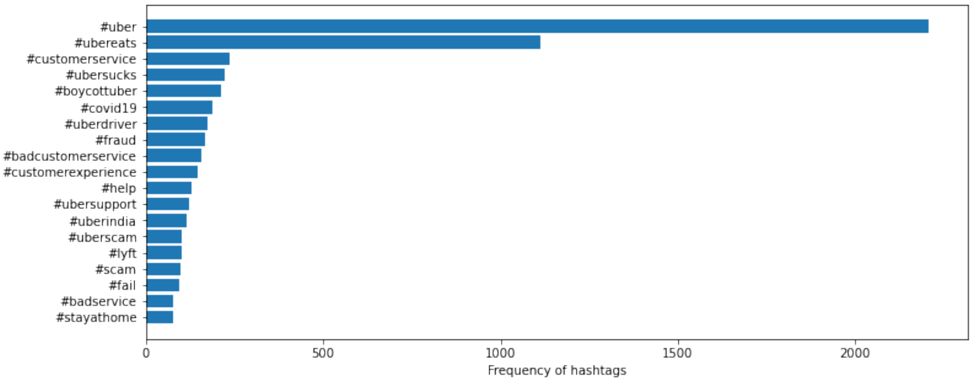


Figure 3.3: Most frequent hashtags in the dataset.

From Figure 3.3 we can see that all the hashtags analyzed are related to Uber Services (#ubereats, #customerservice) and opinions of customers and drivers towards the brand (#ubersucks, #fraud, #badcustomerservice, etc.).

At first instance, we can observe that the majority of hashtags related to the opinions of users seem to provide a negative expression. This result will mean with high probability that the tweets containing these hashtags are going to be reduced into a tweet expressing a negative opinion towards the service.

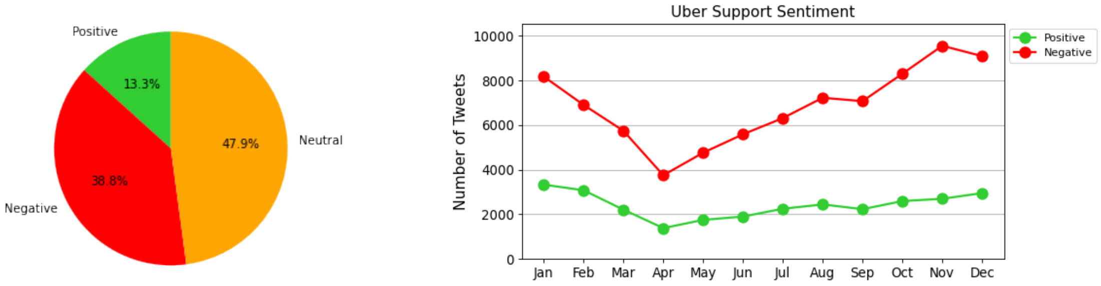
3.2.2.4 Sentiment analysis

Sentiment information has become a fundamental step in every Opinion Mining analysis. In our case, we considered of relevance the sentiment analysis from the collected tweets since these messages posted on Twitter are public opinions that can directly affect Uber brand popularity.

Sentiment extraction can be performed through different methodologies such as Support Vector machines [20], Bayesian network classifiers [21], etc. In this project, we have used the Senpy [9] framework combined with a Docker image which contained the necessary plugins to obtain the sentiment information of the tweets from our dataset by using Sentiment140. To access this framework, we have made several HTTP requests to a specified endpoint (URL). As a result, this tool gave us in the callback the sentiment information of each tweet.

Once this information was collected, each tweet was characterized by a sentiment label expressing a positive, neutral or negative polarity. We added to our Uber Dataframe created in Section 3.2.1 the sentiment information in a new column called "Sentiment".

In Figure 3.4a we grouped the tweets into three main categories according to their predominant sentiment (positive, negative, and neutral) where the **52.1%** of these documents had inner polarity (the resultant **47.9%** of the documents had a neutral expression). Moreover, it is observed that the number of tweets with negative sentiment is predominant (**38.8%**) over the positive ones (**13.3%**). The next step implemented was the elimination the neutral tweets to reduce the common noise produced by them. We decided to filter those tweets and retain only the ones that had a positive or negative sentiment. In Figure 3.4b we describe the monthly volume of tweets that exhibit an inner sentiment value. Furthermore, is seen that negative sentiment is predominant over the positive in all the months during 2020. Also, this figure follows the same tendency as Figure 3.1 where the volume of data during the months from March and April is lower than the rest of the months.



(a) Sentiment percentages

(b) Monthly sentiment of tweets

Figure 3.4: Global sentiment of tweets

Both results analyzed above describes that users who post tweets containing sentiments on the @Uber_Support platform usually express a negative polarity. This result means that customers and drivers are not completely satisfied with the services given by the brand, which is a worrying outcome since this transport company highly depends on user's satisfaction to compete with their competitors. Although we can appreciate that some users post messages with a positive expression, which can mean that some other services provided by this company are positively rated.

However, this first approach performed in the project does not segregate the sentiment information into the different services discussed in the platform. Hence, Chapter 4 will describe in detail the different techniques implemented for obtaining the latent topics and the association of each tweet towards a topic to analyze deeply the sentiment information.

CHAPTER 4

Machine Learning Model and Evaluation

4.1 Introduction

In this chapter we will cover the different machine learning technologies implemented in the project to create a Topic Modelling and a Topic Clustering System from the dataset described in Chapter 3.

Firstly, we will introduce the reader the methodology implemented for cleansing the dataset (Section 4.2). Then, we will focus on the creation of the Topic Modelling System (Section 4.3) followed by their results in the context of the Uber Dataset. After that, in Section 4.4 and Section 4.5 we will describe several Clustering algorithms performed during the project. This algorithms includes the use of K-Means and Genetic Algorithms with their respectively results.

Finally, in Section 4.6 we will combine both Topic and Clustering Systems to summarize the sentiments associated to each topic tweeted by the customers in the platform.

4.2 Data pre-processing

The following section describes the steps implemented for pre-processing our dataset. This data management technique is one of the most important steps to reduce the dimensionality of the corpus and the common noise produced in sentences. In our case, we performed a detailed pre-processing of the tweets in order to improve significantly the results provided in Topic Modelling (Section 4.3).

Once data is collected and analyzed (Section 3.2), we transformed the CSV file into a pandas dataframe where we made a pre-process on the field "Tweet". This column contains each raw tweet in a string format. Each tweet was processed individually through the following steps described below:

- **Tokenization**: This process involves the separation of sentences into smaller units of information (tokens). As a result, each tweet was transformed into a list of separated tokens (words) that computers are able to process more efficiently in the cleansing process. Moreover, all these tokens were transformed into lower case words. The tokenization process has been performed in this project through the NLTK library by using TweetTokenizer module.
- **Removal of characters, numbers and patterns**: This process involves cleansing all the numbers, punctuation marks, emojis, emoticons, URL paths, and symbols from the dataset. Therefore, we filtered all the usernames (tokens starting with the character "@"). Finally, the character "#" was eliminated with the purpose of grouping *hashtags* with the rest of the words from the English dictionary when creating the dictionary of words needed in the machine learning system (Section 4.3.1).
- **Removal of stopwords**: English stopwords were eliminated from the corpus since these words do not provide any semantic meaning. This removing process was performed using the English dictionary of stopwords from the NLTK library.
- **Lemmatization and Part-of-Speech (POS)**: Lemmatization involves the morphological analysis of words to reduce them into their root form. This grouping process of transforming all the inflected words into their lemma will reduce significantly the dimensionality of the corpus. Furthermore, we applied POS-Tagging in these lemmas to retain only nouns, verbs, adverbs, and adjectives, since we consider these grammatical categories as the most relevant in semantic meaning.
- **Removal of unfrequent words**: We performed a counting process in the dataset to determine the frequency of each lemma in our corpus. Then, we filtered the most

unfrequent words from the dictionary to reduce once again the total number of words that our dictionary will be composed of.

- **Stemming**: This process involves the elimination of all the prefixes and suffixes of words to obtain its base form which is known as the stem. The result of stemming might not be a real word, but in most cases it will be possible to interpret the meaning of the stemmed word once performed the pre-processing. Therefore, the stemming process involves less computational time than the lemmatization process, since the stemming process does not analyze the global context of the tweet for cleansing. This process was used in some of the models created in Section 4.3.2 to reduce the tokens from our dictionary in order to check if coherence score results improved. The stemming process was performed following the Snowball method from NLTK library.

- **N-grams**: N-grams refers to the combination of N terms that are presented together in the text. In this project, we implemented bigrams to group pairs of words that are frequently written together. Moreover, to create a bigram in our model, we established a minimum frequency that a pair of words should be written together in the corpus to consider them as a bigram.

Later, we removed all tokens with less than 2 characters since we consider an English word must have at least 3 characters to provide any significant information. Finally, we cleaned all the empty tweets that appeared after this cleansing process which reduced our corpus to 212400 tweets. As a result, we created a dictionary of 7964 tokens (unique words). This dictionary was created by using the corpora dictionary module from Gensim [8] and it will be used to fed LDA model (section 4.3.1).

Furthermore, we consider of relevance to emphasize the importance of following a correct pre-processing order. For example, we should not perform the removal of unfrequent words before lemmatizing. This effect will eliminate some inflected and low-frequency words that can be characterized into one highly frequent lemma. Thus, we give the reader the pre-processing process example followed in the project:

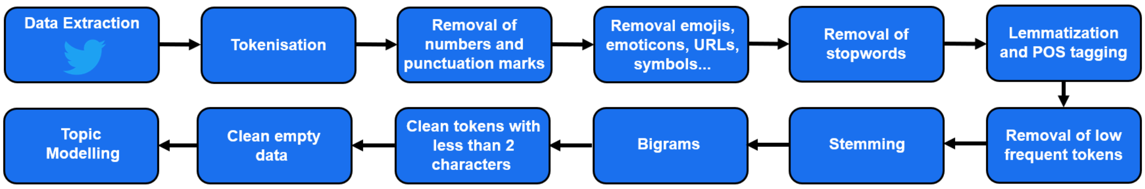

Data Extraction
Tokenisation
Removal of numbers and punctuation marks
Removal emojis, emoticons, URLs, symbols...
Removal of stopwords
Lemmatization and POS tagging
Removal of low frequent tokens
Stemming
Bigrams
Clean tokens with less than 2 characters
Clean empty data
Topic Modelling

Figure 4.1: Pre-processing process.

4.3 Topic Modelling System

Once the data has been pre-processed, we can perform a Topic Modelling system with the purpose of extracting the *latent* topics discussed by Uber users in our dataset. To perform this task, we have decided to use the Latent Dirichlet Allocation (LDA) machine learning algorithm combined with the Gensim library [8]. The following image describes the process described to obtain the *latent* topics of the dataset, which is detailed in the following sections.

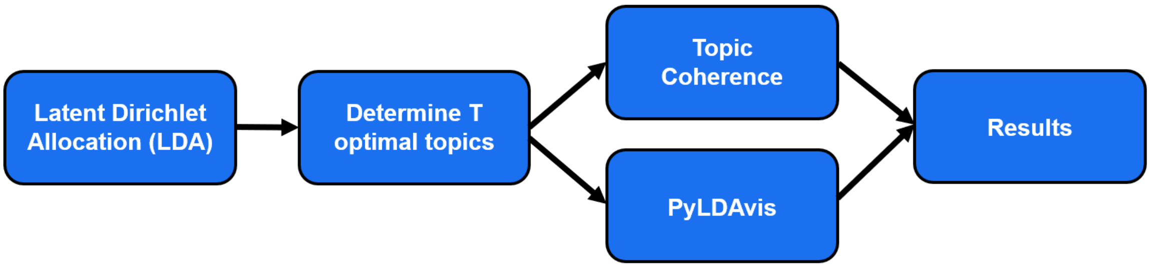


Figure 4.2: Topic Modelling Process.

4.3.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) [22] is an unsupervised[1] machine learning model used in Natural Language Processing for extracting the most discussed topics in the corpus. This algorithm considers that each document is described by a distribution of topics, and each topic can be reduced to a mixture of words based on a frequency to be assigned to that topic. This generative probabilistic model will allow us once more to reduce the dimensionality (the first dimensionality reduction appeared with pre-processing) of the corpus to obtain relevant information for analysis. In this project, we have used LDA to obtain the underlying structure of *latent* topics in our dataset.

Once documents were pre-processed (Section 4.2) and the dictionary of words was created, we transformed the corpus into a bag-of-words by using doc2bow Gensim's module. This bag-of-words was composed of a matrix where each row was characterized as a document. These rows were composed of tuples where the first values represented the position of a word in the dictionary and the second values described the frequency of this word tweeted in the same post. As the last step, we implemented LDA using this bag-of-words, the dictionary and the pre-processed corpus to create the model.

[1] Latent Dirichlet allocation originally works in an unsupervised method, but this does not mean that LDA cannot be performed as a supervised machine learning method [23].

4.3.2 Determining T optimal number of topics

The unsupervised nature of Latent Dirichlet Allocation requires from the user to provide the initial definition of the T number of topics to obtain results. Hence, the user needs to determine this T value to make LDA distribute the words from the corpus into these predefined topics. We can assume that not all T values will be valid to perform a proper Topic Modelling approach.

Therefore, we need to feed the machine learning algorithm with the optimal value of T topics to perform the best result possible. However, our corpus does not have a prior definition of labels or characteristics that can help us to know this T optimal value of topics. For this reason, we need to perform an iterative methodology to determine the optimal value of T by interpreting the distribution of words provided by LDA in every iteration. Furthermore, to perform a correct interpretation of the model, we need to focus on a metric that provides the performance of each iteration from LDA.

In this project, we introduce the concept of Topic Coherence as the metric for determining the optimal value of T to provide the best Topic Modelling results possible. Topic Coherence measures the value of a single topic by interpreting the degree of semantic similarity between high scoring words in the topic [24]. In other words, Topic Coherence gives computers the capacity of measuring how the different words from a selected topic are related between them. There are several Topic Coherence metrics that researchers can use. In this project we have implemented the c_v measure through the Gensim library [8], which provides values from the range [0, 1]. Higher values of c_v score would result in higher semantic similarity between words from the same topic. In contrast, values of c_v close to 0 will mean a lack of semantic similarity between words from the topics. Figure 4.3 describes the iterative methodology mentioned above and it is detailed in the following paragraphs:

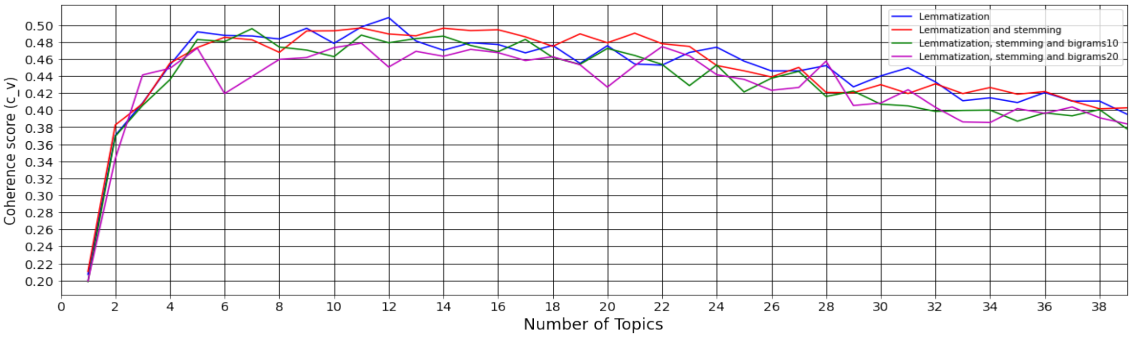


Figure 4.3: Coherence score from LDA models with respect the number of topics.

The previous figure shows the *c_v* Topic Coherence values of different LDA models implemented by iterating T from 1 topic to 40 topics. These models created has different pre-processing techniques (Section 4.2) with the purpose of obtaining a wide variety of results for Topic Modelling from the same dataset to interpret them independently. In our case, these models are reduced into:

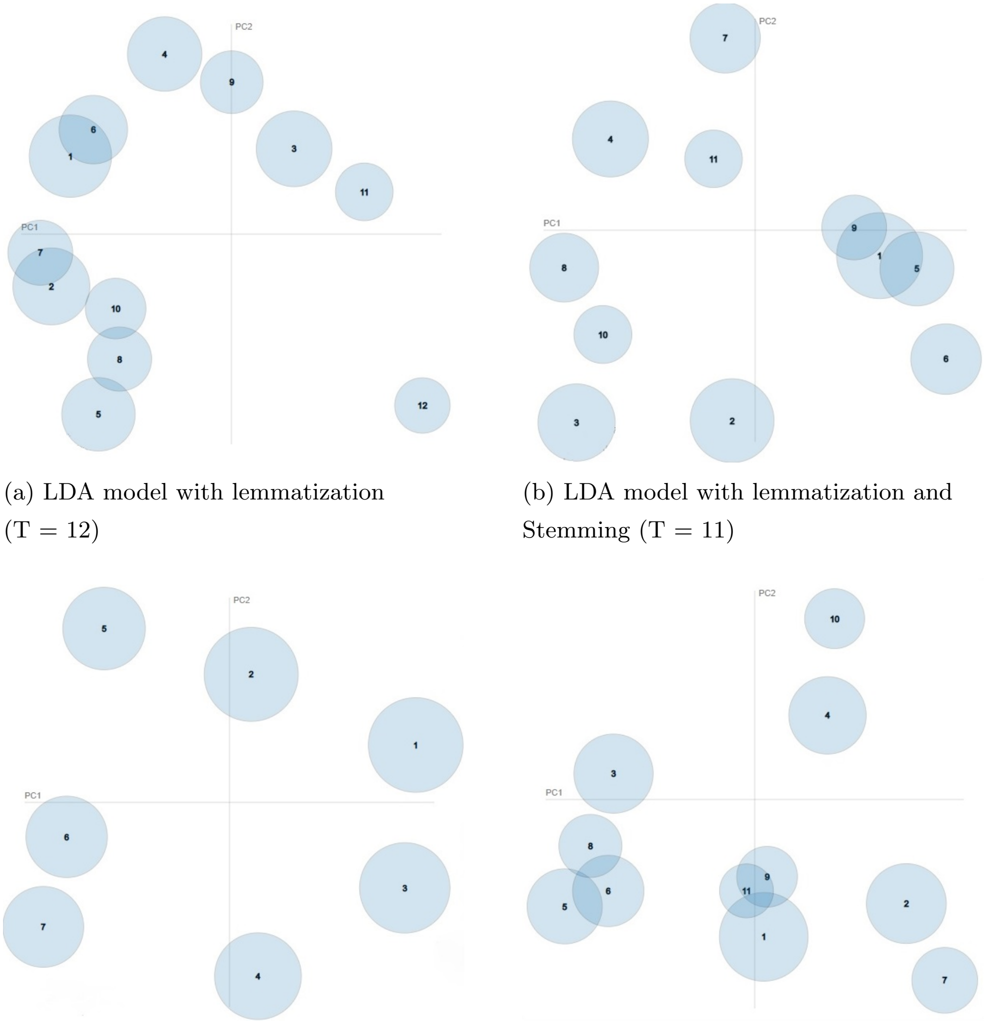
- Lemmatization without stemming nor bigrams.
- Lemmatization and stemming.
- Lemmatization, stemming and variations in the frequency to form bigrams.

We can see in all models from Figure 4.3 that topics between T=2 and T=5 experiments a continuous growth of the *c_v* values which means that the semantic similarity between words from each topic is incremented. Furthermore, models with values between T=15 and T=40 describes a significant decrease on the *c_v* which means that the addition of a high number of topics in the models will result in word distributions through topics that will not have a relation between them. Keeping these two effects in mind, we decided to select the highest *c_v* values in each of the models created as a first approach since they will give us the highest semantic similarity of each model.

It is important to clarify before continuing that *c_v* is a measure that helps to determine the optimal number of topics, but this measurement requires human interpretability to determine if the T topic selected from each model is appropriate. In our particular case, we experimented that some of the models described in Figure 4.3 had higher values of c_v score than others. However, after the individual analysis of each topic (particularly the words this topic contains) from those models, we experimented that some of them could be grouped as the same topic.

Hence, to understand visually how words in the corpus were fitted among the different T topics declared, pyLDAvis [17] library was implemented on those highest values of *c_v* measure of each model. This tool was very useful to give a “name” to each of the T topics presented and to check how the words from each topic were distributed in a 2-dimensional space. Figure 4.4 shows in detail how words are distributed among the space in each LDA models described above.

After analyzing all the pyLDAvis models from Figure 4.4 and the words each topic contained, we decided to select the model where the maximum *c_v* score was located at 7 topics (Figure 4.4c) with a value of 0.4956. This model uses lemmatization, stemming, and bigrams with a minimum frequency to create them to 10 times. The principal reason to select this model was that all its topics (specifically the words they contain) were well separated from each other. For this reason, topics did not have similarities between them and we can ensure that our Topic Modelling proposed is viable for its analysis (Section 4.3.3). Moreover, we can appreciate from this selected figure (Figure 4.4c) with T=7 that all their topics have nearly the same number of words (tokens) from the dataset since all topics have a similar size.



(a) LDA model with lemmatization (T = 12)

(b) LDA model with lemmatization and Stemming (T = 11)

(c) LDA model with lemmatization, stemming and 10 times minimum frequency for making a bigram (T = 7)

(d) LDA model with lemmatization, stemming and 20 times minimum frequency for making a bigram (T = 11)

Figure 4.4: PyLDAvis model representations

4.3.3 Topic Modelling Results

In this section we introduce the results of the Topic Modelling System in our dataset. As mentioned before, these results are based on a LDA model performed with T=7 topics. Each of the topics defined has different and unique word distribution that needs to be interpreted by a human to determine if this word distribution can be considered as a topic.

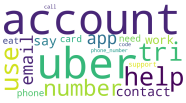
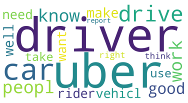
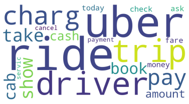
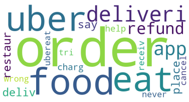
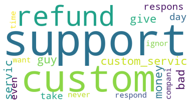
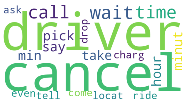
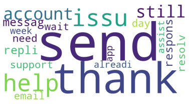
Topic Name	Wordcloud	Topic Description
Uber account/ Uber App		Customers reporting issues when logging into Uber app or validating their accounts.
Uber drivers		Customers reporting issues or giving their opinion about Uber drivers. This topic also includes Uber drivers asking for support about their licences and regulation.
Money and payments		Users contacting with Uber Support platform due to money issues and wrong payments on their accounts.
Uber Eats		Clients contacting with Uber Support to resolve issues related with Uber Eats platform.
Opinions about Customer Service		Tweets sharing the experience about contacting Customer Service or asking for additional support or refunds.
Time and cancellation		Time related issues, including cancellation of the trip by the driver at the last minute and longer than expected waiting time to the driver to arrive.
Contact with Uber Support		Clients or drivers contacting directly with Uber Support to provide alternative ways of contact (email, telephone, direct message...) or answering previous conversations.

Table 4.1: Description of topics and most frequent words.

Therefore, before giving a name to each of the topics, previous research on transport platforms and customer services was realized to understand its global context. As a result, table 4.1 shows these 7 most discussed topics in our dataset with its most frequent words providing a word cloud and a detailed description of each of the topics.

However, this result does not provide the topic of each document but a probabilistic distribution of the words through the topics. As mentioned in Section 4.3.1, documents are composed of a mixture of probabilities of belonging to each topic depending on the words they contain. For this reason, we describe in the following section a Clustering System for grouping documents into several centroid-based clusters to ensure an efficient grouping process of documents that share similarities between them.

4.4 Topic Clustering System

4.4.1 Introduction

As mentioned before, LDA (Section 4.3.1) represents each tweet as a mixture of topic probabilities depending on the words they contain. For this reason, each tweet can be represented as a vector containing the different probabilities to be assigned to each of the topics defined. In our case, this tweet-vector is characterized by a 7-dimensional array of topic-probabilities distribution since T is selected as 7 topics.

Once each document is characterized as a vector of probabilities, we created a matrix of tweets (212400 x 7) where we can use it to place documents into a 7-dimensional space. This dimensional space will be characterized by the representation of the pure topics located on the corners of this space. However, this matrix do not provide the topic of the document but only a distribution of probabilities to the different topics. For this reason, we need to perform clustering techniques in our corpus. Clustering is an unsupervised technique that refers to the task of grouping a set of data points (here LDA probability vectors) that present similar characteristics (in our case topics) into the same group. The following sections will describe several clustering techniques used in this project in order to group documents into several centroid-based clusters. Specifically, we will introduce K-Means algorithm. Later we will describe several Genetic Algorithms (GA) [25] aiming to optimize the clustering results provided in this system.

4.4.2 Evaluation metrics

Before describing the first clustering technique used in the project and its results, we will first need to explain several evaluation metrics used for clustering. This metrics were really useful for evaluating its performance:

- **Inertia (distortion)**: This evaluation metric is based on the concept of the Sum of the Squared Error (SSE) using the Euclidean Distances for describing the error between instances (here LDA coordinates for each document) and their cluster-centroid assigned. Inertia's mathematical expression is described below:

$$Inertia = \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2 \tag{4.1}$$

Equation 4.1 is described as follows: **S** represents the total number of instances in each of the clusters, x_j are the different instances from the total set of **S** documents and μ_i is the cluster-centroid whose Sum of Squared Error (SSE) to the document is minimum.

- **Silhouette Score**: This metric was created by Peter J.Rousseeuw in 1987 for measuring the inter-cluster and intra-cluster separation [26]. Before describing Silhouette Score we need first to know the concept of Silhouette Coefficient that is described in Equation 4.2:

$$S(i) = \frac{b_i - a_i}{max(a_i, b_i)} \tag{4.2}$$

Where S(i) is the Silhouette Coefficient of each instance from the dataset. a_i refers to the mean distance between the i th data point analyzed and the rest of the instances from the same cluster (measuring the intra-clustering distance), and b_i refers to the mean distance between this i th point analysed to the instances from the nearest cluster (measuring the inter-clustering distance). Silhouette Coefficient yields as a result floating values in the range [-1, 1]. Silhouette Coefficients near -1 indicates that the i th data point has been grouped in a wrong cluster, and values near to 1 indicates that the data point is correctly assigned to that cluster and is far from the rest of clusters. As a result, Silhouette Score is calculated as the mean of all S(i) (Silhouette scores) of the data points.

4.4.3 K-Means clustering algorithm

Prior designing the Genetic Algorithm, we decided to implement K-means clustering algorithm to establish a basis and check the viability of our model. This algorithm has been created throughout the scikit-learn [10] clustering package .

4.4.3.1 Introduction to K-means

K-Means clustering algorithm was first implemented by Mac Queen in 1967 [27] and is one of the most widely used algorithms for grouping data into several K coherent groups. K-Means is an unsupervised hard and partitional clustering algorithm where the user needs to define K previous clusters for making the algorithm to converge to an optimal solution. The partitional nature of K-means will result in the creation of K non-overlapping groups where each instance will only belong to a single cluster.

The basic idea of this algorithm is closely related with the optimization the inertia criterion imposed in (Eq 4.1). K-means algorithm is performed as follows: At first instance, the K centroids are defined by randomly selecting K different instances[2]. The data points from the model we are analyzing will be assigned to the clusters whose inertia is the minimum (Eq. 4.1). After this cluster-data point association, centroids (μ_i) will move to the mean of its distribution of data points **S** as explained in Eq.4.3.

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{4.3}$$

This process is repeated until the position of the cluster-centroids does no variate in each iteration, where we can affirm that the algorithm has converged to an optimal solution.

4.4.3.2 Evaluation of K-Means

K-means has been implemented in our project by using the above-mentioned matrix of tweets (Section 4.4) with the aim of clustering these documents into several groups (topics). However, as we mentioned before, K-Means is an unsupervised method that requires from the user the selection of the number of clusters in order to partitionate the space. For this reason, we need first to determine this optimal number of clusters to group our documents (tweets) into several K groups.

Firstly, we followed an iterative process by making variations on the total number of clusters for grouping the tweets. In each iteration, we measured the total value of inertia (Eq. 4.1) once the K-means algorithm has finished in order to determine the optimal number of clusters required in this problem. This heuristic process is called the Elbow Method and it is described in Figure 4.5 where we used the YellowBrick library [18] for plotting the results.

[2] There are many inicialization techniques proposed in K-means. In our case we used K-Means ++ [28]

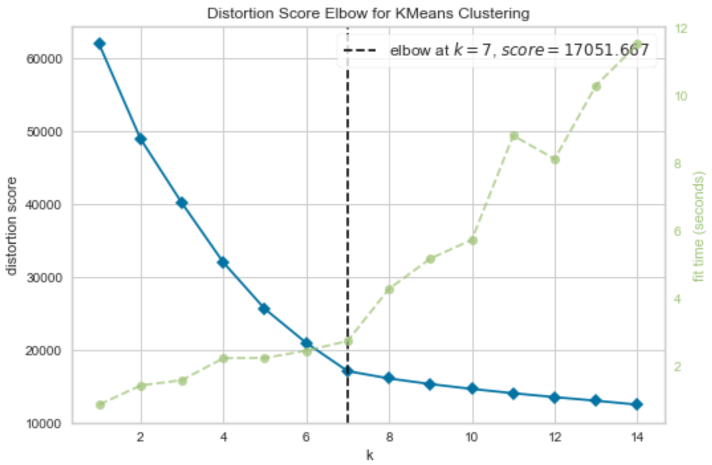


Figure 4.5: Evaluation of the optimal number of clusters with the Elbow Method.

Figure 4.5 shows that the optimal value for clustering the dataset refers to K=7 with a total inertia value of 17051,66. The rationale behind selecting K=7 as the total number of clusters for optimizing the process arises from two main aspects that need to be analyzed in detail. On the one hand, adding more clusters to partionate the space always minimises the distance between documents and their closest cluster since the space contained in each cluster is reduced. On the other hand, appart from the first effect mentioned, this reduction is more abrupt in the first values of K since adding more clusters to the problem also allows them to be distributed more efficiently through the space. Hence, there is an optimal K value (in our case K=7) where this abrupt slope changes its tendency. The selection of higher K values from this inflected result will reduce inertia values but an inefficient clustering result will be obtained (where some clusters will be overlapped). Finally, from this figure, we can also see that the time used for evaluation inertia in each K group of clusters is incremented when adding more clusters to the problem. This result is expected since K-means has to analyze the distance between all documents and clusters to determine which cluster has the minimum distance to each document in each iteration. Consequently, adding more clusters in the algorithm will result in more operations that needs to be done.

As we have seen with this previous analysis, the Elbow Method Curve (Figure 4.5) is a process that can yield bad interpretations since inertia values decrease when we add more clusters to the process. Hence, this method highly depends on the interpretation of the user for determining the optimal value of K that describes the problem. To solve this problem, we also implemented the Silhouette Score (Section 4.4.2) which commonly provides more precise results for selecting the optimal number of clusters to partitionate the space.

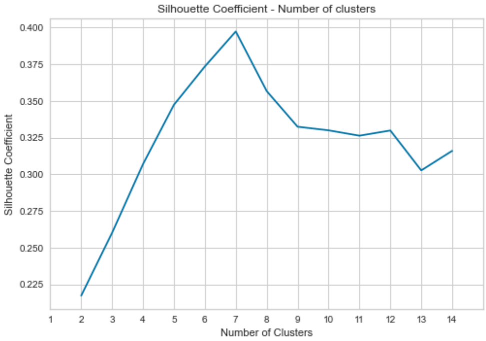


Figure 4.6: Silhouette score.

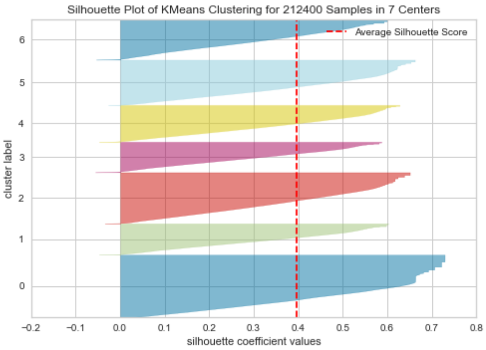
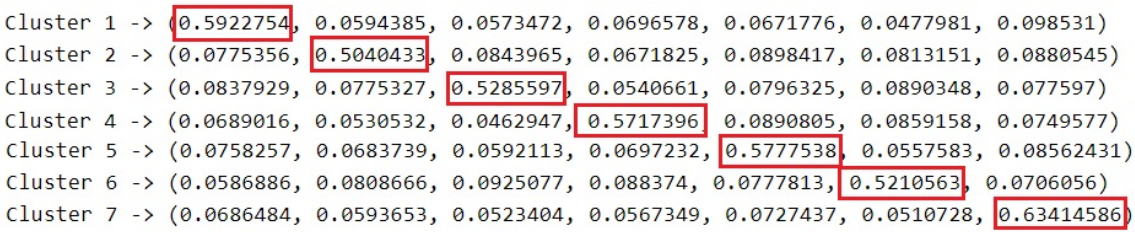


Figure 4.7: Silhouette diagram (K=7).

In Figure 4.6 we followed the same methodology as in Figure 4.5 by increasing the number of clusters to obtain the optimal value of K, but this time using the Silhouette Score. As a result, we obtained that the optimal number of clusters for grouping our dataset is finally K=7 with a value of 0.3972 of Silhouette score. The combination of both Inertia and Silhouette scores concludes with the correct interpretation of using K=7 clusters for partitioning our dataset. Moreover, we created a Silhouette diagram for this seven optimal clusters for evaluating individually each cluster's Silhouette score (Figure 4.7). We can see that cluster 0 is well separated from the rest of the clusters since its Silhouette Score is higher than the rest of the clusters, and cluster 1 and 3 has less documents than the rest of the clusters.

4.4.3.3 K-Means results

At this point, we have analyzed the impact of K-means in our dataset by measuring its performance with the Inertia and Silhouette metrics. We concluded the analysis of this algorithm with a distribution of 7 clusters to partitionate the 7-dimensional space. This clusters have associated a centroid which defines the media of all the documents associated to it. However, in order to see if this cluster-centroids can represent a topic created in LDA (Section 4.3), we first need to see how they are distributed in the space.



```
Cluster 1 -> (0.5922754, 0.0594385, 0.0573472, 0.0696578, 0.0671776, 0.0477981, 0.098531)
Cluster 2 -> (0.0775356, 0.5040433, 0.0843965, 0.0671825, 0.0898417, 0.0813151, 0.0880545)
Cluster 3 -> (0.0837929, 0.0775327, 0.5285597, 0.0540661, 0.0796325, 0.0890348, 0.077597)
Cluster 4 -> (0.0689016, 0.0530532, 0.0462947, 0.5717396  0.0890805, 0.0859158, 0.0749577)
Cluster 5 -> (0.0758257, 0.0683739, 0.0592113, 0.0697232, 0.5777538, 0.0557583, 0.08562431)
Cluster 6 -> (0.0586886, 0.0808666, 0.0925077, 0.088374, 0.0777813, 0.5210563, 0.0706056)
Cluster 7 -> (0.0686484, 0.0593653, 0.0523404, 0.0567349, 0.0727437, 0.0510728, 0.63414586)
```

Figure 4.8: K-Means cluster-centroids.

It can be seen from the previous figure that this cluster-centroids are uniformly distributed in the 7-dimensional space since their predominant component (the probability values which are highlighted in Figure 4.8) are not repeated in any of the centroids. It is important to remember that in this clustering problem, the pure topics are located at the corners of the 7-dimensional space. Hence, we can ensure that this cluster-centroids has been distributed efficiently to group documents (tweets) that share similar characteristics (in this case topic values) resulting in a cluster which can be defined as a topic.

As a result from the clustering process imposed by K-Means, each document is grouped into a single cluster which defines an unique topic. For this reason, we can give each document from the corpus a label that can define its topic. This label represents the cluster whose Euclidean Distance to this document is the minimum deviation possible with a configuration of K=7 clusters. To conclude with K-means, Figure 4.9 shows in a pie chart the percentage of documents assigned to each topic-cluster.

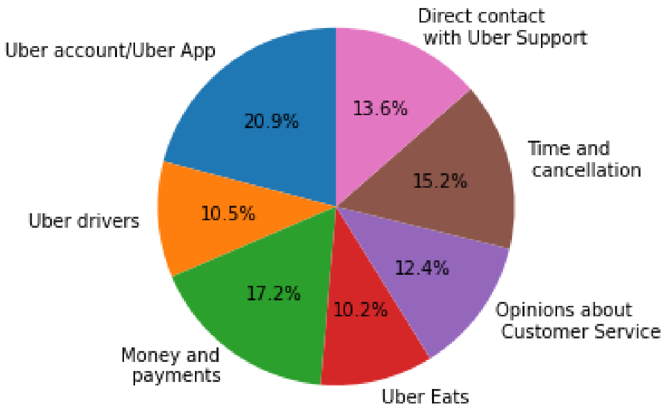


Figure 4.9: K-Means distribution of documents.

It is seen that the topic related to ***"Uber account/Uber app"*** (**20.9%**) is the most discussed in the dataset. This result is expected since all Uber's services highly depend on the use of smartphones and technology to interact between customers and the services provided by Uber company. Hence, customers can experience some problems with the app, which they will have to communicate to the brand through this support platform. Moreover, there is a worrying result with the topics related to ***"Money and payments"*** (**17.2%**) and ***"Time and cancellation"*** (**15.2%**). This result may mean that some customers (specifically, the **32.4%** of the tweets) are experiencing issues related to money and cancelled trips, which can result with high probability in a bad experience where customers will not feel comfortable using Uber services again. Finally, we can see that some of Uber's secondary services, such as ***"Uber Eats"***, has a relevant influence on the dataset (**10.2%**).

4.5 Genetic Algorithms

Once we have created a Topic Clustering System in Section 4.4, the aim of this section is to perform Genetic Algorithms aiming to optimize the K-Means approach. Firstly, we will introduce the reader the concept of Genetic Algorithms and their relation with biology. Later, we will describe several study cases implemented in the project to create a Topic Clustering System based on this algorithms.

4.5.1 Introduction to Genetic Algorithms

Genetic Algorithms (GA) [25] are evolutionary processes inspired by the ideas of natural selection, genetic evolution and biology. These algorithms have been applied by researchers in many study fields such as economics [29], biology [30], and Engineering [31, 32].

In biology, natural selection is based on choosing those individuals who present the best characteristics for adapting to the environment they are living in. Hence, these individuals will be able to survive in the environment, and therefore, they will be the ones selected in the mating of the species. Consequently, the process will create new descendants who will combine both parents' inherent and ideal genomes. During this mating process, there is the possibility that this new offspring manifests mutations in their genes that will make them genetically different from their parents. Thus, the random mutations on the genomes of descendants combined with the characteristics of their predecessors can cause the possibility of creating new offspring that could be able to adapt more efficiently to the environment. This natural selection process repeated during the entire life of individuals, and new offspring will conserve the genetic line of their predecessors.

The Genetic Algorithms used in this project takes as a basis the above-mentioned evolutionary process. Firstly, these algorithms have to create individuals who composes a population that evolves through the algorithm's different generations. However, in Genetic Algorithms, the criterion used to determine if an individual from a population is fitter than another is through its evaluation by taking as a reference an objective function defined by the user. Hence, the fitness function will indicate numerically how optimal is the analyzed individual from the population in each generation. Moreover, several operators are used in genetic algorithms to resemble the idea of natural selection and evolutionary processes of selection, mating and mutation explained above. These genetic operators will be explained in detail in Sections 4.5.1.1, 4.5.1.2 and 4.5.1.3.

In conclusion, the Genetic Algorithms described in the project takes as a basis the idea from Figure 4.10. In order to finish the algorithm, the user will need define a stop criteria. Some of the most common stop criteria standards are: a predefined number of generations, the achievement of an individual for performing a determined value in the fitness function, etc. As a result, we will obtain the fittest individual from the last population.

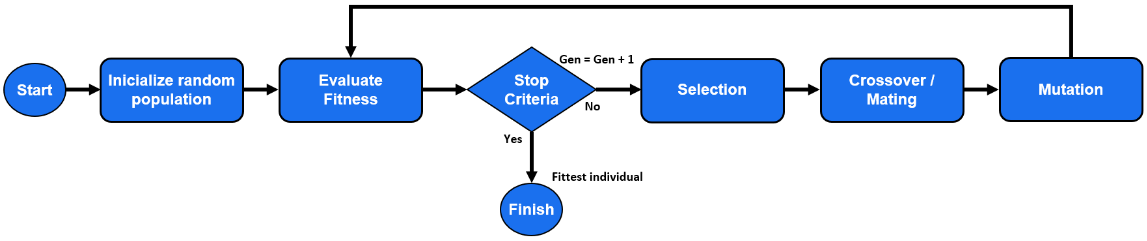


Figure 4.10: Genetic Algorithm process.

Genetic algorithms has several pros and cons that should be analyzed before using these techniques to solve the problem described. On the one hand, the nature of genetic algorithms is based on creating a high number of individuals that will explore the search space through the generations. Hence, their solutions can provide fit individuals located on the global minimum of nearby this solution. For this reason, these algorithms are perfect techniques for global convergence solutions.

On the other hand, the random characteristic imposed in the mutation operator makes GA bad techniques for searching the local convergence of the problems. Due to this problem, if a local convergence solution is needed to be applied in the problem, several techniques can be performed, such as the Method of Steepest Descent or the Newton-Raphson Method. These techniques are based on determining the gradient of the fitness function for making the solution move in the direction where a local minimum is located for performing a local convergence optimization.

4.5.1.1 Selection operator

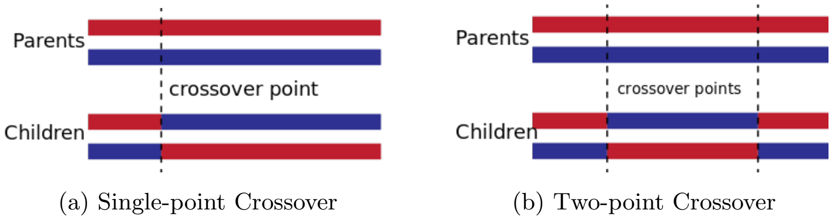
This operator is used in GA to accelerate the convergence of the evolutionary process by selecting, in the majority of the cases, the fittest individuals from the population before performing the crossover. There are several methods to describe the selection process. In our case, we have used the tournament selection which is described below:

- **Tournament selection**: this deterministic process chooses N individuals from the population randomly to compete for entry into the mating pool. The winner of the tournament (the fittest individual) will be the one who will perform the crossover.

In this project, the tournament selection is repeated S times where S is the size of the population. Moreover, it is necessary to select an appropriate value of N for performing the global convergence. On the one hand, choosing high N values will increase the selection pressure and may result in premature convergence. For example, choosing N as the size of the population (S) will result in selecting the fittest individual from the initial population in every iteration from the algorithm resulting in a bad exploration of the space. On the other hand, the selection of low N values will slow down significantly the convergence of the genetic algorithm [33]. For example, if we select the limit case with N=1, the selection process will describe a problem where the fittest individual is always chosen randomly.

4.5.1.2 Crossover operator

The crossover operator involves creating new descendants that share the characteristics (genes) of their parents. Hence, this process creates offspring that will guarantee to maintain the fittest genes from the previous population. Some examples used for performing the crossover in genetic algorithms are the Single-point crossover and the k-point crossover, where both randomly divide the parental genes to create new offspring. The following figure describes these two crossover methods visually.



(a) Single-point Crossover

(b) Two-point Crossover

Figure 4.11: Crossover methodologies.

4.5.1.3 Mutation operator

The mutation operator is applied after recombining the parental genes for making new offspring that presents different characteristics from their predecessors by changing some of their genes with a certain probability. The mutation is used in Genetic Algorithms to avoid its population converging into a local minimum solution by making their individuals explore the search space to converge into the global minimum (or a near solution) of the problem. For this reason, the mutation rate has to be adjusted to balance between exploration (searching for the global minimum) and exploitation (making the algorithm to converge into an optimal solution).

High mutation rates result in more exploration of the space but prevent the population from converging to a global solution, while low mutation rates can result in premature convergence.

It is essential to highlight that the mutation operator has to be defined, in most cases, by the user depending on the problem described. In our case, we will discuss in the following sections the different mutation operators used for clustering.

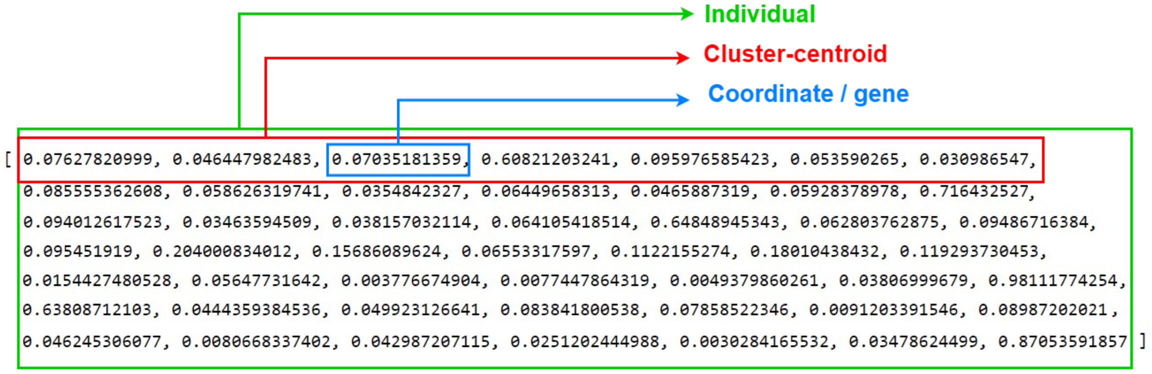
4.5.2 Genetic Algorithm case of study

Keeping in mind the above-mentioned introduction section, it is necessary to define an initial population that will satisfy the clustering problem. Moreover, an objective function will be created to define the fitness of the candidate solutions in every generation.

We will focus on the optimization of K=7 clusters since K-means evaluation yields to 7 clusters as the best solution for clustering the dataset (Section 4.4.3.2).

4.5.2.1 Initial Population Description

The creation of the initial population is the first step to take in mind in every genetic algorithm, and it is different depending on the problem that has been proposed. In our case, as we are describing a clustering problem, this population is composed of individuals that represent a predefined number of cluster-centroids. Figure 4.12 shows an example of an individual performed during the project with a detailed explanation.



```
[ 0.07627820999, 0.046447982483, 0.07035181359, 0.60821203241, 0.095976585423, 0.053590265, 0.030986547,
  0.085555362608, 0.058626319741, 0.0354842327, 0.06449658313, 0.0465887319, 0.05928378978, 0.716432527,
  0.094012617523, 0.03463594509, 0.038157032114, 0.064105418514, 0.64848945343, 0.062803762875, 0.09486716384,
  0.095451919, 0.204000834012, 0.15686089624, 0.06553317597, 0.1122155274, 0.18010438432, 0.119293730453,
  0.0154427480528, 0.05647731642, 0.003776674904, 0.0077447864319, 0.0049379860261, 0.03806999679, 0.98111774254,
  0.63808712103, 0.0444359384536, 0.049923126641, 0.083841800538, 0.07858522346, 0.0091203391546, 0.08987202021,
  0.046245306077, 0.0080668337402, 0.042987207115, 0.0251202444988, 0.0030284165532, 0.03478624499, 0.87053591857 ]
```

Figure 4.12: Individual representing a solution based on 7 cluster-centroids.

The individual shown in Figure 4.12 represents a possible solution from the problem of clustering. In this case, this individual comprises 49 different floating numbers that satisfy the constraint that all their genes are values between the range [0,1]. These coordinates are combined in groups of 7 values to define the dimensions of a cluster-centroid. The rationale behind the selection of creating groups of 7 coordinates to form a cluster arises from the definition of the optimal number of topics in Section 4.3 resulting in 7 different topics that describe a 7-dimensional space of probabilities from the range [0,1].

4.5.2.2 Fitness Function

The candidate solutions described in Section 4.5.2.1 are evaluated individually through the fitness function to compare them in the search space. In our case, the definition of the fitness function describes a problem of minimizing the distance between the individuals (cluster-centroids) and their assigned documents (tweets) by using the Euclidean Distances. Specifically, this fitness function is defined as the mean squared error (MSE) of the Inertia function (Section 4.4.2). The following mathematical expression (Eq. 4.4) describes in detail the evaluation function used in the genetic algorithm for clustering documents:

$$Fitness = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\|x_j - \mu_i\|^2 \tag{4.4}$$

Equation 4.4 is described as follows: K defines the total number of clusters used for partitioning the space, $|S_i|$ refers to the number of documents whose Euclidean Distance to a cluster is the minimum (the K-ith cluster), x_j defines the LDA coordinates of a document contained in the S-ith group of documents and μ_i are the coordinates of the K i-th cluster.

The evaluation function described above considers in its analysis the total number of documents of the S group for normalizing the solution. As a consequence, all clusters have the same importance in the evaluation process, since the distance error between the documents located in the S i-th group and their closest cluster-centroid are divided by the total number of documents located in that cluster ($|S_i|$). This means that each cluster has an importance of 1/K from the total value of the fitness function independently on the number of documents these centroids are assigned to. Moreover, this fitness function describes the constraint that the number of documents contained in each S i-th group must be a value higher than 0 documents for performing clustering.

At first instance, the Inertia function (Eq. 4.1) was considered before performing the fitness function described above. However, the Inertia function describes the Sum-of-Squared-Error of all documents to their associated cluster without normalizing it by the total number of documents ($|S_i|$) in each cluster. For this reason, this function does not analyze each cluster with the same importance but only the sum of Euclidean Distances of all documents to their closest cluster.

4.5.3 First Case of Study

The first study case implemented in the project takes into consideration the use of a random population following the schema from Section 4.5.2.1 to be evaluated through the fitness function described in Section 4.5.2.2. Its main implication in the project involved the exploration of the 7-dimensional space by performing random searches using the different genetic operators (selection, crossover and mutation) to minimize the result of the K-Means approach while at the same time represent a valid Topic Clustering System.

The results provided by the first study case mathematically described a solution that optimized the distance between documents and their closest cluster in comparison to the K-Means algorithm. However, the representation of their clusters throughout the space did not describe the feasibility of a Topic Clustering System since the best individual's clusters did not represent a unique topic but an overlapped result of several clusters in the same topic.

For this reason, the possibility of having created a genetic problem whose individuals are moving through invalid positions of the space was considered. Hence, as described in [34], in order to perform a Topic Clustering system with GA, all cluster centers have to be placed within the T-dimensional standard simplex (in our problem, a 7-dimensional simplex). The following mathematical expression describes the concept of the standard simplex.

$$\left\{ x \,\epsilon\, R^k : x_0 + \cdots + x_{k-1} = 1,\, x_i \geq 0 \text{ for } i = 0, \ldots, k-1 \right\} \tag{4.5}$$

Each cluster (here composed of k=7 dimensions) must satisfy the constraint that all their coordinates (x_i) have to sum a probability value equals to 1. Hence, the second study case performed in this project will consider the use of placing all the clustering solutions within the 7-dimensional simplex. To do this, we will define an initial population and genetic operators that satisfy the probability simplex constraint by following the methodology described in [34].

In order to provide the reader with a complete description of this first study case, we included it in the **Appendix D** with a detailed description of the different operators, results and conclusions.

4.5.4 Second Case of Study

This section describes the second case of study implemented in the project, which considers the constraint that all the clustering solutions must be in the 7-dimensional simplex. To do this, we will follow in its majority the methodology described in [34] to define an initial population and the genetic operators to move within the simplex. Later, we will conclude with the results provided by using this genetic algorithm combined with a local convergence technique.

4.5.4.1 Description of operators

The **initial population** of this study case follows the scheme described in Section 4.5.2.1 and is divided into two groups. Both groups satisfy the constraint imposed by the simplex (the sum of each cluster coordinates is equals to the unit). The first group of individuals defines clustering solutions located on the corners of the 7-dimensional space where each individual is composed of non-overlapping clusters. The second group of individuals are linear combinations from individuals randomly selected from the first group. This combination of clustering solutions is preceded by the following equation:

$$C_3 = C_1 * R + C_2 * (1 - R) \tag{4.6}$$

Where R is a floating value from the range (0,1), C_1 and C_2 are clustering solutions from the first group of individuals that creates a new cluster (C_3) which is also placed within the 7-dimensional simplex. In our case, the R value is composed of an array of three different values (specifically, 0.25, 0.50 and 0.75) which creates three new individuals in every combination from the first group.

The **selection process** is preceded by the tournament selection. We selected N=5 in order to obtain the fittest individuals from the population but without making high selection pressure on the convergence of the algorithm (more details explained in Section 4.5.1.1).

The **mating process** is performed with a certain probability rate (*P(crossover) <p1 = 0.8*) where two parental individuals are selected to perform Equation 4.6 to create new offspring (specifically, we created two R random instances in order to create two new individuals from this parental recombination).

The **mutation process** will modify these new individuals with a certain probability rate ($P(mutate\ offspring) < p2 = 0.35$). Moreover, mutation is composed of two main techniques: the first one takes place the 40% of the times an individual is mutated and the second technique is performed the 60% of the times. The **first mutation process** involves the exchange of two random coordinates from the cluster of an individual with a certain probability rate ($P(first\ mutation) < m_{prob} = 0.2$). The **second mutation operator** involves the replacement of the individual selected to be mutated into a new individual which is created as follows: firstly, the best individual from the population is selected. Then, for each cluster of this best individual, we create a segment between the nearest and the farthest document from this cluster solution. Finally, this cluster will move towards the line created between the center of the previous segment created. The maximum distance this cluster will move depends on a random probability value between the range (0,1) that will determine the percentage of distance to the center of the segment. This process is repeated for all of the clusters from the individual.

As we can see, all the processes mentioned above guarantee the creation of clustering solutions that moves towards the probability simplex. In order to provide a summary to the reader, the following table provides the different operators and values involved in this study case:

Parameter	Methodology	Value	
Individuals	Corner solutions	Population = 10	
	Combination of corner solutions with Eq.4.6	Population = 90 ; R=(0.25, 0.5, 0.75)	
Selection	Tournament Selection	N = 5	
Crossover	Combination of individuals with Eq.4.6	p1 = 0.8 R =>two random values between (0,1)	
Mutation	First technique	p2 = 0.35	p(firstTechnique) = 0.4 m_prob = 0.2
	Second technique		p(secondTechnique) =0.6

Table 4.2: Genetic operators in the second case of study.

4.5.4.2 Results

We iterated the genetic algorithm a high number of generations for allowing the population to converge into the global minimum of the problem described by the fitness function (Section 4.5.2.2). The following figure shows the best individual from each iteration of the algorithm and the average fitness values of its population.

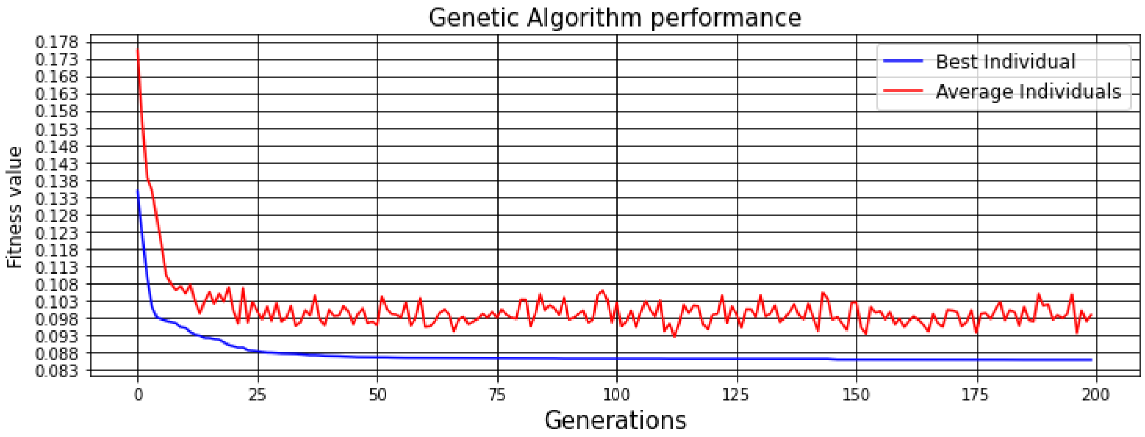


Figure 4.13: Genetic Algorithm convergence.

From Figure 4.13 it is seen that the convergence process is faster during the first iterations in comparison with the last generations. This convergence tendency occurs because the first individuals from the population can easily find solutions in the space which are better than their parents. However, during the last generations, the result starts to describe an asymptotic convergence because the new offspring generated are not able to obtain better solutions than their parents.

Once captured the best individual at the iteration 200[3], we assessed in Table 4.4 both genetic and K-means algorithms through the fitness function (Eq. 4.4). We can appreciate that the K-means algorithm performed a lower fitness value, which means that their distances between documents and cluster-centroids are more optimized in comparison to the Genetic Algorithm.

Algorithms	K-Means	GA
Fitness	**0.08262**	0.08576

Table 4.3: Evaluation of K-means and Genetic Algorithm of the second case of study

However, as we mentioned in Section 4.5.1, Genetic Algorithms aims to search for the global convergence of the problem, but in some cases, their solutions may be near this

[3] In this study case, the optimal fitness value obtained by GA was 0.08576. However, several implementations of this GA has been performed in the project prior to obtaining this fit individual. For further details, we encourage the reader to see Appendix E.

optimal value. For this reason, we also applied a local convergence algorithm aiming to optimize the value provided in GA from Table 4.4. This local convergence algorithm was followed by the Sequential Least Squares Programming (SLSQP) package from the Scipy library [12].

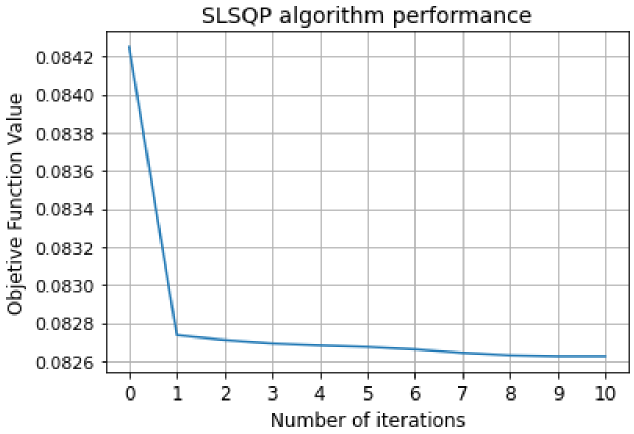


Figure 4.14: SLSQP algorithm performance.

In order to feed this machine learning algorithm, we introduced the best individual obtained from GA, the fitness function to optimize (Eq. 4.4), the step size ($h = 1 * 10^{-4}$) to calculate the finite derivatives to obtain the gradient of the function and the linear constraints to satisfy that all clustering solutions from the individual must be placed on the 7-dimensional simplex. As a result, Figure 4.14 describes the best solutions achieved in the algorithm at iteration 10 with a fitness value of 0.08262.

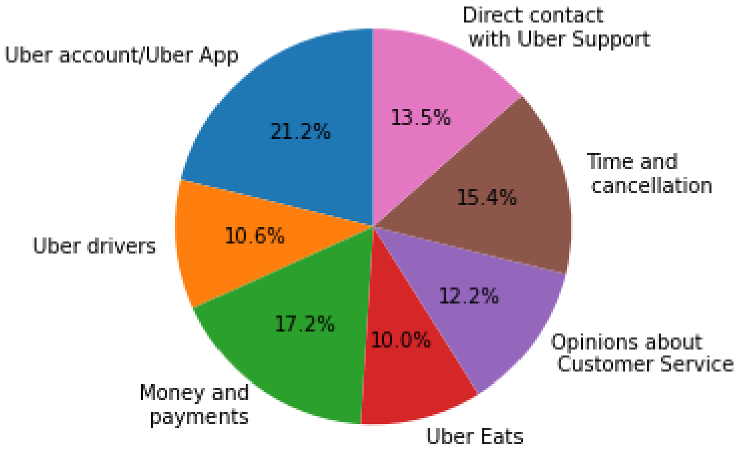
4.5.4.3 Conclusion

To conclude the second study case, we can analyse the results provided in Section 4.5.4.2 with two different perspectives: On one hand, as both algorithms (K-Means and hybridized GA) describes the same mathematical performance, we can assume that their solutions have reached the global minimum of the problem.

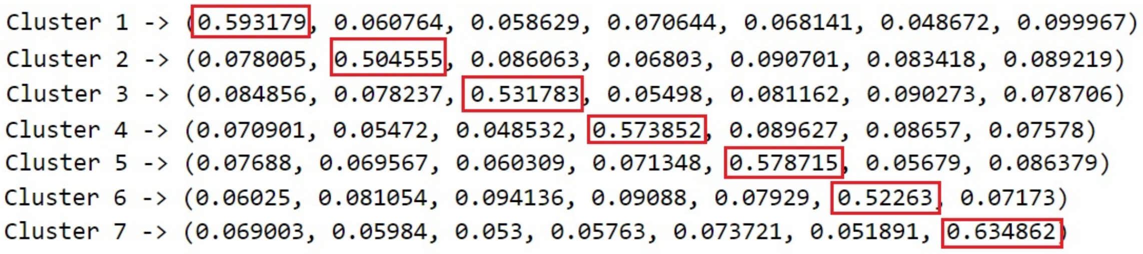
Algorithms	K-Means	GA	Hybridized GA
Fitness	**0.08262**	0.08576	**0.08262**

Table 4.4: Evaluation of K-means, Genetic Algorithm and Hybridized GA

On the other hand, contrary to the first study case (Section 4.5.3), the solutions given by the hybrid process (genetic and local convergence algorithms) describes a valid Topic Clustering solution where each cluster can be defined as a topic since their cluster coordinates are homogeneously distributed throughout the 7-dimensional space as we can see in Figure 4.15b. Moreover, we provide the reader as in Section 4.4.3.3 a pie chart containing the different documents associated to each cluster/topic in Figure 4.15a which describes the same tweet distribution as defined and explained in K-means solution.



(a) Pie chart solution



```
Cluster 1 -> (0.593179, 0.060764, 0.058629, 0.070644, 0.068141, 0.048672, 0.099967)
Cluster 2 -> (0.078005, 0.504555, 0.086063, 0.06803, 0.090701, 0.083418, 0.089219)
Cluster 3 -> (0.084856, 0.078237, 0.531783, 0.05498, 0.081162, 0.090273, 0.078706)
Cluster 4 -> (0.070901, 0.05472, 0.048532, 0.573852, 0.089627, 0.08657, 0.07578)
Cluster 5 -> (0.07688, 0.069567, 0.060309, 0.071348, 0.578715, 0.05679, 0.086379)
Cluster 6 -> (0.06025, 0.081054, 0.094136, 0.09088, 0.07929, 0.52263, 0.07173)
Cluster 7 -> (0.069003, 0.05984, 0.053, 0.05763, 0.073721, 0.051891, 0.634862)
```

(b) Cluster centroids solution

Figure 4.15: Genetic Algorithm second study case results

4.6 Sentiment analysis

4.6.1 Sentiment of words and phrases

Once performed the pre-processing of our dataset (Section 4.2), we have filtered the most valuable words from each tweet. In this section, we are interested in determining if it is possible to describe the polarity of the words and phrases from the corpus created. We can assume at first instance that these words individually may not represent (in most cases) a sentiment value. However, their placement on negative tweets and positive ones could describe a vocabulary of words with polarity in the context of both transport and customer services.

To do this, we took as a reference the frequency of these words in both positive and negative tweets. For this reason, we characterized each word (or phrase) into a negative or positive one depending on its frequency on the corpus. To understand visually how these words are frequently repeated in the two categories (positive and negative), we created a scatterplot by using the Scattertext library [19].

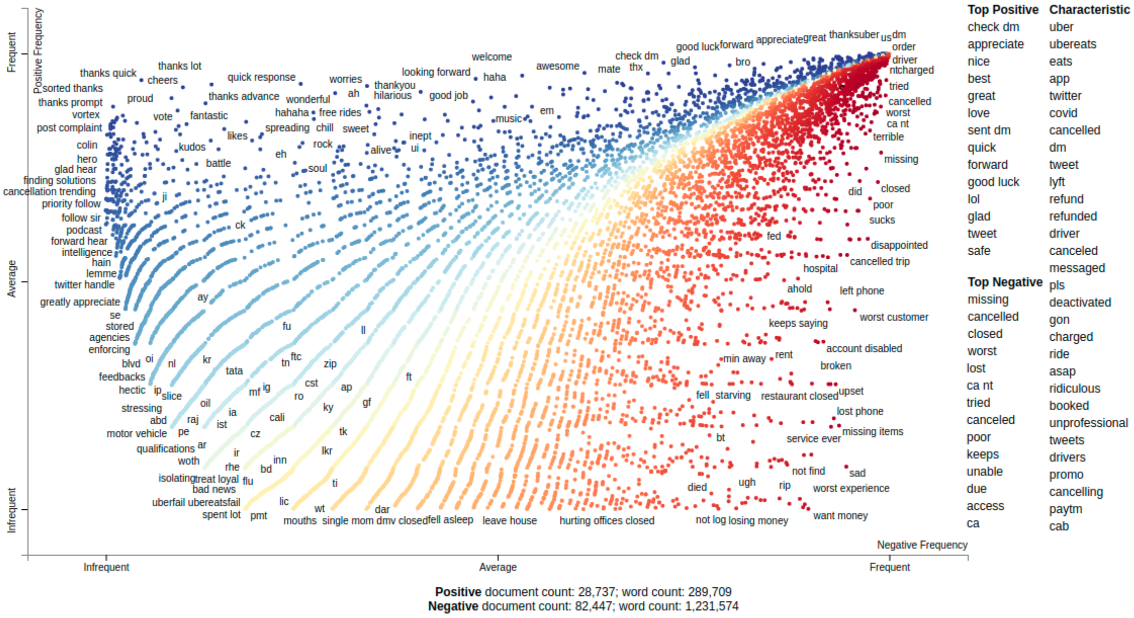


Figure 4.16: Frequency of words for both Positive and Negative categories

Figure 4.16 shows the distribution of words into a 2-dimensional space where their axis represents the frequency of the positive and negative categories for each word. This figure shows relevant results that need to be interpreted:

On one hand, we can see that some phrases and words such as **“quick response”** and **“worries”** are highly frequent in **positive tweets**. This means that we can associate these terms with a positive sentiment value in the context of transport services. On the other hand, there are several expressions such as **“want money”, “restaurant closed”** or **“cancelled trip”** appears with higher frequency in the **negative tweets** from our dataset. This means that these phrases are commonly used in negative tweets referring to our transport service dataset.

However, there are several words such as **“thanksuber” or “driver”** which are contained with high frequency in both categories. This means that these words are usually written on both negative and positive tweets. For example, the first term (**“thanksuber”**) can be associated with both categories since users may post this word using an ironic expression providing a negative opinion towards the platform, or conversely, they are expressing their gratitude to the response provided by the service.

Finally, it is important to highlight that the words described in Figure 4.16 may not represent a sentiment value if we analyze them individually. However, due to their distribution throughout the tweets, these words may represent a sentiment value in the context of our dataset. For example, the word “closed” does not represent a sentiment value if we analyze it individually, but in the context of our transport dataset, this word is highly repeated in the negative tweets which may refer to a disappointed expression on several canceled trips.

4.6.2 Sentiment of topics

Once collected the sentiment information from the users tweeting to the Uber Customer Support Service (Section 3.2.2.4), we summarized this high volume of information by grouping the tweets into the different topic-clusters defined by the combination of the Topic Modelling System (Section 4.4) and the Genetic algorithm created for Topic Clustering (Section 4.5). As a result, Figure 4.17 shows the sentiment information of each topic from the dataset.

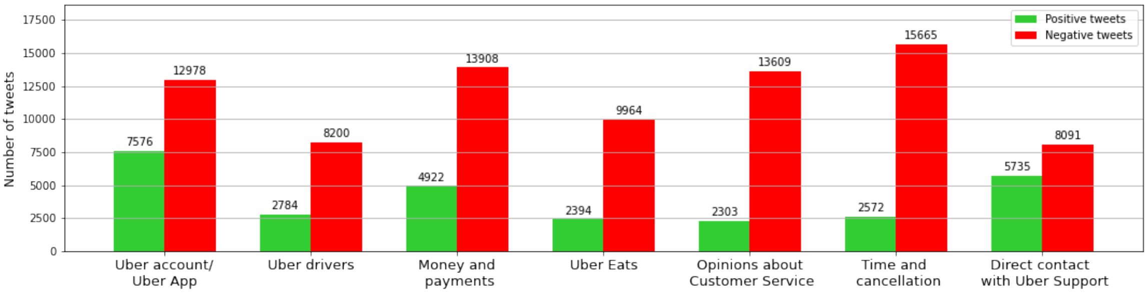


Figure 4.17: Sentiment associated to topics using the Genetic Algorithm for clustering.

From the previous figure (Figure 4.17), we can conclude two main results by interpreting the sentiment values the systems has provided.

Firstly, it is observed that all the discussed topics in the dataset show more negative tweets than positive ones. This result means that Uber users address this platform (@Uber_Support) to tweet about complaints and problems with a negative expression on their posts with more frequency than using positive expressions independently of the topic they are referring to. Specifically, the tweets related to the ***"Time and cancellation"***, ***"Opinions about Customer Service"*** and ***"Uber Eats"*** topics present the highest negative sentiment values since these topics show the highest difference between negative tweets and positive ones. Based on these results, we can affirm that users whose posts contain an opinion or sentiment have a negative stance on these specific topics. This worrying result could result in a lost of customers, which will use Uber's competitors to satisfy their demands. In conclusion, we believe that Uber should consider the modification of both cancellation and food delivery policies to fulfil the customer's needs if yet is not on their business model policies.

However, we can also appreciate several topics has a proportion of positive and negative tweets intimately close. Hence, we believe that some of these most discussed topics make Uber users feel more satisfied than others. Some examples of these topics are "***Direct contact with Uber Support***" and "***Uber account/Uber app***". From the first topic (***"Direct contact with Uber Support"***) we believe that the rationale of having high positive tweets arises from the answers of previous conversations with the platform (eg. "thank you @Uber_Support for the response :)") where the word ***"thank"*** is highly frequent in this topic as described in Table 4.1. Concerning the topic ("***Uber account/Uber app***"), we can assume that people are not completely satisfied with the services provided by the Uber App. However, the high values of positive tweets on this topic could mean that many users do not have problems with the App, and therefore they contact the platform to express their gratitude. Of course, this result will highly depend on the user's device they are using to run the App. For this reason, we believe that Uber should also consider in their business model policies the constant update of the different mobile operating systems where the App can be run (such as IOS, Android, etc.) if yet is not on their agenda.

CHAPTER 5

Conclusions and future work

This chapter describes the conclusions extracted from this project, the results obtained and the thoughts about the future work.

5.1 Conclusions

In this project, we have described a methodology to characterize and analyze the messages posted by users to customer services based on the Twitter domain. Specifically, this project has focused on the analysis of the customer voice from @Uber_Support transport service platform during the complete year of 2020.

Firstly, we have made a global analysis of the data collected which helped us to understand how users publish information in the context of both transport and customer services. The first conclusion regarding this general description was that the restrictions and confinements that occurred due to the COVID-19 disease directly affected the daily volume of questions and concerns of users towards these services. Furthermore, we detected that users who posted opinions and sentiments on their tweets highly provided a negative expression. Moreover, we noticed that those tweets containing opinions usually describes similar vocabulary expressions which are related to the context of both transport and customer services (e.g **"cancelled trip"** which is highly frequent in negative tweets and **"quick response"** which is highly frequent in positive ones).

However, the initial description created does not characterize the underlying structure of the user's information. Therefore, to provide high-quality approaches that can help companies understand user's demands, we considered the importance of describing the latent topics discussed in the platform. Hence, we performed several pre-processing techniques on the collected data, and we combined them with a Topic Modelling System which concluded that customers and drivers usually posted in 2020 tweets to @Uber_Support concerning seven different topics.

The next process followed in the project was the design of a Topic Clustering System to partitionate each tweet into one of the predefined topics described above. After several genetic algorithm study cases, we concluded that the optimal procedure to create this system was through the creation of a hybridized genetic algorithm which combined the use of global and local convergence techniques to optimize the results by using the probability simplex as the main constraint of the problem. In addition to this algorithm, we compared its viability for Topic Clustering with K-Means approach implemented with scikit-learn [10]. Both algorithms described the same results of creating seven clusters, where each of them is distributed around an unique topic.

The combination of these two data mining systems previously mentioned allowed us to summarize the high volume of data in a detailed description of the sentiment information associated to each topic-cluster. The final conclusion of this sentiment information analysis can provide brands the ability to determine if their service towards those topics is positively rated or conversely if they need to change their business model to satisfy customer needs. In the case of the dataset analysed, we detected that all the topics described contained more negative tweets rather than positive ones. Furthermore, the tweets related with ***"Time and cancellation"***, ***"Opinions about Customer Service"*** and ***"Uber Eats"*** describes the most user's negative stance which could result in a lost of customers. To conclude, we consider this result a worrying outcome since Uber highly depends on user satisfaction to compete with other companies from the sector.

5.2 Future work

This section describes the future improvements and characteristics that are available to be performed in this project:

- ***Geographical analysis:*** Perform a geographical analysis to determine the different topics and sentiments in each city or country the transport service operates.
- ***Emotion analysis:*** A global emotion analysis from the dataset can describe more insights from customers tweets. Moreover, this analysis can be combined as we did for sentiments in order to obtain the emotions described in each of the topics defined.
- ***Analyse other social networks:*** Explore other social networks using the systems created in order to obtain different results for helping customers with their questions and concerns. Some examples could be the analysis of Facebook social media or other channels where this brand operates.
- ***Competence analysis:*** The analysis of other transport companies (such as Cabify or Lyft) can provide brands the ability to perform better services on those topics users are not satisfied with. Thus, this company will obtain more customers, which will be reduced in more revenues.

APPENDIX A

Impact of this project

This appendix indicates qualitatively the possible social, economic, environmental and ethical implications this project can have once created.

A.1 Social Impact

Social media information has grown in recent years. As a consequence, the public data available on these platforms has increased significantly. Therefore, collecting high pools of information from these platforms for research purposes has become one of the main ways to draw conclusions about public opinion.

In our particular case, public information from Social Media has allowed us to draw conclusions related to the transport and customer service sectors. We also believe that the use of our project can be a starting point in other research projects that involves further work on the subject.

Moreover, companies interested in obtaining more information about their customers and drivers can use our project to modify their business models to make their users more satisfied with the brand. Furthermore, as we mentioned during the project, companies who compete among others in the transport sector can also use this research to draw conclusions about their competitor's topics with bad public opinions to take advantage of it. To conclude, this competition between brands will always benefit the users who use these services.

A.2 Economic Impact

The economic impact of this project can be seen throughout two different perspectives:

Firstly, the use of this data mining system will allow companies to characterize the public opinions of their customers in an easy and faster way compared to the traditional methods based on questionnaires and paper forms. As a result, companies will save high amounts of money with the decrement in the economic effort this type of surveys involves.

Secondly, the possibility of acquiring this project will make companies able to analyze in detail public opinions, which can be used to modify their business models and marketing approaches periodically. This effect will increase the satisfied customers loyal to the company which is reduced into more revenues.

A.3 Environmental Impact

In relation to the environmental impact, we have to consider that this type of projects involves the use of Big Data and Machine Learning technologies which requires high computational resources and time processing for obtaining results. In our case, these technologies were carried out through our own computer and a GSI (Intelligent Systems Group) cluster.

The use of these computers for processing involves high electrical energy consumption demands for both processing and cooling.

Unfortunately, nowadays, the demand for electrical energy in Spain involves using non-renewable resources in its majority (primarily from exploiting carbon and nuclear energy). For this reason, the performance of this type of projects affects the increment of the carbon footprint. Also, we highlight that both manufacturing and recycling of the computers used also directly affects the increment of pollution to the environment.

A.4 Ethical Implications

This section describes the primary ethical consideration of the project:

The main implication of using this project is intimately related to the companies' exchange of human resources. As we mentioned in Section A.2, using the systems created in the project will significantly reduce the number of surveys performed. As a result, this effect may destroy some jobs related to this sector.

However, we believe that our project would also create new and different jobs since this tool requires human interpretability to obtain the results periodically.

APPENDIX B

Economic budget

B.1 Introduction

This appendix details the necessary budget needed for the creation of the project. Its structure will be divided into the different physical and human resources, licences as well as the taxes for a commercial use.

B.2 Physical resources

This project has been mainly performed through a personal computer which has the following characteristics:

- **CPU:** Intel(R) Core(TM) i5-4690 @ 3.50GHz (4 cores)
- **RAM:** 8GB, DDR3, CL10
- **Disks:** SDD: 250GB; HDD: 500GB

Nowadays, a computer with this similar specifications **can approximately cost 650€ in Spain (taxes included)**.

During the implementation of the Genetic Algorithms (Section 4.5). More computing processing resources were required for making the algorithm convergence in a shorter period of time. For this reason, a Big Data cluster from the Intelligent Systems Group was provided to the project. As a result, computing time decreased to the 50% in each iteration of the algorithms. The specifications of this cluster are shown below:

- **CPU:** Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
- **RAM:** 128GB RAM
- **Disks:** SSD of 1 TB and NAS distribution of 12 TB

A cluster with similar characteristics can cost in the market approximately 2500-3000€. However, we can consider that this computer has been amortized since it is a resource from the GSI group that is used in a high number of projects. Thus, we consider this cluster in the budget as 0€.

B.3 Human resources

The human resources to accomplish this project can be reduced into a single researcher. Considering, the case of this project which has been carried through a GSI Research Scholarship (UPM), **the monthly salary rises to 550€** (including the Spanish Social Security) for **80 hours of monthly work** (20 working hours each week). For this reason, the fee per hour is estimated to be **6,875€**. As the dedicated time needed to perform the project rises to **525h**, **the total human resources a project of this characteristics can cost approximately 3609,37€**.

B.4 Software and licences

All the licences from the different tools and frameworks used during the entire project were open-source. For this reason, **the total budget required for software and licences rises to 0€**.

B.5 Taxes

Considering a company interested in buying the product created, the taxes they should pay will depend on the VAT (Value Added Tax) for an advanced software project in the country that has been carried out. As a result, this tax must be added to the total price of the product's final cost.

APPENDIX C

Sample of tweets distribution

The following appendix gives the reader several examples of the tweets and their distribution through the different topics defined in the project. Moreover, we included in the table the polarity given to each tweet as well as the day it was posted on the platform.

Topic 1: Uber account/app	**Date**	**Polarity**
@Uber_Support need help with a verification code my email isn't getting	2020-03-31	Neutral
My uber account has been disabled without any reason @Uber_Support	2020-08-21	Negative
Topic 2: Uber drivers		
@Uber_Support the driver was not wearing any mask during this pandemic, that is risky and I would like to know why Uber did not ask al drivers to wear masks	2020-04-10	Negative
@Uber_Support I need to get a new drivers license and can't get an appointment for 2 months. My license expires in 1 month. Do you have a grace period due to COVID?	2020-08-12	Negative
Topic 3: Money and payments		
Whether you are refunding me or not, this is not a big amount for me. Q: Why did you take this extra charge? Beggars Team @UberINSupport @Uber @UberFacts @Uber_Support	2020-03-19	Negative
@Uber_Support I need a refund you took money from my account and the trip was a cash trip what kind of dubious act is that refund my money	2020-12-25	Negative
Topic 4: Uber Eats		
@Uber_Support my food never arrived	2020-03-31	Neutral
@Uber_Support @UberEats never brought my food but wants to charge me 4.99... never using that again.	2020-04-27	Negative
Topic 5: Opinions about Customer Service		
@Uber_Support everytime I tell you what the issue is, you guys give the same answers all the time and what your saying for me to do is wrong	2020-04-28	Neutral
@Uber_Support has been by far the worst customer service I've ever received. So frustrating that I will not be supporting their business any longer #ubersucks #uber #ubereats	2020-08-18	Negative
Topic 6: Time and Cancellation		
@Uber_Support been on hold for an hour and a half! What do I do at this point???	2020-03-23	Neutral
@Uber_Support So is it normal for a driver to cancel one minute before picking someone up on a SCHEDULED TRIP. Now my wife has to wait 30 minutes for another car because the driver cancelled when she was 1/2 a mile away. This is unacceptable.	2020-08-10	Negative
Topic 7: Direct contact with Uber Support		
@Uber_Support I sent a dm with my phone number and information regarding my concern	2020-04-30	Neutral
@Uber_Support I sent a follow up question. Thanks	2020-08-20	Positive

Table C.1: Sample tweets in each topic.

APPENDIX D

Detailed explanation of the first Genetic Algorithm study case

In this section, we describe in detail the first case of study implemented in the project for creating a genetic algorithm to describe the Topic Clustering problem. Firstly, we provide the reader with all the different operators implemented. Later, we describe in detail the results and conclusions of this GA.

D.1 Description of operators

Parameter	Methodology	Value
Individuals	Random Population	100
Selection	Tournament Selection	30
Crossover	Two Point Crossover	p1 = 0.8
Mutation	Individual mutation	p2 = 0.35
	Gene mutation	p3 = 0.1

Table D.1: Genetic operators in the first case of study.

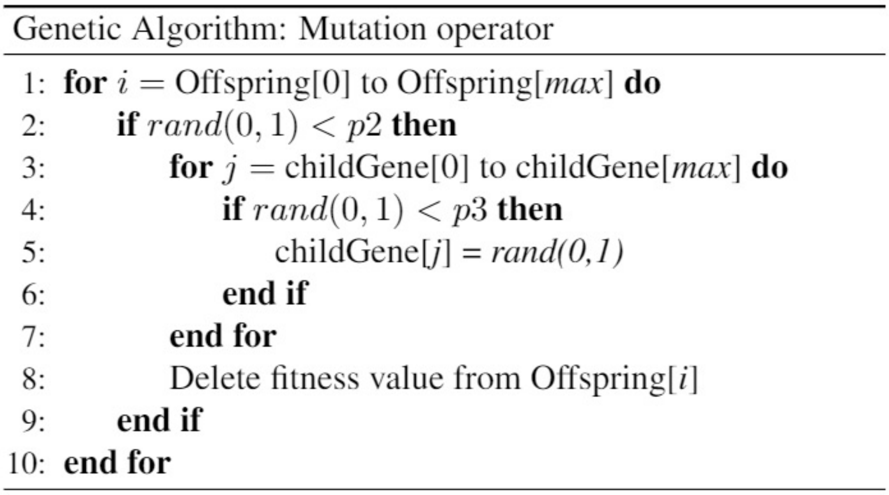
The **initial population** allowed us to explore the search space efficiently. More individuals in the pool result in an efficient exploration of the space. However, as we increase the number of individuals, computing time for each iteration also increases. As we are facing a Big Data problem (with a matrix of documents of 212400x7), computing time is one of the first prerequisites to optimize. For this reason, we selected 100 individuals (clustering solutions) to balance between exploration of the space and computing time. As this initial population was generated randomly, we ended with a partition of the space where some of these clusters were empty (clusters without documents assigned). This case was solved by assigning those invalid strings a high fitness value.

The selection process was preceded thought the tournament selection (described in Section 4.5.1) where we used N = 30 after performing several testing processes. This value resulted in a good acceleration of the convergence of the genetic process.

The crossover process was performed thought the two-point crossover with a certain probability (*P(crossover) <p1 = 0.8)*. This technique was selected because it allows some parent clusters to be copied in the child without modifying their dimensions (since we partitioned each individual in 3 portions) which will result in an efficient exploitation of the problem.

Finally, **the mutation** technique was performed in some offspring of the population with a certain probability rate (*P(mutate offspring)<p2 = 0.35*). Likewise, every gene in the offspring selected whose mutation probability is lower than a certain value (*P(mutate gene)<p3 = 0.1*) will mutate into another random value satisfying the constraint that all

the genes are random floating numbers between the range [0,1]. The selection of p2 and p3 values arises from the previous description to balance between exploitation and exploration described in Section 4.5.1.3. We also provide a pseudocode of this mutation process in the following image:



Genetic Algorithm: Mutation operator

```
for $i$ = Offspring[0] to Offspring[$max$] do
    if $rand(0, 1) < p2$ then
        for $j$ = childGene[0] to childGene[$max$] do
            if $rand(0, 1) < p3$ then
                childGene[$j$] = $rand(0,1)$
            end if
        end for
        Delete fitness value from Offspring[$i$]
    end if
end for
```

Figure D.1: Mutation pseudocode first GA

D.2 Results

We iterated the genetic algorithm a high number of generations for allowing the population to explore the space searching the global minimum of the problem described by the fitness function (Section 4.5.2.2). The following figure shows the logarithmic convergence of the algorithm through the different generations:

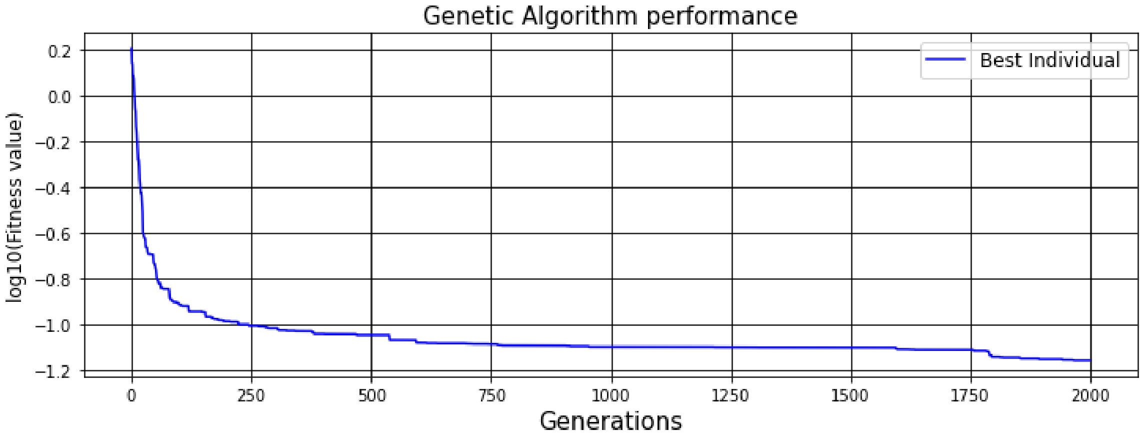


Figure D.2: Genetic Algorithm convergence.

From Figure D.2 it is seen that the convergence process is faster during the first iterations in comparison with the last generations. This convergence tendency occurs because the first individuals from the population can easily find solutions in the space which are better than their parents. However, during the last generations, the convergence process starts to describe an asymptotic result because the new offspring generated are not able to obtain better solutions than their parents.

Once captured the best individual (the one with the minimum fitness function from GA), we assessed both Genetic and K-means algorithms in order to see which of them minimizes more the distance between documents and their closest cluster.

Algorithms	K-Means	GA
Fitness	0.08262	**0.06918**

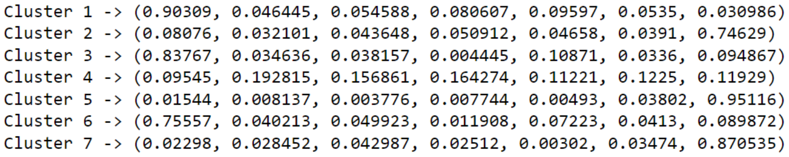
Table D.2: Evaluation of K-means and Genetic Algorithm from the first study case

We can see in Table D.2 that the genetic algorithm mathematically describes lower fitness values for clustering the different tweets in a distribution of K=7 topics.

However, we need to analyze deeply if this optimal individual from the genetic algorithm can be a valid result for performing a Topic Clustering System. To do this, it is necessary to see how the cluster coordinates are distributed throughout the space and how the documents are assigned to this 7 clusters solution.

Cluster	Number of Tweets
1	2842
2	35420
3	1197
4	142100
5	750
6	24168
7	5923

Table D.3: Tweets in each cluster



```
Cluster 1 -> (0.90309, 0.046445, 0.054588, 0.080607, 0.09597, 0.0535, 0.030986)
Cluster 2 -> (0.08076, 0.032101, 0.043648, 0.050912, 0.04658, 0.0391, 0.74629)
Cluster 3 -> (0.83767, 0.034636, 0.038157, 0.004445, 0.10871, 0.0336, 0.094867)
Cluster 4 -> (0.09545, 0.192815, 0.156861, 0.164274, 0.11221, 0.1225, 0.11929)
Cluster 5 -> (0.01544, 0.008137, 0.003776, 0.007744, 0.00493, 0.03802, 0.95116)
Cluster 6 -> (0.75557, 0.040213, 0.049923, 0.011908, 0.07223, 0.0413, 0.089872)
Cluster 7 -> (0.02298, 0.028452, 0.042987, 0.02512, 0.00302, 0.03474, 0.870535)
```

Figure D.3: Cluster-centroids solution in the first study case

From the previous figures, we can analyze two relevant results that need to be interpreted. Firstly, in Table D.3 we can see that the 66,9% of the documents from the dataset are assigned to the fourth cluster. This cluster (Figure D.3) has its coordinates distributed homogeneously through the 7 dimensions (printing values from 0.095 to 0.192). For this reason, the documents whose mixture of topic probabilities are uniformly distributed are assigned to this cluster. Secondly, we can see in Figure D.3 that some clusters are overlapped since their predominant coordinates are the same in each of the clusters (this effect occurs in the case of the first, third and sixth clusters from the individual). As a result, these clusters (which belong to the same topic) divides their documents into them.

Both results conclude that this first study case does not represent the reality of the problem we are facing, which involves assigning a cluster to a unique topic.

D.3 Conclusions

The results provided by the first study case mathematically describes a solution that optimizes the distance between documents and their closest cluster in comparison to the K-Means algorithm. However, as we saw in the previous section, the description of the clusters throughout the space do not represent the feasibility of a Topic Clustering System.

APPENDIX E

Genetic Algorithm Iterations

This appendix includes the different iterations performed in all the study cases from the Genetic Algorithms (Section 4.5) in order to obtain the optimal values of convergence. As a result, we selected the lowest fitness value from the different iterations to continue with these individuals the rest of the studies. The following tables show the different operators modified and their best individual fitness values after a predefined number of generations.

Individuals	Generations	Tournament Selection	Crossover operator	Mutation operator	Fitness value
150	1200	N = 30	p1 = 0.8	p2= 0.2 p3 = 0.05	0.088179
100	2500	N = 10	p1 = 0.8	p2=0.35 p3 = 0.1	0.092738
100	600	N = 20	p1 = 0.8	p2 = 0.35 p3 = 0.1	0.082611
100	**2000**	**N = 15**	**p1 = 0.8**	**p2 = 0.35** **p3 = 0.1**	**0.06918**

Table E.1: Iterations of the genetic algorithm from the first study case.

Individuals	Generations	Tournament Selection	Crossover operator	Mutation operator	Fitness value
100	300	N = 5	p1 = 0.8	p2= 0.25 m_prob = 0.3	0.091397
100	200	N = 5	p1 = 0.8	p2=0.2 m_prob = 0.2	0.08940
100	500	N = 5	p1 = 0.8	p2 = 0.4 m_prob = 0.35	0.09423
100	300	N = 15	p1 = 0.8	p2 = 0.25 m_prob = 0.3	0.091402
100	**200**	**N = 5**	**p1 = 0.8**	**p2 = 0.35** **m_prob = 0.2**	**0.08578**
100	1300	N = 10	p1 = 0.8	p2 = 0.35 m_prob = 0.35	0.089038

Table E.2: Iterations of the genetic algorithm from the second study case.

Bibliography

[1] Arya, Shreyash. The influence of social networks on human society. `https://www.researchgate.net/publication/343949123_The_Influence_of_Social_Networks_on_Human_Society`, 2020. Accessed: 2021-05-25.

[2] Kalpathy Subramanian. Influence of social media in interpersonal communication. *International Journal of Scientific Progress and Research (IJSPR)*, 109:70–75, 08 2017.

[3] Wu He, Xin Tian, Yong Chen, and Dazhi Chong. Actionable social media competitive analytics for understanding customer experiences. *Journal of Computer Information Systems*, 56(2):145–155, 2016.

[4] Scott Wallsten. The competitive effects of the sharing economy: How is uber changing taxis? *Technology Policy Institute*, pages 1–21, 07 2015.

[5] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.

[6] Steven Bird. Nltk: The natural language toolkit. 01 2006.

[7] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[8] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[9] J. Fernando Sánchez-Rada, Carlos A. Iglesias, Ignacio Corcuera-Platas, and Oscar Araque. Senpy: A Pragmatic Linked Sentiment Analysis Framework. In *Proceedings DSAA 2016 Special Track on Emotion and Sentiment in Intelligent Systems and Big Social Data Analysis (SentISData)*, pages 735–742, Montreal, Canada, October 2016. IEEE.

[10] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012.

[11] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan

Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[12] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[13] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[14] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[15] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[16] MATLAB. *9.7.0.1190202 (R2019b)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[17] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[18] Benjamin Bengfort, Rebecca Bilbro, Nathan Danielsen, Larry Gray, Kristen McIntyre, Prema Roman, Zijie Poh, et al. Yellowbrick, 2018.

[19] Jason S. Kessler. Scattertext: a browser-based tool for visualizing how corpora differ. *Proceedings of ACL 2017, System Demonstrations*, pages 85–90, 2017.

[20] Anton Borg and Martin Boldt. Using vader sentiment and svm for predicting customer response sentiment. *Expert Systems with Applications*, 162:113746, 2020.

[21] Gonzalo A. Ruz, Pablo A. Henríquez, and Aldo Mascareño. Sentiment analysis of twitter data during critical events through bayesian networks classifiers. *Future Generation Computer Systems*, 106:92–104, 2020.

[22] David M. Blei, Andrew Y. Ng and Michael I. Jordan . Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[23] David Blei and Jon McAuliffe. Supervised topic models. *Advances in Neural Information Processing Systems*, 3, 03 2010.

[24] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[25] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.

[26] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[27] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. volume 1, pages 281–297, 01 1967.

[28] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[29] Adrian Drake and Robert Marks. Genetic algorithms in economics and finance: Forecasting stock market prices and foreign exchange — a review. *In S. H. Chen (Ed.), Genetic Algorithms and Genetic Programming in Computational Finance*, pages 29–54, 2002, New York: Springer.

[30] Adamopoulos, Adam and Perdikuri, Katerina. Using a genetic algorithm for detecting repetitions in biological sequences. `https://www.researchgate.net/publication/242089942_Using_a_Genetic_Algorithm_for_Detecting_Repetitions_in_Biological_Sequences/references`, 2004. Accessed: 2021-05-25.

[31] Abolfazl Pourrajabian, Maziar Dehghan, and Saeed Rahgozar. Genetic algorithms for the design and optimization of horizontal axis wind turbine (hawt) blades: A continuous approach or a binary one? *Sustainable Energy Technologies and Assessments*, 44:101022, 2021.

[32] Kalyanmoy Deb. *Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction*, pages 3–34. Springer London, London, 2011.

[33] S. Legg, M. Hutter, and A. Kumar. Tournament versus fitness uniform selection. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 2, pages 2144–2151 Vol.2, 2004.

[34] Demitrios E. Pournarakis, Dionisios N. Sotiropoulos, and George M. Giaglis. A computational model for mining consumer perceptions in social media. *Decision Support Systems*, 93:98–110, 2017.