FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Knowledge Extraction from Social Networks for Near Real-Time Transport Network Evaluation: An Ensemble Approach

Eduardo Leandro Dias Carneiro

Mestrado em Engenharia Informática e Computação

Supervisor: Tânia Daniela Lopes da Rocha Fontes

Co-Supervisor: Rosaldo José Fernandes Rossetti

July 28, 2023

Knowledge Extraction from Social Networks for Near Real-Time Transport Network Evaluation: An Ensemble Approach

Eduardo Leandro Dias Carneiro

Mestrado em Engenharia Informática e Computação

July 28, 2023

Abstract

A pervasive use of social media by millions of individuals as a platform for daily discussions on diverse topics has established it as an invaluable source of real-time information. Public transport operators and city planners have acknowledged the significance of social media in the context of transport networks, prompting them to extract and store relevant data from these platforms. However, the utilization of such data for practical applications remains limited due to its inherently unstructured nature, encompassing various languages, irony, symbols, and expressions.

The abundance of information available on social media presents an opportunity to devise innovative methods for faster and more efficient identification and evaluation of traffic events, surpassing the limitations of traditional methods. This work aimed to develop a pipeline or tool capable of automating this evaluation process by leveraging direct content extracted from social media, along with other data sources.

The proposed pipeline encompasses three tasks considered the most relevant for an automatic evaluation: (i) transports-related text classification, (ii) sentiment analysis, and (iii) topic modeling, together with labeling. These tasks collectively facilitate the identification of relevant content, analysis of user sentiment, and identification of topics discussed, particularly those related to traffic disruptions and their implications.

Ensemble techniques are employed in the first two tasks to enhance the classification process. For text classification, a combination of standard machine learning algorithms (SVM, LR, and RF) and various Google BERT models is utilized. Meanwhile, the ensemble used in sentiment analysis integrates VADER, TextBlob, Afinn, and BERT models. The final task, topic modeling and labeling, uses LDA in conjunction with LLMs to identify topics and generate their corresponding labels using namely ChatGPT and Bard.

The pipeline was tested using appropriate metrics for each task, commonly used in the literature, and having as a base vocabulary and texts typically used in real-life cases. The tests demonstrated promising results, with text classification achieving close to 96% accuracy. The sentiment analysis ensemble produced satisfactory results, greater than 60%, and allowed to combat the dispersion of the algorithms used in the literature. Furthermore, the topic modeling implementation demonstrated to be capable of generating suitable labels for the topics tested.

The results proved that this work is viable and has the potential to continue to be improved. They also suggest that this work could be very useful to help improve decision-making related to transport management in urban areas, thus benefiting a diverse set of users.

Keywords: Transport Networks, Social Media, Knowledge Extraction, Ensemble, Word Embedding, Text Classification, Sentiment Analysis, Topic Modeling

Resumo

O uso generalizado das redes sociais por milhões de pessoas, como uma plataforma para discussões diárias sobre diversos tópicos, fez destas redes uma fonte importante de informações gerada em tempo real. Operadores de transporte público e urbanistas reconhecem atualmente a importância das redes sociais no contexto das redes de transporte, extraindo e armazenando dessas mesmas plataformas dados relevantes. No entanto, a utilização destes dados é ainda limitada devido à sua natureza inerentemente não estruturada, abrangendo geralmente vários tipos de linguagens e recorrendo muitas vezes a ironia, símbolos e expressões.

A abundância de informações disponíveis nas redes sociais apresenta uma oportunidade para desenvolver métodos inovadores para identificação e avaliação mais rápida e eficiente de eventos de tráfego, superando as limitações dos métodos tradicionais. Este trabalho teve como objetivo desenvolver um *pipeline* ou ferramenta capaz de automatizar esse processo de avaliação, aproveitando o conteúdo extraído diretamente das redes sociais, juntamente com outras fontes de dados.

O *pipeline* proposto, desenhado com base numa revisão de literatura, engloba três tarefas consideradas as mais relevantes para uma avaliação automática: (i) classificação de texto relacionado com transportes, (ii) análise de sentimentos e (iii) modelação e rotulagem de tópicos. Estas tarefas facilitam coletivamente a identificação de conteúdo relevante, análise de sentimentos dos utilizadores e identificação dos tópicos discutidos, particularmente os relacionados com disrupções de tráfego e as suas implicações.

Para melhorar o processo de classificação, são usadas técnicas de *ensemble* nas duas primeiras tarefas. Para classificação de texto, é utilizada uma combinação de algoritmos padrão de aprendizagem automática (SVM, LR e RF) e vários modelos Google BERT. Para a análise de sentimento, o *ensemble* usado integra os modelos VADER, TextBlob, Afinn e BERT. A tarefa final, modelação de tópicos, utiliza LDA em conjunto com LLMs para identificar tópicos e gerar rótulos correspondentes, usando nomeadamente o ChatGPT e o Bard.

O *pipeline* foi testado usando métricas adequadas para cada tarefa, habitualmente usadas na literatura, e tendo como base vocabulários e textos normalmente utilizados em casos reais. Os testes demonstraram resultados promissores, com a classificação de texto a alcançar resultados próximos de 96% de precisão. O *ensemble* de análise de sentimento produziu resultados satisfatórios, superiores a 60%, e permitiu combater a dispersão de algoritmos utilizados na literatura. A modelação de tópicos demonstrou ser capaz de gerar rótulos adequados para os tópicos testados.

Os resultados provaram que este trabalho é viável e tem potencial para continuar a ser melhorado. Também sugerem que este trabalho poderá ser bastante útil para ajudar a melhorar as tomadas de decisão relacionadas com a gestão de transportes em espaço urbano, beneficiando assim um conjunto diversificado de utilizadores.

Palavras-Chave: Redes de Transporte, Redes Sociais, Extração de Conhecimento, Representação Vetorial de Palavras, Classificação de Texto, Análise de Sentimentos, Modelação de Tópicos

Acknowledgments

I want to start by thanking my supervisor, Professor Tânia Fontes, and my co-supervisor Professor Rosaldo Rossetti. This was a complex project and my first experience within the research community. As with almost any project, there were ups and downs, but the guidance, constant feedback, and patience I got from both helped me keep working on the project and face the difficulties no matter how hard they were. It also allowed me to improve not only as a student but as a person. Working with both gave me a new perspective on what investigation can be and how it should be conducted. I would also like to give an extended thank you to INESC TEC, particularly CESE, for providing the resources that made it possible to develop this work.

To my family, I want to express the great appreciation I have for them. No matter how often I complained about something, they always had words of reinforcement and never stopped encouraging me to produce the best possible results. Even though sometimes I doubted myself, I never doubted for a second they believed in me and in what I was capable of. None of what I achieved would be possible without them and their support.

Lastly, I thank my friends and colleagues I made throughout the years, both in school and in college. Everyone I met inspired me with their unique ideas, projects, and ambitions. I believe being surrounded by all these amazing people was part of why I am who I am today.

Eduardo Leandro Dias Carneiro

This work was financed by National Funds through the FCT—Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within the project e-LOG (ref. EXPL/ECI-TRA/0679/2021).

"We dont have a choice on whether we do social media,
the question is how well we do it"

Erik Qualman

Contents

List of Figures

List of Tables

Abbreviations and Symbols

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag-of-Words
DT	Decision Tree
Glove	Global Vectors for Word Representation
GS-DMM	Gibbs Sampling Dirichlet Multinomial Mixture
HTML	Hypertext Markup Language
KNN	K-Nearest Neighbors
LDA	Latent Dirichlet Allocation
LLM	Large Language Model
LR	Logistic Regression
MCC	Manual Classified Counting
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
RF	Random Forest
ROSTCM 6	ROST Content Mining System Version 6.0
SA	Sentiment Analysis
SM	Social Media
SVM	Support Vector Machine
TC	Text Classification
TM	Topic Modeling
VADER	Valence Aware Dictionary and sEntiment Reasoners
VEM	Variational Expectation Maximization

Chapter 1

Introduction

This chapter is divided into four sections. Section 1.1 presents the context for the project theme, including statistics about the current congestion problems and the growth of social networks. Section 1.2 explains the motivations for exploring this problem using Social Media. The main objectives for the work developed during this project are described in Section 1.3. Finally, Section 1.4 gives an overview of how the rest of this document is structured.

1.1 Context

In the coming fifteen to thirty years, an estimated two-thirds of the population will be concentrated in urban and suburban regions, intensifying the already substantial pressure on urban transportation networks [15]. As urbanization continues to gather momentum, the demand for efficient transportation solutions will be more crucial than ever before. Forum [37] highlights that the global passenger demand will more than double between 2015 and 2050. Consequently, transport networks face substantial impacts, making monitoring and managing the evolving landscape increasingly challenging.

In 2017, congestion cost Britain, Germany, and the United States citizens almost 461 billion dollars, and taking as an example the city of Boston, commuters spent almost 14% of their travel time stuck in traffic [35]. An audit made in 2019 concluded that inefficiencies in urban mobility and road congestion, in particular, cost the Europe Union almost 110 billion euros per year [85]. Depending on the zone congested, this problem can have many adverse effects [28] like increased commuter costs, safety problems due to more crashes, environmental and public health problems related to all the emissions due to the wasted fuel, and reduced economic competitiveness.

In order to properly manage the transportation networks, it is important to keep track of all these daily travelers and their commutes. Currently, multiple methods are available to perform traffic counts like Manual Classified Counting (MCC), Video Image Detection, or Pneumatic Tube [91]. All these have pros and cons/restrictions related to how they need to be set up, how they function, and how much they cost to maintain. Since the sensors used in some of these methods can only provide counts about the number of vehicles that cross a specific part of the network, the

data they generate lacks context and does not allow atypical event detection, which is crucial to evaluate transport networks. Most of these methods are usually only used on highways or urban areas, making information in suburban areas almost nonexistent.

With this increased growth and the cons presented for the current methods, there is a necessity to look for new data sources that can help perform this type of analysis regarding traffic. One of these sources might be online social media (referred to during the rest of this work just as social media). It is a significant part of society, and according to Simon Kemp [56], statistics show that at the beginning of 2022, there were 4.62 billion active users, representing 58.4% of the world population.

Part of these social media users discuss and talk about their daily lives on these social networks, which produces an enormous amount of untreated data. These conversations and interactions cover a vast amount of topics [66], like sports, politics, health, and pop culture. Transports are also a topic discussed, making social media very important for entities like transportation providers, as demonstrated for the city of New York [55].

1.2 Motivation

The information from social media comes in many different forms. Users can write in different languages, use text symbols, emojis, or images, and use context-specific language or forms of expression like irony. Calisir and Brambilla [19] research demonstrates that storing the data is a relatively simple task, but processing it, considering all these different characteristics requires complex methodologies.

Some transport network operators and city planners already store information from social media [72] but have yet to properly use it to automatically evaluate road network status and user satisfaction. Using this data adequately could be the next step in traffic management.

With the correct techniques, the knowledge extracted from this stored data can benefit society, providing near real-time information for a variety of subjects, like politics or sports, as explained by Immonen et al. [50].

Davis and Saunders [32] explain that there are already experiences in which social media data is used to understand how users feel about their transportation services regarding subjects like cleanliness or delays. These experiences show potential for a more generalized use, like evaluating entire transport networks regarding complex subjects like accidents or congestion.

All this leads to the belief that a methodology or tool can be defined so that social media data can be used to evaluate transport networks in near real-time automatically. Something like this could provide different users with a new way to perform these assessments without needing other methods involving sensors or manual evaluation.

1.3 Objectives

The main goal of the present dissertation is to study how the use of social networks to retrieve information about transport networks in near real-time can be improved. This goal is divided into three objectives:

- Compare different algorithms/models for automatically evaluating a transport network in near real-time, for a real-life case, using the information provided in social media.
- Create a pipeline capable of making this evaluation, starting from the identification phase of transportation-related content from social media until the presentation of the results.
- Improving the solutions that are already available and contributing to this topic with new ideas by providing detailed information about both the studied and the implemented approaches.

Starting with the methodologies comparison and definition of the most appropriate one, as the state-of-the-art section will show, multiple approaches are already being considered for projects like this.

To fulfill the objectives described above, the idea is to study and test multiple hypotheses and understand the ones that best fit the domain in question. Three different tasks, text classification, sentiment analysis, and topic modeling, are analyzed so it is possible to understand how they can contribute to traffic/transport network evaluation. The two main possibilities to tackle these tasks are finding a really good individual model that provides what is intended or combining different solutions using ensemble techniques. Identifying good real-life cases is also part of ensuring that what is developed can be evaluated.

As for the pipeline implementation, while developing, making sure each part of this process is well connected is another important step to guarantee this work can be used outside of a non-academic environment.

After creating the pipeline, the next step is properly evaluating it using adequate performance metrics. Making this guarantees that decision-makers or other stakeholders interested in the work can recreate it and compare the results with the ones from their implementations or ideas. After using the performance metrics, it is also important to identify real-life cases and test what was implemented using them, which will demonstrate if the work can be applied in everyday life.

This project follows up the dissertations developed by Pereira [92] and Murçós [79]. Pereira tested traditional machine learning text classifiers with a different embedding approach to implement a better transportation-related classifier while also exploring topic modeling applied to generic social media content. Murçós developed a transportation-related classifier using a Google BERT model without additional training and explored the use of a sentiment analysis algorithm to look for possible traffic issues.

1.4 Structure

This work is structured into seven different chapters.

The first chapter introduces the work, providing context and delineating the motivations, objectives, and possible contributions.

Chapter 2 is the background for this project and presents an overview of the topics discussed in this document, explaining the concepts necessary to understand it.

In Chapter 3, the background is followed by a systematic review of the work developed by other authors concerning this subject, giving insights into the different approaches that can be found.

Chapter 4 details the methodological approach used to answer the identified problem. It starts with the formalization of the problem, making the connection between the systematic review, the objectives, and the implementation expectations, followed by the data used, which is described in detail. There is a brief explanation of the methods considered to deal with each section of the work, but on a high level, since the implementation details are part of the next chapter. The last subject discussed in this chapter is the performance metrics chosen to evaluate the developed work.

The implementation is presented in Chapter 5. It is organized according to the three main tasks: text classification, sentiment analysis, and topic modeling. The algorithms used are explained in detail for each part so that the working process can be understood and easily replicated.

Thorough Chapter 6 the performance metrics results from the tests run for all the components are displayed, followed by a demonstration of the developed work applied in real-life situations. The chapter ends with a discussion and evaluation of the results.

Lastly, Chapter 7 concludes the work developed during the dissertation, reflecting on the results obtained and giving suggestions for future work so that this project can keep growing and improving.

Chapter 2

Background

This chapter introduces and briefly explains the most important concepts to understand the developed work. It is organized into four sections, starting with Section 2.1, where an introduction to social media networks and the concept of microblogging is provided. Next, in Section 2.2, the Natural Language Processing (NLP) concept is explained, with an in-depth dive into the four main areas relevant to this work: embeddings, text classification, sentiment analysis, and topic modeling. The last section related to a concept, Section 2.3, discusses what ensembles are, their variations, how they can be implemented, and their advantages.

2.1 Social Media

According to Oxford Advanced Learner's Dictionary, microblogging can be defined as "the activity of sending regular short messages, photos or videos over the internet, either to a selected group of people or so that they can be viewed by anyone, as a means of keeping people informed about your activities and thoughts" [88].

Multiple platforms can be used for microblogging, which can have different shapes depending on the information a user wants to share or read. Some examples of these social networks and their microblogging form are:

- LinkedIn [1] [64, 52] - it was officially launched in 2003, and unlike other social media platforms, LinkedIn is focused on networking. Users can message persons, post updates, share, and like content about professional careers, jobs, or projects.
- Facebook [2] [73] - available since 2004, it is a social network where it is possible to make text posts and share images, videos, and links. Users can follow or add other people to a friends list so they can chat and see what they post. Other users can then comment, like, or share these posts.

[1] www.linkedin.com
[2] www.facebook.com

- Twitter [3] [109] - released in 2006, allows users to share their opinions in real-time using images, GIFs, videos, audio, and a maximum of 280 characters. Users without an account can read but can not publish, share, or interact with anything.
- Sina Weibo [4] [27, 42] - released in 2009, is essentially a Chinese version of Twitter release as a response to the Twitter ban imposed by the Chinese government. According to its creators, users can create and post a feed and attach multi-media and long-form content. Unlike Twitter, since 2016, there has been no character limit.
- Pinterest [5] [95] - launched in 2010, it is a visual discovery engine for finding ideas like outfits, decorations, and manual arts. Allows users to upload images and videos and associate them with a link, a title, and a description. Other users can then add the images to their boards, which are collections of posts.
- Instagram [6] [74, 75] - also released in 2010, its users can upload photos and videos, sharing them with everyone (public profile), their followers (private profile), or a selective group of friends. Like on Facebook, they can also view, comment, like, and share posts.

The short format used in these platforms encourages users to give regular updates on their daily life, which produces a vast amount of data. This information is primarily public and available in near real-time, making it a great source to substitute or complement surveys or sensor data that currently power most data analysis tools.

More social media networks could be detailed here. Still, they were disregarded either because they were a variation of one of the networks presented or because the content outputted by most users did not fit the microblogging definition. Therefore they are out of scope.

2.2 Natural Language Processing

Natural Language Processing, commonly referred to as NLP, is one of Artificial Intelligence's (AI) key components and can be defined as a set of computational techniques that give a computing device the capability of analyzing and representing a human language, either written or spoken, with the intent of using it for a variety of tasks that require "human-like" comprehension [63].

NLP can have multiple applications, depending on the content analyzed and the expected result. The most common core applications [58] are text classification, information extraction,

[3] www.twitter.com
[4] www.weibo.com
[5] www.pinterest.com
[6] www.instagram.com

conversational agent, information retrieval, and question answering. These core applications include common tasks most people are already familiar with, like spam detection, text translation, assistant AI like Alexa [7] or Siri [8], and chatbots like ChatGPT [9] or Google Bard [10].

Throughout history, NLP has been changing and evolving to what it has become today. It started around the 1950s with the Georgetown-IBM [47], an experience in which a computer could translate Russian sentences to English using a machine translation system. Later, Noam Chomsky's work [25] was the first step to establishing computational linguistics, and his ideas inspired the creation of rule-based systems for parsing and generating natural language. During the 70s and the 80s, these rule-based systems were the standard for NLP, and they relied on handcrafted linguistic rules and knowledge bases and had very few capabilities [53]. From the late 80s until the 00s, scientists started exploring machine learning techniques like Hidden Markov Models [97] and statistical language models. This era saw the rise of corpus linguistics, where large-scale textual data became crucial for training and evaluating NLP models. Although during the 00s neural networks and deep learning were already being studied, with Yoshua Bengio and his team proposing the first concept of a neural network in 2001 [11], it was only during the 2010s that they became popular and the norm for most tasks since they proved to be highly effective in various NLP tasks, including language modeling, sentiment analysis, and named entity recognition. In the last year and a half, Large Language Models, commonly known as LLMs, algorithms that use deep learning techniques and large data sets to understand, summarize, generate, and predict new content, are becoming more relevant and popular [116], mainly due to the rise of ChatGPT supported by an LLM called GPT-4 [86].

Today, NLP continues to evolve rapidly, driven by advancements in deep learning, the availability of large-scale annotated datasets, and the integration of multi-modal information (e.g., text, images, and audio). Researchers are exploring innovative techniques like transfer learning, reinforcement learning, and self-supervised learning to enhance the capabilities of NLP models and tackle real-world challenges [58].

Contrary to other sources of data, human language does not follow a defined structure, making it very challenging to work with. These difficulties are mainly related to the lack of precision in the text, the tone and inflection, and the constant evolution of languages [58].

The concept of embeddings is explored in the following subsection, and then three NLP applications relevant to this work are explained in detail.

2.2.1 Embeddings

Text Embeddings are, at their core, a numerical way to represent words and their context in a given sentence, as demonstrated in Figure 2.1. They are a fundamental concept in NLP and machine learning [54]. In embeddings, words are dense numerical vectors in a multi-dimensional space,

[7] https://alexa.amazon.com
[8] www.apple.com/siri
[9] https://chat.openai.com
[10] https://bard.google.com

capturing semantic and syntactic relationships between words. This technique has revolutionized how computers process and understand language [54].

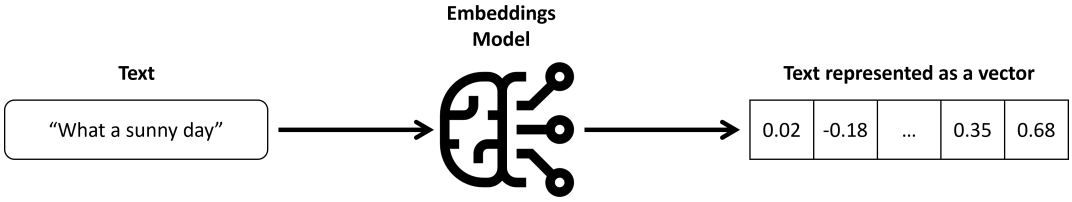

Figure 2.1: Sentence represented as a vector.

Traditional methods of representing words, such as one-hot vectors, where each word gets a binary value indicating its presence or absence in a pre-defined vocabulary, suffer from a high-dimensional and sparse representation, besides the fact that words can have different meanings depending on the context in which they are used, something that can not be accounted using this type of approach. Word embeddings, on the other hand, provide a dense and continuous representation, encoding valuable information about their meaning and usage [54].

The process of generating word embeddings can be very resource-consuming since it involves training models on a large dataset of text. These models learn to predict a word based on its context or predict the context given a word [80]. They capture the underlying semantic and syntactic relationships between words by considering statistical patterns in the data. The resulting embeddings form a distributed representation, where similar words are closer together in the embedding space [80]. Figure 2.2 demonstrates how words with similar relations are grouped on the vectorial space, while words without a relation are separated.

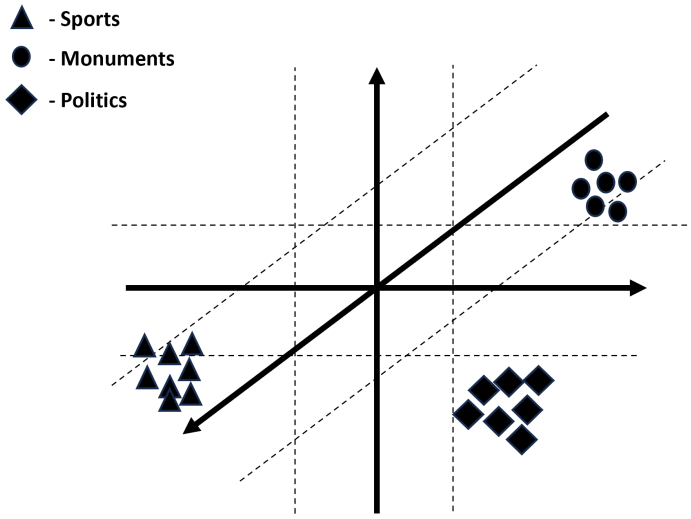

Figure 2.2: Differences between embeddings represented on the vectorial space, according to the topic they are related to.

There is a diversity of methods capable of generating word embeddings, and two of the most popular ones are Word2Vec [76] and Global Vectors for Word Representation (GloVe) [90].

Word2Vec uses a shallow neural network to learn word representations. It employs either a skip-gram or a continuous bag-of-words (CBOW) architecture. The skip-gram model predicts the context words given a target word, while CBOW predicts the target word from the context [76]. Both approaches effectively generate embeddings by adjusting the model's parameters during training. GloVe combines the advantages of global matrix factorization methods and local context window methods to create embeddings [90]. It leverages co-occurrence statistics derived from large text corpora to capture word relationships.

Word embeddings have many applications in the NLP area. They are used in various tasks such as sentiment analysis, text classification, machine translation, question-answering systems, and information retrieval. This type of representation, using vector space, enables algorithms to understand and interpret textual data more effectively. Furthermore, they also facilitate the understanding of semantic relationships between words. It is possible to perform vector operations in the embedding space, so for example, subtracting the vector representation of "lion" from "lioness" and adding the vector representation of "actress" yields a vector close to "actor", showcasing the ability of word embeddings to capture gender relationships [70], as displayed on Figure 2.3. The same thought process can be applied to things like translation or verb conjugation.

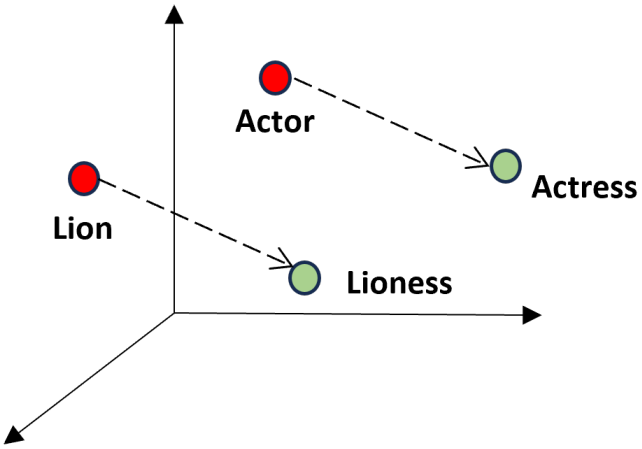

Figure 2.3: Vector representations for similar relations.

Word embeddings transformed the field of NLP by providing efficient and meaningful representations of words. They capture semantic and syntactic relationships, enabling algorithms to comprehend and analyze text more accurately. With their wide range of applications, word embeddings continue to play a crucial role in advancing language understanding and driving innovations in artificial intelligence.

2.2.2 Text Classification

Text classification is one of the tasks included in the NLP domain and can have many utilities found in everyday life, like spam detection or news categorization. It consists in assigning a set of pre-defined tags or labels to a given text, and it is usually a supervised task, which means for

it to be implemented, algorithms need a data set of manually labeled text on which they can be trained [69].

Transforming the chosen documents/texts into numerical representations that machine learning algorithms can process is part of a text classification task. The most common methods for document representation include bag-of-words, term frequency-inverse document frequency (TF-IDF), or the ones explained in the previous subsection, word embeddings like Word2Vec and GloVe. These representations are crucial to obtain better results [54].

Once documents are numerically represented, feature extraction techniques are employed to identify relevant patterns and characteristics. Extracted features serve as input to the classifiers [103]. The techniques used include:

- N-grams capture the frequency and co-occurrence patterns of words or characters in a text [103]. By analyzing n-gram frequencies, models can learn patterns and probabilities to generate more coherent and contextually appropriate text.
- Part-of-speech tagging consists of analyzing and categorizing with a tag each word in a given sentence, which allows to understand their syntactic function and context [103].
- Syntactic parsing is the process of analyzing the grammatical structure of a sentence to determine the relationships and hierarchical arrangement of words and phrases [103]. It involves assigning a parse tree or dependency structure that represents the syntactic dependencies among the words.

Depending on the available time, resources, and other restrictive factors, there is a wide variety of machine learning algorithms that can be used to perform the classification [39], such as Naive Bayes (NB), support vector machines (SVM), decision trees (DT), random forests (RF), and deep learning models like convolutional neural networks and recurrent neural networks. These models learn from the features extracted from the documents to make predictions.

To build a supervised text classifier that outputs satisfying results, a well-labeled and vast dataset is required for training. This dataset consists of documents with known categories or labels. The classifier is trained on this data using supervised learning techniques. This labeled data is also important to evaluate the classifier's performance, which is usually assessed using metrics like accuracy, precision, recall, and F1-score [46].

Nowadays, text classification still faces several challenges. One challenge is handling high-dimensional data, which arises due to a corpus's large number of unique words or features [33]. The "curse of dimensionality" can impact the classifier's performance. Dimensionality reduction techniques, such as feature selection or dimensionality reduction algorithms like principal component analysis, can help mitigate this challenge. Another challenge is dealing with imbalanced data. These tasks often involve imbalanced datasets where some categories have significantly fewer examples than others. The imbalance can bias the classifier towards the majority class and result in poor performance for minority classes. Techniques like oversampling, undersampling, or using class weights are the most common methods to deal with this issue [89].

Pre-trained language models, such as BERT and GPT, are now gaining more popularity and have already proven effective in capturing contextual information and improving classification accuracy when fine-tuned for specific problems.

Advancements in NLP research and the availability of large-scale labeled datasets continue to drive the progress of text classification, making it an exciting and evolving field of study.

2.2.3 Sentiment Analysis

Sentiment analysis, also called opinion mining, is a field of natural language processing that focuses on determining and extracting subjective information from textual data. Considered a text classification task, it involves analyzing and understanding the sentiment or emotion expressed in a piece of text, such as a review, a tweet, or a private message [54]. This process is crucial in many daily life applications, i.e., social media monitoring, brand reputation management, market research, and customer feedback analysis. By automatically interpreting the users' sentiments, businesses and organizations can gain valuable insights into public opinion and make data-driven decisions.

The main objective of sentiment analysis is to classify the sentiment of a given text. This classification value, called polarity, can have a qualitative value, such as positive, negative, or neutral, or a quantitative value, for example, between -1 and 1. The scales used to quantify this sentiment vary depending on the algorithm, but the objective is always the same. What changes is the level of precision with which the sentiment prediction is presented [71]. With some algorithms, there is also the possibility to detect things like irony.

Sentiment analysis has evolved a lot since it was first created, and nowadays, multiple approaches can be used to do it, ranging from rule-based methods to machine learning techniques. Rule-based/Lexicon methods involve creating a set of predefined rules or patterns that match certain words or phrases associated with positive or negative sentiment. The polarity of a text will be an average sum of the polarity found for each word that matches a rule. While this approach is straightforward, it often lacks the flexibility to handle complex language nuances and context.

Machine learning approaches, on the other hand, have gained popularity in sentiment analysis due to their ability to handle the inherent complexities of natural language. These techniques rely on large datasets to train models that can automatically learn patterns and features indicative of sentiment. Supervised machine learning algorithms, such as SVM, NB, and deep learning models, like RNNs and CNNs, are among the most commonly used for sentiment classification tasks, depending on the user end goal and available time and resources [71].

The process of performing a sentiment analysis task usually involves several steps. First, the text data must be preprocessed by removing unnecessary content, such as punctuation, emojis, stop words, and URLs, which are things that matter to humans but are irrelevant to computers. Next, the text is tokenized into individual words or phrases, and each token is assigned a sentiment polarity (positive, negative, or neutral) based on the training data. Feature extraction techniques, such as bag-of-words or word embeddings like Word2Vec or GloVe, explained in Subsection 2.2.1, are often employed to capture semantic meaning and context. Once the features are extracted, a

sentiment classification model is trained using labeled data. The model is then evaluated using adequate performance metrics. After the model is deemed satisfactory, which implies results at least better than 50% for the desired performance metric (better than a random guess [111]), it can be applied to new, unseen data for sentiment prediction.

However, just like explained for text classification in general, sentiment analysis also has its challenges [20]. Language ambiguity, sarcasm, irony, and cultural nuances can make determining sentiment difficult and sometimes give the exact opposite result from what is the truth. The accuracy of sentiment analysis models heavily depends on the quality and diversity of the training data, as well as the robustness of the feature representation.

To overcome these challenges, researchers keep exploring new techniques and approaches, not only on the classifiers but also on how things like preprocessing are dealt with. Approaches such as transfer learning, domain adaptation, and ensemble methods (explained in the next section) can improve the performance of sentiment analysis models. Transfer learning is a technique in machine learning where knowledge gained from one task or domain is applied to improve performance on another related task or domain [114]. It involves using a pre-trained model as a starting point and fine-tuning it for the new task, leading to improved performance, reduced training time, and lower data requirements. Domain adaptation, in the context of machine learning, is the process of adapting a model trained on a source domain to perform well on a target domain, where the source and target domains have different distributions of data [36]. It involves mitigating the domain shift by learning domain-invariant features or aligning the source and target domains to improve the model's performance on the target domain.

2.2.4 Topic Modeling

Topic modeling is a popular technique in NLP that aims to discover hidden topics within a collection of documents. It is a statistical modeling approach that assigns topics to documents based on word co-occurrence patterns. It provides a way to understand and organize large volumes of textual data by identifying the main themes or subjects present in the documents [57].

The history of topic modeling [26] can be traced back to the late 1990s with the development of algorithms like Latent Semantic Indexing and Probabilistic Latent Semantic Analysis. These early models used matrix factorization and probabilistic modeling to uncover latent topics. However, the breakthrough in topic modeling came with the introduction of Latent Dirichlet Allocation (LDA) in 2003 [13]. LDA became the standard for topic modeling due to its simplicity and effectiveness.

The most common approaches to topic modeling include LDA and its variations. LDA assumes that each document is a mixture of topics, and each topic is a distribution of words. The model uses probabilistic inference to estimate the distribution of topics in the documents and the distribution of words in the topics [13]. Other popular models include Non-negative Matrix Factorization, which approximates the document-term matrix using non-negative factors, and Hierarchical Dirichlet Process, an extension of LDA that allows for an unbounded number of topics.

There have been several advancements and extensions in topic modeling in recent years. One notable development is the incorporation of neural networks into topic modeling frameworks.

Variational Autoencoders and Generative Adversarial Networks have been combined with topic modeling to improve the quality of generated topics and enable more flexible modeling. Neural topic models, such as the Neural Variational Document Model and the GAN-based Topic Model, have shown promising results in capturing intricate topic structures. There are also already BERT models adapted to perform this type of task.

Despite the progress made in topic modeling, several challenges are still associated with this field. One challenge is the determination of the optimal number of topics. Selecting the appropriate number of topics is crucial for meaningful interpretations and effective organization of the documents. Still, it can be complicated to understand which is the best number since this varies a lot depending on the text studied. Understanding the topics generated by the models is another challenge, as topic quality and coherence can be subjective and difficult to assess automatically. Sometimes the words alone might not be enough for someone to understand a topic, a good example of this would be a list with just names of football players from a team but without a team name. Furthermore, topic modeling often struggles with short and noisy text, such as tweets or chat messages, where the lack of contextual information and limited word usage can impact the accuracy and interpretability of the topics.

Topic modeling can be a valuable technique in NLP that allows discovering latent thematic structures within a collection of documents. Recent advancements have incorporated neural networks into topic modeling, leading to more sophisticated and flexible approaches. However, challenges remain in determining the optimal number of topics, evaluating topic quality, and handling short and noisy text. Overcoming these challenges will further enhance the capabilities of topic modeling and its applications in various domains, including information retrieval, content analysis, and recommendation systems.

2.3 Ensembles

At its core, Ensemble Learning consists in a Machine Learning approach in which a set of models that might not be so good individually are combined to give a more accurate prediction [118]. However, if the algorithms are not chosen carefully, the results can be worse than for the individual approaches. For this to not happen, choosing accurate/precise, and diverse models is important. Otherwise, the predictions will be almost identical, and the increased resources will not reflect on the final prediction.

Ensembles can be applied both in regressions tasks and classification one [118]. They can also be homogeneous and heterogeneous. Homogeneous ones consist of a collection of classifiers of the same type that use different datasets. At the same time, heterogeneous ensembles work with different types of classifiers that are fed the same type of data. The algorithms used in ensembles are also called learners and can be classified as weak or strong. Weak learners are expected to have accuracy values (or other similar performance metrics) slightly above average (50% in accuracy case) since less than that would be worse than a random guess and just be inadequate. On the contrary, strong learners must already have robust performance metrics results independently.

Ensembles can also diverge at other points. Some can be parallel, while others can be sequential. Being parallel means each learner makes his prediction individually, so all the learners can run simultaneously, and when all are finished, their predictions can be combined. Meanwhile, in sequential ensembles, the output from each learner will feed the next one, so learners run one at a time, making this approach slower but less resource dependent [18].

Most ensembles can usually be fitted into one of three categories: Bagging, Stacking, and Boosting [18]. Figure 2.4 provides an example of how these categories of ensembles diverge.

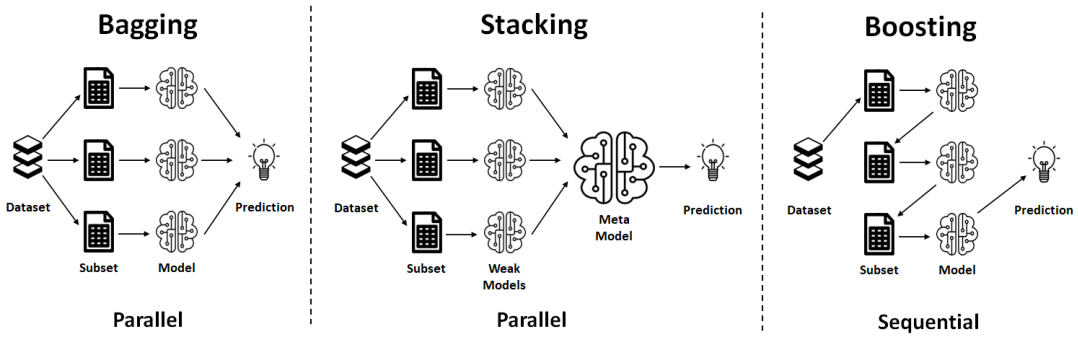

Figure 2.4: Types of ensembles.

Regarding these three categories introduced, bagging, boosting, and stacking, since the remaining categories are ramifications of these three, it is important to talk in detail about each one of them:

- **Bagging**: Short for Bootstrap Aggregating, involves training multiple models independently on different subsets of the training data and then combining their predictions to obtain the final result. Each base model is trained independently on its respective subset, and their predictions are combined using techniques like averaging or voting. Bagging helps reduce the variance of the ensemble by introducing diversity through different subsets of training data. It is particularly effective when the base models are prone to overfitting, or the dataset is limited. Random Forest is a popular bagging algorithm combining decision trees with an averaging approach, providing robust predictions and feature importance measures [18].

- **Stacking**: For this approach, the idea is for each model to give its prediction and then for a final meta-model to be trained from all these intermediate predictions[3]. It learns to make predictions based on the outputs of the base models, which is a good approach for models that cannot learn the entire problem but only an extent of it. The key idea behind stacking is to exploit the individual strengths of different models and let the meta-model learn how to combine them best. The base models can be diverse regarding the algorithms used, hyperparameters, or even feature representations [18].

- **Boosting**: It is an ensemble learning method combining weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model, and then trained sequentially so that each model can try to fix the weaknesses of

its predecessor. Each weak learner is trained on a modified version of the training data, where more weight is given to the instances that the previous learners misclassified. This iterative process allows subsequent weak learners to focus on the instances that are harder to classify, thereby gradually reducing the overall error of the ensemble. Some of the already implemented boosting algorithms include AdaBoost, Gradient Boosting, and XGBoost [18].

2.4 Summary

This chapter explained vital concepts for understanding the literature review and the work developed. First, it was possible to understand that multiple social media networks are available, each with different purposes and characteristics regarding, e.g., their content and privacy policies. Then, the NLP area was explained, starting by explaining what embeddings are, how they can be generated, and what they can be used for. After detailing the importance of embeddings, text classification, sentiment analysis, and topic modeling were clarified, and it was possible to understand some of the older and newer approaches and some of the challenges faced when performing these tasks. Lastly, the concept of Ensemble was explored, with particular attention to the three main approaches: bagging, stacking, and boosting.

Chapter 3

Transport Network Evaluation Using Social Media Data

The present chapter explains what other studies have been conducted for subjects related to this project. First, there is an explanation of how the systematic review was conducted (Section 3.1). Next, the results of the search are presented in a table that summarizes the studied papers, followed by subsections that explain the most relevant parts of this research (Section 3.2). At the end of this section is a subsection highlighting the most common problems the authors faced during their work. The last section (Section 3.3) summarizes the highlights of this chapter.

3.1 Methodology

A systematic review was conducted to help better understand the topic in question and study different methodologies to deal with the identified topics. This review uses papers that relate transport networks/companies and social media data with text classification (TC), sentiment analysis (SA), and topic modeling (TM).

Since this is a relatively new area of study, it was considered that only studying papers that included all the NLP topics (TC, SA, TM) referred to was too restrictive. Therefore, if a work only discusses one of them, that is considered enough for a study to be relevant to this analysis.

The systematic review search was done on the Web of Science [1] platform, using the following query to restrict the papers:

(sentiment OR "opinion mining" OR "topic model*") AND ("social platform" or "social network" OR "social media" OR "personal opinion*" or twitter) AND ("transit OR transport* or rider or "smart cit* or "traffic congestion").

[1] https://www.webofscience.com/wos/woscc/basic-search

The title, abstract, and author keywords fields were the chosen parameters for the search. This filtering was also done using the *OR* condition, so the search could return results related to one or more of these fields to be eligible.

The last query search to retrieve papers for the review was performed in May. During this search, the query yielded two hundred-one different results (30/05/2022).

Figure 3.1 demonstrates the distribution of papers per year according to their publication date. It is possible to understand that this is still a new subject, with the oldest paper being from 2012. It is also possible to notice a clear increase in interest in this subject with more papers published in recent years.

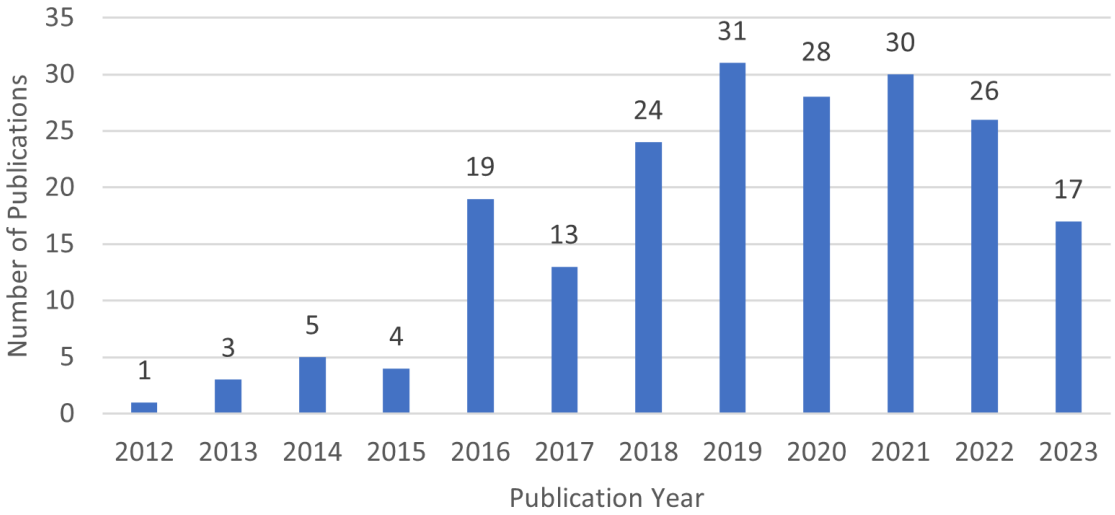

Figure 3.1: Papers count distribution according to the year of publication, retrieved from Web of Science on 30/05/2023.

After using the query, it was still necessary to define the inclusion criteria so that the number of results could be narrowed and that only papers helpful for this review would be studied in dept. The selection of the studies was made with the following criteria:

1. Papers written in English.
2. Papers based on transport network evaluation using social media data or some similar source (short texts).
3. Papers that use text classification and/or sentiment analysis and/or topic modeling.
4. Papers only related to land transportation.
5. Papers with well-defined methodologies (algorithms used, hyper-parameters, and results analysis).

3.2 Search Results

After using the inclusion criteria to filter the results obtained by the query search, the final number of papers used for the systematic review was twenty-four.

Although not every paper covers all these topics, most of the works talk about: the data extraction process, in which the authors decide how they are going to retrieve the information needed to evaluate the transport network/service in question; the preprocessing, that consists in cleaning all the information and preparing it to be used in the following tasks; the text classification, that can be just identifying which content is related with transports and which is not, or it can be more complex and divide the ones related with transports into different categories (e.g., bus, trains, highways); sentiment analysis, determine the polarity for each text and detect possible problems that might be related with negative feelings; and topic modeling, used to understand the most relevant topics that are being written about.

Table 3.1 and 3.2 shows a summary of the papers, with their most relevant points, like their objective, chosen location, amount of retrieved data and the period in which this was done, and approaches used for the different tasks used in the evaluation.

Table 3.1: Papers reviewed during the systematic review with NLP tasks applied to the domain of transportation networks.

Ref.	Local	Source	Data			Text Classification				Sentiment Analysis		Main goal
			Period	Year	N	Approach	Assumption	N.º of keywords	Data manually labelled	Method	Polarity	
[113]	Boston	T	1 day	2018	5K	GS-DMM	-	10	-	-	-	Use citizens social media data to detect events
[31]	-	TRB Meeting Papers	7 years	2008/2014	18.357K	LDA	-	-	-	-	-	Extract the most important topics in papers about transportation
[59]	California	T	3 weeks	2016	10.4K	LDA	-	17	-	VADER	P,N,E	Evaluate users' opinions quality of service
[43]	Salt Lake City	T	1 week	2017	403	LDA	VEM	-	-	RSentiment	P, N, VP, VN, S, E	Mining the public opinion on transportation systems using social media
[107]	Santiago do Chile	T	3 years	2014/2016	110K	LDA	-	118	-	SentiStrength	P, N, E	Level of satisfaction of bus users
[62]	Nanjing	SW	4 years	2014/18	50.970K	SVM + LDA	-	-	-	ROSTCM 6	P,N	Discover the public opinion about the metro system
[96]	Miami - Dade County	T	1 year	2017/18	430.201K	SVM, DT, NB	Stem	-	1000	AFINN	-5,5	Evaluating public opinions on transportation networks
[30]	Rabat Marrakech Casablanca	T, F	-	-	4750	-	-	-	4750	SVM, DT, NB, KNN	P, N	Analyze user's opinion about the traffic in three cities
[9]	-	T	-	-	565	-	-	-	565	NB, KNN	P, N	Understand which algorithm is better for sentiment analysis of tweets related with transportation services
[87]	Madrid	T	2 months	2019	27.603K	LDA	-	-	-	BERT	0.1	Use Twitter data to link itransport complaints to space
[100]	New York	T	3 months	2016	353.807K	-	-	-	-	Dictionary	P, N, E	Enriching traffic information using social media data
[78]	Kenya	T	6 years	2015/2021	770K	-	-	-	-	BERT	P, N, E	Use social media data to highlight problems with Kenya traffic safety
[101]	USA & Canada	T	4 years	2010/2014	63K	-	-	10	-	Lexicon	P, N	Understand users opinions about public transportation companies and search for relations between this opinions and the way entities comunicate online
[23]	Shangai	SW	8 months	2012	14.5M	-	-	28	6k	-	-	Trying to track congestion and accidents using social media data
[10]	-	T	-	-	-	-	-	14K	-	VADER	P,N,E	Identify accidents and extract information about them using social media and APIs
[6]	Greater Manchester	T	2 days	2020	27 878	Linear SVM	-	-	-	SentiStrength, DeepAI API, GotIt API	P, N, E, N, P, VN, VP, E, P, N, E, C	Investigates the relationship between the actual transport network status, and tweets sentiments

Sources: F: Facebook, SW: Shina Weibo, T: Twitter.

Classification approach: SVM: Support Vector Machine; DT: Decision Tree; NB: Naive Bayes; KNN: K-Nearest Neighbor; LDA: Latent Dirichlet Allocation; GS-DMM: Gibbs Sampling Dirichlet Multinomial Mixture; VEM: Variational Expectation Maximization.

Sentiment Analysis approach: VADER: Valence Aware Dictionary and sEntiment Reason; ROSTCM 6: ROST Content Mining System Version 6.0; BERT: Bidirectional Encoder Representations from Transformer.

Polarity: P: Positive, N: Negative, E: Neutral, VP: Very Positive, VN: Very Negative, S: Sarcasm, C: Confusing.

Table 3.2: Papers reviewed during the systematic review with NLP tasks applied to the domain of transportation networks (Continued).

Ref.	Local	Source	Data			Text Classification				Sentiment Analysis		Main goal
			Period	Year	N	Approach	Assumption	N.º of keywords	Data manually labelled	Method	Polarity	
[4]	-	T	-	-	1.4k	-	-	-	-	SentiWordNet	P,N	Rate the transportation system entities using content from social media
[110]	Portugal	T	2 days	2016	420	-	-	1	-	Lexicon	P,N, E	Develop a framework capable of perform automatically content analysis for transportation services Applied in the context of a manifestations against uber
[8]	Indonesia	T	2 months	2016	126.4K	-	-	4	-	SVM, NB, DT	P,N	Measure GO-JEK and Grab customer satisfaction through Twitter data.
[119]	Turkey	T	5 years	2016/2021	600K	-	-	1	-	Turkish BERT	P, N	Exploring the barriers and drivers for biking from citizens' perspectives and making suggestions for improvements
[67]	Shenzhen	SW	11 months	2018/2019	177K	-	-	2	-	SKEP	P, N	Understand the factors that influence customer satisfaction of transit services
[77]	India	T	11 months	2019	40K	-	-	7	-	Lexicon Rsentiment SentimentR	P, N, E	Identifying improvements in the service quality of the rail transport industry services in India
[24]	New York City	T	2 years	2020/2022	300K	BERT	-	24	-	-	P, N	Understand travel mode choices during the pandemic and measure satisfaction
[82]	Jakarta	T	2 years	2020/2022	21K	Manualy	-	-	-	SVM	P, N, E	Examine the public opinion, particularly Jakarta Bus Rapid Transportation users

Sources: F: Facebook, SW: Shina Weibo, T: Twitter.

Classification approach: SVM: Support Vector Machine; DT: Decision Tree; NB: Naive Bayes; KNN: K-Nearest Neighbor; LDA: Latent Dirichlet Allocation; GS-DMM: Gibbs Sampling Dirichlet Multinomial Mixture; VEM: Variational Expectation Maximization.

Sentiment Analysis approach: VADER: Valence Aware Dictionary and sEntiment Reason; ROSTCM 6: ROST Content Mining System Version 6.0; BERT: Bidirectional Encoder Representations from Transformer.

Polarity: P: Positive, N: Negative, E: Neutral, VP: Very Positive, VN: Very Negative, S: Sarcasm, C: Confusing.

Since the query used to search for the papers was applied to three different fields, title, abstract, and author keywords, it is important to understand the most common words associated with the articles that were studied during this review. To do this, word clouds representing the titles, abstracts, and authors' keywords can be observed in Figure 3.2, Figure 3.3, and Figure 3.4, respectively. As it is possible to see, social media and particularly Twitter/tweet are common terms in the three images, together with the word "data", since this is the foundation of any work. Machine learning tasks like sentiment analysis, text classification, and topic modeling can also be found and therefore are relevant to be analyzed during this literature review.

Figure 3.2: Word cloud of articles' titles.

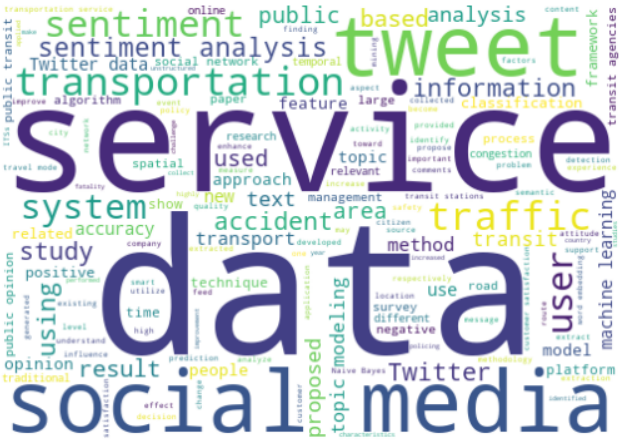

Figure 3.3: Word cloud of articles' abstracts.

The following subsections analyze each of what are considered the most important topics for the subject studied. This analysis compares the different approaches studied during the review, identifies them, and points together similar use cases.

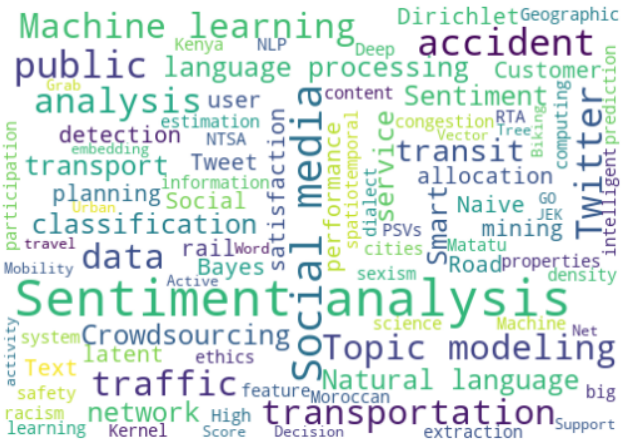

Figure 3.4: Word cloud of articles' authors' keywords.

3.2.1 Data Extraction

As explained in Section 2.1, multiple social networks are available and can be used for microblogging, so their data is a great source of information for the type of work reviewed. The papers analyzed during this review cover three of them: Facebook (comments), Twitter, and Sina Weibo.

Facebook is only used in two papers [30, 44] where information is extracted directly from accounts related to transports. An example of this is collecting comments in publications from the subway company account of a respective city, which people usually use to complain about problems like delays. For the remaining authors, Twitter or Weibo is a much better choice due to how these social networks are structured, encouraging users to write constant updates on their pages. These two social networks are also easier to extract information from by using an API or a web scrapping tool.

The specific city often influences the choice of data sources under study. As Weibo is predominantly utilized in China, this social network is usually the most used in Chinese studies. Consequently, papers like [62, 23] focus on specific Chinese locations, such as Nanjing city, and rely on Weibo as their primary data source. In other geographies, researchers use Twitter not only for the reason explained previously but also due to the language barrier that would exist for non-Chinese persons working with content written in Chinese.

The language extracted is also mostly related to the city/zone chosen for analysis. Most papers work with English content [113, 59, 43, 96, 100, 6], some deal with content in Spanish [87, 107], Arab [30], Chinese [62, 23], Indonesian [98, 8] and Turkish [119].

The extraction process possibilities are detailed in Figure 3.5, and it can be done in three ways: manual extraction, using the application programming interface (API) provided by each social network, or implementing a web scraping tool from scratch.

Although it is referred to, none of the authors use the manual approach because it would be time-consuming for thousands of microblogging texts. Some papers like [59, 43, 87] use the API approach that is made available by Twitter itself [1]. Using a language like Python or Java, this

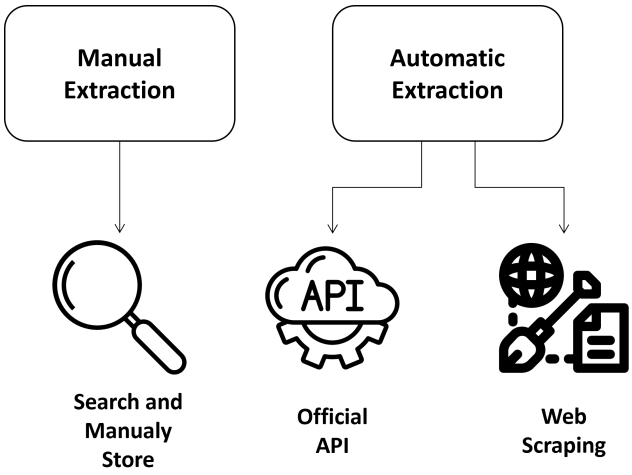

Figure 3.5: Diagram with the extraction possibilities according to the literature reviewed.

API can then be used to automatically download the tweets in a known format and with all the information that Twitter allows users to have access to. The remaining approach, using a web scraper, was adopted by authors like Mendez et al. [107] and Qi et al. [96] due to restrictions regarding the Twitter API. According to the authors, depending on the user profile, this API restricts the number of tweets that can be downloaded each month and the rate at which this can be done. Implementing or using a web scraper is probably the best choice for heavier research. A web scraper searches Twitter automatically and stores data retrieved from the hypertext markup language code of the page [93]. Although this allows users to store more content than when using the API, it has limitations regarding the information that can be extracted, like the location of each text.

For some authors [113, 59, 43, 62, 96, 87, 100], the filtering process for content related to transports starts during the extraction phase. Two different approaches are used only to extract useful information. One approach is searching for content using hashtags or words related to different means of transportation, like a specific bus company name or a highway code. The number of words can vary, depending on the level of precision required. While some only use one word, others use more. These words or hashtags can be general transportation words, such as ”bus”, ”street”, or ”crash”, or more specific ones, usually related to public transportation companies [107, 87] or private ones [110].

The other approach is researching accounts related to transportation and then only downloading their tweets or responses from other users.

The last part of the extraction process is related to the location of the tweets. To correctly evaluate a particular network, it is essential only to use tweets that talk about the zone covered by it. Part of the works [107, 30] do this by using a similar approach to what some authors do to extract only tweets related to transportation, which is using words or hashtags related to the location pretended (e.g., "Fifth Avenue"). Other authors like Witanto et al. [113] choose a technique that can only be done when using the Twitter API, which restricts the search used for

the extraction to a certain location, like a city or a district. According to their paper, this approach has some problems because tweets do not always have a location associated. Sometimes this location is incorrect or might not be helpful due to a user being on the move and not tweeting right after he saw or experienced something.

3.2.2 Data Preprocessing

After extracting the content, it is necessary to do preprocessing so the raw data can be used in the following stages of the chosen methodology. This phase can usually be divided into three steps: data cleaning, transformation, and reduction. Sometimes an extra step can be considered between the cleaning and the transformation, which is data integration. However, since in the reviewed work, only two papers use data from multiple data sources [30, 44], and no information is given about the merge process, it is not relevant to talk about this step.

The preprocessing is done almost the same way for every paper studied during the systematic review. Most authors start by removing unnecessary white spaces that are only common between words since most social networks removed them at the start and finish of each published content. URLs, user tagging, punctuation, special characters, and emojis are also removed since they do not contribute anything to text classification or sentiment analysis.

In some cases [43, 107, 87], the letters are also converted to lowercase. Almost every programming language has a package or library to do this since it is relatively common when doing NLP.

Some authors also decide to remove stop words [43, 87]. These words do not add anything to a text when an algorithm processes them. Determiners, coordinating conjunctions, and prepositions are examples of this. Removing these words saves computing time and power.

Unlike all the other methodologies, during the preprocessing, Qi et al. [96] and Ali et al. [5] also transform every word into its stem. The stem is responsible for the lexical meaning of a word, and for most NLP techniques, it is only what is needed to infer something from a word.

Lemmatization can also be found in some works [5], and it consists in reducing words to their base or canonical form, known as a lemma. The process involves transforming inflected or derived words into their dictionary or base form to unify words with similar meanings. For other authors [87], this is considered possibly problematic since it could interfere with their pattern recognition techniques.

Regarding possible translations, there is important information in the paper written by Dahbi et al. [30], in which a translation of Arabic characters to Latin ones (Arabish) and from the Moroccan dialect to the standard Arabic is performed. Although the difference in accuracy (for sentiment analysis) with translated content versus non-translated is only 1% this might still be helpful for other situations like the use of slang, which is very common among younger generations.

More specifically, for the papers that use data from Twitter (tweets), there are two approaches to deal with retweets, the word used to describe "sharing" someone's post. Some papers [43, 107] delete all the retweets from the extracted data because they consider it duplicated information and irrelevant for analysis. However, the remaining authors believe that a retweet can indicate how

valid information is, so repeating a tweet can help with the final evaluation made for a network. Taking as an example traffic congestion, if someone tweets about it and then persons retweet it, this can be taken into consideration as if eleven persons tweeted about the congestion. In Mendez et al. [107] work, tweets from official media and transportation-related accounts are also removed since they do not represent individual opinions.

Finally, regarding data enhancement, when the location is unavailable for a specific text, if possible, some authors [107, 87] use a lexicon approach to add this information to each dataset entry. This is done using a dictionary of transportation-related terms, like train station names or bus stops. If a word from the dictionary is present in a text, then the location for that word can be associated with the entire microblogging content. According to the papers, this information is not perfect. Still, it can help get better results during the evaluation process since it gives a more precise notion of where the opinion comes from.

3.2.3 Text Classification

As previously explained in the Subsection 3.2.1, some methodologies already try to filter the content during the extraction process, so they only get microblogging related to transports. However, the methods presented are not perfect, making it usually necessary to apply some algorithm to classify texts as relevant for transports or not.

Some authors try to do this classification using unsupervised topic modeling (a task explained more in detail ahead) and then select the content from the topics they assume relate the most to transportation. The most common approach to unsupervised classification is the Latent Dirichlet Allocation [13]. The papers by Osorio et al.[87] and others[31, 43, 62, 59], all use the LDA approach, but what changes is the number of topics each paper considers. This number of topics can be decided by multiple tries or by using an optimization algorithm like Variational Expectation Maximization [51] used by Haghihi et al., or the Robust Probabilistic Counting methodology used by Mendez et al., a heuristic algorithm used to determine the optimal number of topics [117]. It is important to highlight that for all the papers, one of these topics tends to include all the non-transports-related content.

The number of topics used varies a lot depending on the detail of classifications authors want, with some numbers being two, five, ten, twenty, and fifty. Kulkarni et al. [59] implemented the LDA approach by using the Java MALLET framework. However, this information is not usually provided in the research work. Kulkarni et al. also tried different topic numbers, three, but each trial's relevant topics were identical. For this type of approach, Osorio et al. [87] was the only study that included a performance metric result, the accuracy, which was 69%. The other unsupervised approach used is Gibbs Sampling Dirichlet Multinomial Mixture (GS-DMM) [115], which Witanto et al. [113] use as an alternative.

As a supervised approach, Support Vector Machine (SVM) is one of the options [62, 96, 6] used for transports-related text classification. Li et al. [62] trained it using labeled data and divided the 50 970 microblogs into four categories: traffic evaluation, information reporting accounts, traffic demand accounts, and irrelevant data. Meanwhile, Qi et al. [96] used the SVM to classify

transport-related content into categories like "Bicycle" or "Boat". Besides SVM, their work also used Decision Trees and Naive Bayes, however, there is no comparison between the accuracy of each algorithm. The work developed by Almohammad and Georgakis [6] also used an SVM, but it was a more specific version (Linear SVM), used when the classification can only have two possible values, in this case, related or not with transportation.

A few works [96, 10] follow a dictionary approach, which is explained by Gal-Tzur et al. [38] and it consists of using a lexicon with multiple words manually labeled. Each word is graded on a scale from 0 to 5, with five representing the maximum relation level with transports.

Unfortunately, due to the dimension of the data used during these approaches, most papers do not present evaluation metrics for the chosen algorithms, making it difficult to compare and understand which ones are the best methods.

3.2.4 Sentiment Analysis

As explained in Section 2.2.3, Sentiment Analysis is also an NLP task. SA algorithms can give quantitative or qualitative results. Qualitative results usually have one of three values: Positive, Negative, or Neutral. Sometimes the Neutral option does not exist. Quantitative results have a scale that goes from a negative value (negative sentiment) to a positive value (positive sentiment), with the values close to zero representing the neutral result.

Dahbi et al.[30], Atmadja et al.[9], Osorio et al. [87], and Ali et al. [5] use a fine-tuned approach, in which additional training was done with data specific to the problem context. The algorithms found in their works are SVM, Naive Bayes, K-Nearest Neighbors (KNN), Decision Trees (DT), and Logistic Regression (LR).

The first authors used four different algorithms to tackle this problem, and SVM got the best result, with 94% accuracy. The second paper compares the KNN and the NB algorithms, with KNN having a 67.7% accuracy and NB a 66.2%. However, the training and the testing dataset size were deficient, with only 500 tweets to train and 65 to test, which according to the authors, might be a reduced number to make conclusions. On the other hand, Osorio et al. used the Bidirectional Encoder Representations from Transformers (BERT) [34] technique. The model was fine-tuned using two datasets, one with random data and another related only to transports. The tests showed that it could detect the polarity of a tweet (1 - positive sentiment; 0 - negative sentiment) with an accuracy of 90%.

The majority of papers in the field use established algorithms or dedicated tools for conducting sentiment analysis. However, in some works the authors implemented their own sentiment analysis method, using a lexicon approach to make the classification (e.g. [110, 77]). No reasons were given for this decision, and it can even be considered abnormal since there were already lexicon solutions available that allowed adding, removing, and changing entries of the lexicon. However, it might be possible that the authors wanted more control over the polarity calculus, so a new implementation was the best choice for them.

It is worth noting that each paper often employs a distinct approach, leading to a wide range of methodologies being utilized throughout the research. Consequently, the methodology section of each paper provides an explanation specific to the chosen algorithm or tool.

The first group, lexicon methods, is the most common and includes several options. Valence Aware Dictionary and sEntiment Reasoner (VADER) [48] was used by two authors [59, 22], and it is a lexicon-based algorithm that uses dictionaries and rules for each type of sentiment (positive, neutral, or negative). Each word gets a value, and the sum of all the values gives the text result. For the authors, tweets with a neutral score greater than 0.8 were considered neutral. Santos et al. [100] use a similar approach for their Route Sentiment tool.

The Rsentiment" package [14] was used in two projects, Haghighi et al. [43] and Mishra et al. [77], with the first implementation being done using the language R. The algorithm from this package returns a qualitative result that can have one of six different values: Positive, Negative, Very Positive, Very Negative, Sarcasm, and Neutral.

Li et al. [62] uses a Chinese tool called ROST Content Mining System Version 6.0 (ROSTCM 6), a specific tool for dealing with Chinese characters. This tool performs a binary evaluation, attributing only a positive or negative sentiment to a text.

Qi et al. [96] use the AFINN lexicon [81]. It contains more than 3300 words, and each word has a score from -5 to 5, with the negative numbers representing words associated with negative polarity and the positive numbers the opposite. The polarity of a text is the resulting sum of each word value.

The second group of approaches important to highlight is the machine learning one. Starting with a version of BERT pre-trained for sentiment analysis, called twitter-roberta-base-sentiment, it was one of these types of the approach chosen by an author [78].

Sentiment Knowledge Enhanced Pre-training is only used in one project [67], and it is an advanced framework developed for sentiment analysis, which aims to capture fine-grained sentiment information from text. It utilizes large-scale pre-training on unlabeled data, combined with knowledge-enhanced techniques, to improve the accuracy and depth of sentiment analysis.

In Mendez et al. [107] the initial tool chosen was SentiStrength [102], which combines both lexicon and machine learning techniques to improve the prediction. According to its creators, this tool can report: two sentiment strengths, a quantitative value for positive and negative feelings; qualitative binary or trinary (positive/negative/neutral); and a single scale (-4 to +4). However, the authors worked with Spanish content, and the accuracy was only 41% for it, so they decided to do the sentiment analysis manually.

Lastly, the Almohammad and Georgakis work [6] uses three different approaches: Senti Strength (previously explained); DeepAI API sentiment analysis API that classifies each text as very negative, negative, neutral, positive, or very positive; and GotIt API that also gives a qualitative analysis that can be positive, negative, neutral or conflicting (considered the same as neutral by the authors). The paper does not provide a comparison between these three approaches.

Like in the Text Classification, most papers do not present comparisons or common performance results for each algorithm, tool, or technique used. However, it was still helpful to read

them and understand how many different possibilities exist to deal with this task that is part of the work conducted during the dissertation.

3.2.5 Topic Modeling

Another very important task for some of the authors is topic modeling, which, as explained earlier, can be used for classifying the text as related to transports or not, but can also be used during the transport networks evaluation to give more context about the topics users are talking about, and possibly identify a specific event that happened or is happening.

The first approach identified [78] for Topic Modeling was Structural Topic Model. It combines the strengths of topic modeling and network analysis to uncover hidden themes and relationships within a corpus by considering both the content of documents and the connections between them, providing a holistic understanding of complex datasets. It allows for the discovery of what is being discussed and how different topics relate to one another.

LDA is probably the most common approach used [62, 5, 59, 43, 96]. Some projects use it to identify topics in general [62, 5, 59, 43], while others try to use it to separate content according to the mean of transportation [96], e.g., bus or train.

The Machine Learning for Language Toolkit (MALLET) is also one of the approaches present in the literature[107], used for natural language processing and text mining. It provides a suite of tools and algorithms for tasks like document classification, topic modeling, and information extraction. It is widely used in research and industry due to its efficiency, flexibility, and support for various machine-learning techniques. The number of topics varies according to the authors' objectives.

There is not much diversity regarding topic modeling. Most authors tend to use the LDA approach, probably because it is so popular for TM in general. As for the number of topics, most authors do not provide an exact reason for the chosen number, and some describe the process of choosing it as trial and error.

3.2.6 Microblogging Challenges and Opportunities

During the analysis of existing literature, it became evident that a majority of authors face common obstacles in their research endeavors, i.e., data size, lack of location information, and lack of data. One such challenge revolves around the size of the content used. Microblogging platforms typically host short-form content, which poses difficulties when employing algorithms designed for longer sentences. Consequently, obtaining accurate results for essential Natural Language Processing tasks like sentiment analysis or text classification becomes a challenging task.

Another problem, referenced particularly by Almohammad and Georgakis [6], is the lack of precise location information. Because most users decide not to disclose their location, being unable to associate a tweet with a restricted area makes it harder to accurately evaluate a specific part of a network since results can get mixed up. Sometimes the location available is also very vague, e.g., city name; therefore, only macro analysis can be conducted.

Occasionally the lack of data is also a problem because it can lead to an evaluation being a product of a small number of tweets that can have some bias or even be false. Méndez et al. [107] and Osorio et al. [87] studies show that the amount of data is unequal throughout the day, with clear peaks during rush hours. Haghighi et al. [43] also highlighted that low-income people and senior residents might not have a device that allows them to micro-blog, which means a significant portion of commuters/drivers' opinions is not considered during these evaluations.

As for the authors' conclusions, some investigators [87, 62, 43] were able to identify concrete problems in the train and metro networks studied, and they could even associate the problems with specific stations or lines.

Li et al. [62] and Méndez et al. [107] work showed that this type of methodologies can have more coverage than the typical surveys done to users. According to the authors, this additional coverage can be beneficial to detect problems in zones or services with less traffic.

Santos et al. [100] found what they believed was a correlation between the number of tweets and the congestion levels. Since they did not have access to the raw data from sensors (private information), they used the Jam Factor (JF) from HERE API [106] that gives a value between 0 (no congestion) and 1 (congested) for a given place at a certain time. However, contrary to their work, Almohammad and Georgakis [6] could not find a relation between the tweets and the atypical traffic events for their case study. This lack of correlation can result from errors caused by the problems explained previously in this subsection for these two authors.

All the authors analyzed insist that these methodologies have much potential and still have space to evolve. For them, the future work should pass by improving the chosen algorithms, either by using more data to train or by changing specific characteristics, like the number of topics when using an LDA approach, using libraries to translate certain abbreviations or slang, and combining the social media data with content other sources, like sensors.

3.3 Summary

The systematic review showed that most authors use the API provided by the chosen network for data extraction. Although some of them use the web scrapper methodology, this is due to the necessity of downloading massive amounts of data in a short period, which is usually blocked by restrictions. Twitter also seems to be the best choice when the intention is to evaluate transport networks located in Western countries. The keywords used during the extraction can vary a lot depending on the network being evaluated, but some words can be generalized, like "accident" or "congestion".

The preprocessing is almost always done the same way: removing unnecessary components, like white spaces, emojis, and punctuation. After this, all the letters can be converted to lowercase. The steps that can be conducted or not, depending on the algorithms or tools used for the nest phases (TC and SA), are removing stop words and converting every word to its stem. Depending on the text origin, a translation can also be done to avoid processing slang words or distinctive

dialects. Trying to enrich the data with location information has proven to be difficult for most authors, but it can be a good support for the evaluation phase.

Text classification can be done in many ways, but the LDA technique is the most common. This technique can be used to find the most relevant topics and then apply them in a supervised method either to distinguish between content related or not with transports or to identify possible topics that are part of the transports subject. The lack of evaluation metrics makes it difficult to compare papers and understand which methodology is more appropriate for the defined problem. Other traditional approaches like neural networks or SVMs are also common choices among the reviewed works.

The literature shows that Sentiment Analysis for social media content related to transports can be done using many different methods. Most tools perform very well with English content, but the accuracy tends to drop for other languages. Once again, due to the lack of common metrics for performance assessment and the different datasets used to test it is difficult to withdraw conclusions about which models are the best. So, trying multiple methodologies and seeing which one gives the best result seems to be the best course of action. Using a combination of SA algorithms results is also a possibility.

As for topic modeling, not many works use it to complement transport network evaluations but the ones that do tend to have LDA as their choice. The number of topics used varies a lot across works.

Eventually, the issues that can be faced are mainly related to the data used. According to the papers, this will likely be an even bigger problem for real-time evaluations, in which there is less available content, and therefore if the quality of the majority is not acceptable, it will not be possible to conduct good evaluations. Implementing trust lists can be a good decision to avoid being influenced by possible bias.

Chapter 4

Methodological Approach

As the previous chapters demonstrated, social media data has shown the potential to be used for automatic transport network status evaluations and atypical traffic event detection. However, the studies are still very broad, and an adequate pipeline that encapsulates the multiple approaches that can be used to contribute to this evaluation is still missing. Besides that, most studies focus on periodic evaluations instead of near real-time ones. Due to the content's dimension, lack of structure, and missing geographic information, these solutions are also far from what is believed can be achieved.

This study endeavors to enhance the evaluation of transport networks by leveraging social media data, enabling more accurate and near real-time analysis that is not limited to specific contexts. The main issues related to this problem are:

- For transports-related text classification, is it possible to group multiple solutions and develop one that makes the extraction more accurate, avoiding false positives and including more content by having fewer false negatives?
- For sentiment analysis, which of the solution found in the literature is the most adequate, and is it possible to group multiple solutions and develop one that makes the results more homogeneous and accurate?
- For topic modeling, can different topics be identified on social media content and can they be described with a sentence or a text instead of just a set of words?

To address these issues, the following chapter gives an overview of the methodology suggested to automatically evaluate the state of traffic networks. It is organized into three sections, each addressing a different aspect of the methodology approach thought for the project.

It starts by addressing all the different data used to develop and evaluate the methodology (Section 4.1). Then, the methods used to tackle each part of the problem are presented (Section 4.2), starting with the architecture chosen for the project. The architecture comprises three areas of study: (a) the transportation-related classification process with an in-depth examination of the possible ensemble approaches and each algorithm that incorporates them; (b) the different

sentiment analysis algorithms used in the literature, followed by a suggested solution to deal with the differences between them; and (c) the topic modeling and topics label, solutions are suggested to generate text that briefly describes each topic identified. This methods section finishes with an overview of the metrics and approaches chosen to evaluate these solutions.

4.1 Data

In this work, two subsets of data were used for different purposes. The first group, the social media content, is used to feed an automatic transport networks evaluator and study how this can be developed or improved. This is the data used not only to train different models but also to evaluate them. The second group, the data regarding events and traffic accidents, can be used for two different purposes: search for possible situations useful to assess how well the developed evaluation methodology performs and, in the future, for transfer learning when combined with the social media content.

To conduct this work, New York City was selected as the primary case study due to various compelling reasons. Firstly, the prevalence of English as the primary language in the city simplifies potential future applications of this solution in other regions. Moreover, the abundance of publicly available data from the city council and other organizations, which is regularly updated and accessible through platforms like the New York City Open Data[1], provides a rich resource for research purposes. This choice was also influenced by the comparatively less restrictive data privacy regulations in the United States compared to certain other regions, particularly those in Europe.

However, having data from diverse locations is crucial for comprehending the scalability of the proposed pipeline. Furthermore, analyzing specific cases, such as individual accidents or traffic delays due to events like parades, allows to gain valuable insights into the practicality and effectiveness of the developed system. By developing this work for various contexts, its usefulness can extend beyond a single city or country and potentially benefit a global audience.

In this work, to better understand how the algorithms perform, the results for only New York City were compared with another subset of data that additionally includes two cities that also have English as their primary language: Melbourne and London. Table 4.1 gives an overview of the cities, providing several facts crucial to understanding their dimension, the population's main characteristics and preferences, and the amount of content available to be used.

4.1.1 Social Media Content - Tweets

As the subsection title suggests, Twitter was the chosen social network chosen to support this project with microblogging data.

A tweet is a message consisting of up to 280 characters that can be posted on the social media platform Twitter. Users can share their thoughts, opinions, news, or any other type of information

[1] https://opendata.cityofnewyork.us/

Table 4.1: Studied cities characterization.

		New York	Melbourne	London
DOMAIN				
- Area (km^2)		738.4	9992	1572
- Main public transport		S, B	T, TR, B	S, B, T
POPULATION (2020)				
- Population (M)		8.3	5.8	8.8
- Population (people.km^2)		10,194	453	5,598
- Main language		EN	EN	EN
- Most spoken languages		EN, SP, CH	EN, CH, C	EN, B, P
- Average people's age		36.9	36	35.9
- Social media networks		T, F, I	F, I, TT	F, I, T
MESSAGES*				
Bounding box	South-West	-74.255641 40.495865	144.593742 -38.433859	-0.510365 51.286702
	North-East	-73.699793 40.91533	45.512529 -37.511274	0.334043 51.691824
- Messages available (M)		9.2	0.84	5.8

Messages*: Retrieved from the 16th of May to the 6th of July of 2017.
Languages: B: Bengali, C: Cantonese, CH: Chinese, EN: English, P: Polish, SP: Spanish.
Public Transports: B: Bus, S: Subway, T: Train, TR: Tram.
Social Networks: F: Facebook, I: Instagram, TT: TikTok, T: Twitter.

through tweets. A tweet can include text, photos, videos, links, and hashtags. It can be private, only visible and interactive to the user followers, or public, meaning anyone can share it (retweet), like it, or reply.

According to the website Internet Live Stats [105], over 500 million tweets are written daily, making this network perfect for extracting data that can be used for real-time evaluations. Also, due to a 280 characters restriction for each tweet, most users tend to use the app as a diary where they talk about various topics throughout the day.

The data provided for most tweets can is described in Table 4.2. About one percent of the extracted tweets also contain geo-location information, which can be a pair of coordinates, a bounding box, or just a location name.

Figure 4.1 shows an example of a tweet, screenshotted from Elon Musk's account[2], which currently is the majority owner of the social network Twitter. In this image, it is possible to see

[2] https://twitter.com/elonmusk/status/1663267596689350656

Table 4.2: Tweet components.

	Categories
Metadata	Author, date, location
Message content	Text, image, video, hyperlink, hashtag, explicit recipient, pole
Interactions	Retweets, favorites, replies, quotes, bookmarks, views

Figure 4.1: Mobile Twitter application (IOS) showing a tweet from Elon Musk's account.

the information that was described in Table 4.2. This screenshot was taken on the mobile Twitter application.

Twitter was the chosen social network because it is used like a diary by most of its users, it is available for free worldwide, so anyone with a device and access to the internet can tweet, and it has available an API for data extraction, even though now it has a cost associated. This makes it the ideal social network for extracting daily information used to feed projects like the one being developed.

Other social networks, like the ones introduced in Section 2.1 could also be used to support a project like this. Facebook comments or Weibo microblogs would be the next best choices following Twitter. Still, Facebook was excluded because the extraction process would be much more complex when compared with Twitter since most relevant content would be on comments and not on posts. Weibo was also not a viable option because most contents are written in Mandarin, a language in which fluency is lacking, which would make the manual classification for post-evaluation much harder. Other social networks, like Instagram, would involve a different scope, in this case, Computer Vision, which differs from the subjects intended for the project.

The importance of extracting meaningful insights from vast amounts of information cannot be overstated. By exploring the data, it is possible to gain a unique perspective that helps navigate complexities, uncover connections, and visualize the intricate fabric of information. More spatial and temporal information about the used Twitter data can be found in Pereira's work [92]. Since only a part of the data is relevant for this work, there is no point in replicating the already performed analysis.

4.1.2 Events & Traffic Accidents

In order to evaluate the effectiveness of the methodology when applied to real-life cases, it was essential to gather information on various situations that could potentially impact traffic. Consequently, this study involved collecting data on a range of events, such as sports games, concerts, and other special and permitted events, along with data on traffic accidents. Table 4.3 provides an

Table 4.3: Overview of New York City data sources and their characteristics.

		Sports Concerts Special Events	**Permitted events**	**Accidents**
Size (entries)		13.4K	16.9M	2K
Format		CSV	CSV	CSV
Number of parameters		13	12	21
Years		2010-2023	2008-2023	2017
Availability		Free	Free	Free
Date retrieved		02/2023	02/2023	02/2023
Update Frequency		-	As needed	Daily
Relevant fields	**Event type**	e.g., Concerts, basketball games, hockey games, boxing matches, graduation ceremonies	e.g., Sports youth special events, farmers market, religious events, sidewalk sales	-
	Location	Organization name, facility name, coordinates, state, county	Borough, address, street side	Coordinates, street name, address, borough, zip-code
	Date	Event registration time (date + hour), event close time (date + hour)	Setup time (date + hour), breakdown time (date + hour)	Collision time (date + hour)
	Persons involved	-	-	Nº killed/injured, nº pedestrians k/i, nº cyclists k/i, nº drivers k/i

overview of the different data sources chosen to support the developed work due to their characteristics and relevancy.

Events (e.g., concerts, sports matches, movie premieres, parades) were extracted from two sources: the New York State official website [3] and the NYC Open Data portal [4]. The first dataset [83] provides information regarding sports events, concerts, and special events since 2010. Although it is not specified by the entity that gives the data, special events seem to take part in large arenas but can not be classified as a concert or sports, like car races or theater shows. This information is updated, but the entity responsible for it does not specify the frequency at which this is done. The NYC Open Data has a dataset [84] with all the permits emitted by the NYC Council since 2008, and the dataset is updated as needed according to the responsible persons.

The Crash Mapper [5] is an online interactive dashboard, and it was used to identify traffic

[3] https://data.ny.gov
[4] https://data.cityofnewyork.us
[5] https://crashmapper.org/#/

accidents. This website has been compiling information about traffic incidents daily since 2011. It allows for multiple filters, with which users can choose between fatal and non-fatal accidents, injuries, type of person involved (cyclist, pedestrian, motorist), the period in analysis, type of vehicle, and boundaries (borough, intersections, districts, etc.). This information is updated daily, and it is available to be extracted for free on their website. Figure 4.2 displays the platform's appearance.

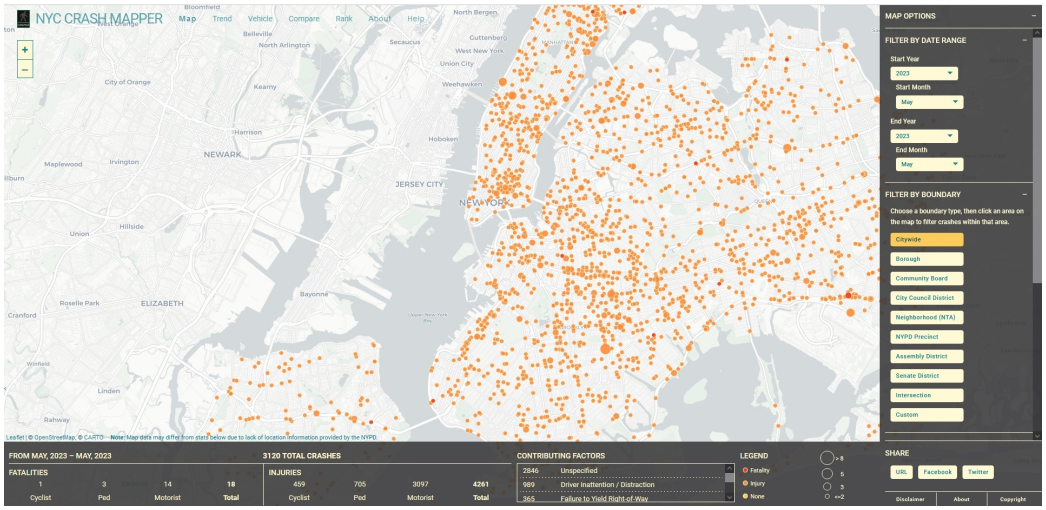

Figure 4.2: NYC Crash Mapper website interface.

Having data to classify the traffic state with a quantitative or qualitative value, regarding mainly congestion, would also benefit this work. This type of data is available from multiple sources, one of them being TomTom [6], but it is not free to use, even for academic works, so it could not be considered.

4.2 Architecture

The architecture considered for this project has two main sections: data extraction and preprocessing and evaluation of the transports networks' state. The work developed focuses on the second part, the evaluation since the extraction process was out of scope due to the use of a previously extracted dataset and the current limitations imposed by the current Twitter policies.

The network evaluation process suggestion comprises three main tasks. The information from each one complements the others so that the result can be as helpful and complete as possible for interested users. After obtaining the data from social media and preprocessing it, the first step is a classification task, used to select only the content related to transportation, which is the relevant data for the purpose in question. Sentiment Analysis is the following one, crucial to understanding how the public feels, mainly positively/negatively, regarding a particular topic or event; Topic Modeling is the last task, and it is combined with Text Generation so that it is possible to give a brief description of each topic, which can be a single sentence or a text that describe the current

[6] https://developer.tomtom.com/products/traffic-api

situations users are writing about. Combining the sentiment analysis result with this description should allow someone to identify what is happening and how heavily this affects the persons commenting on it.

After concluding each of these tasks, it is also crucial to evaluate the performance of the implemented pipeline. This performance assessment needs to be done both on an individual level, for each task, and on a macro level, for the pipeline as a whole. For the individual evaluation, it can be done using a subset of the available data, and by using calculating various performance metrics to understand how both the individual algorithms and the ensembles perform. The macro evaluation is more on a qualitative approach. It is carried out by applying the pipeline for periods and places in which real-life cases took part and trying to see if the pipeline can identify and describe them significantly.

The architecture described is outlined in Figure 4.3. The following subsections provide a more detailed explanation of the methodological approach considered for each part of this work, extending the already presented components/tasks.

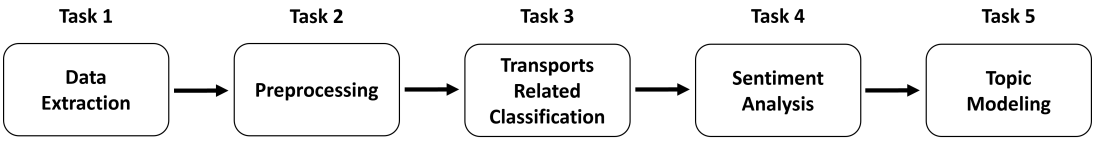

Figure 4.3: Project macro architecture.

4.2.1 Data Extraction and Preprocessing

The data extraction process was not focused on during this work, and consequently, the data preprocessing reflected what had already been done to the data previously. Using the same data also allows a correlation line between works which can help to understand if the newly developed methods represent an improvement. Still, there was a reflection period about how this could be improved or changed for the future. The considerations made for these subjects can be found in Appendix B.

The diagram displayed in Figure 4.4 represents the process used to extract and preprocess the social media content.

Pereira [92] previously extracted the Twitter content used for this work during his dissertation. The decision to use pre-extracted data instead of extracting new one is supported by three main reasons: first, this dataset[92] already has millions of tweets from the three different cities, which is important for studying different traffic situations and making the evaluation as robust as possible; secondly, time restrictions waiting for the extraction would delay the development due to not having content to train and test the models; lastly, during the start of the work there was already the information that eventually the Twitter API was going to suffer changes and the free tier that allowed extraction would disappear, making a new extraction no longer a viable solution due to its cost. These policies change ended up actually happening in the middle of the project.

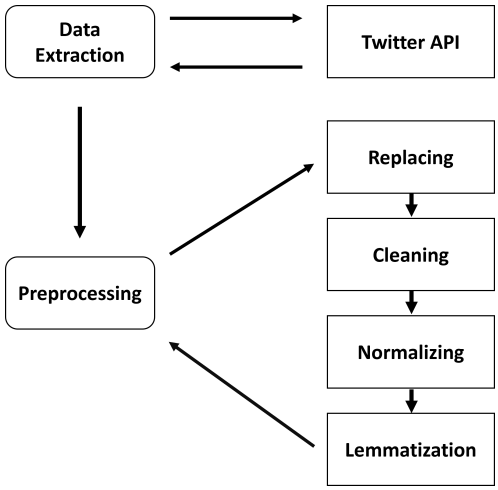

Figure 4.4: Data extraction and preprocessing architecture.

It is important to highlight that due to changes to Twitter policies, its previous API was deprecated in April 2023, and the extraction process described in Pereira's work [92] is no longer replicable. It is still possible to extract content from Twitter using their new API, but now it always has a cost associated, and there are much more limitations.

Regarding the pre-processing, it is important to first detail the steps that were part of the treatment done [92, 79] to the data that is being used in this work. The first step is Replacing, which consists in substituting contractions with their long form and numbers with their text occurrences. This is followed by Cleaning, which eliminates unwanted or defective data from the text, i.e., removing HTML tags, special characters, hashtags, URLs, non-ASCII characters, user mentions, stop words, and punctuation irrelevant to the classification task. Normalizing is the third step, and it aims to standardize the text by converting it to a consistent format, which includes converting all text to lowercase and removing extra white space. The last process is called Lemmatization, which is grouping together the inflected forms of a word (only performed for the verbs). These four steps constitute a relatively common preprocessing approach for most NLP tasks.

4.2.2 Ensemble Transports Related Text Classification

Text classification is a vast subject, and there are already diverse approaches to this problem, either focused on transportation or other subjects. Considering the results from previous works studied during the literature review, an ensemble combining the predictions from two or more other models to make a final prediction is an exciting approach to combining the best of each model/algorithm [18]. Figure 4.5 shows how the text classification process is structured.

After thinking about the most interesting algorithms to be included in the ensemble, the decision was to have three different types of approaches (Figure 4.6): the traditional machine learning algorithms, which are Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF); pre-trained Google Bert models without additional training involved; and lastly, a fine-tuned Google Bert model.

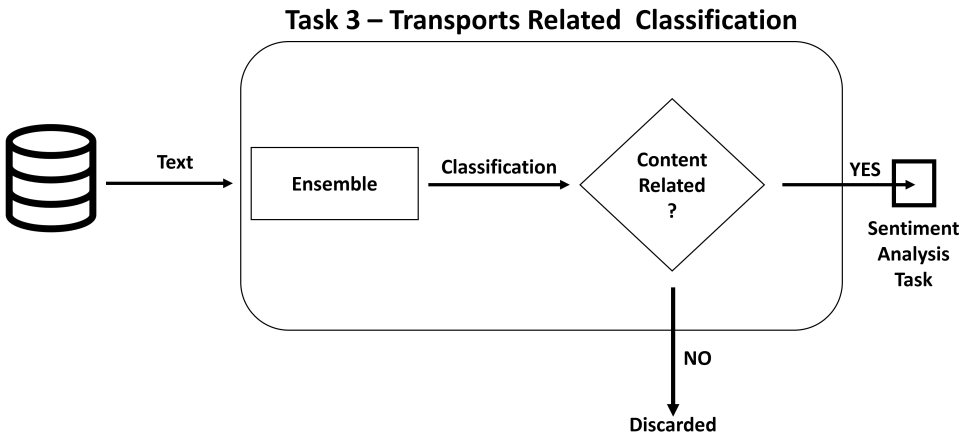

Figure 4.5: Transport-related text classification architecture.

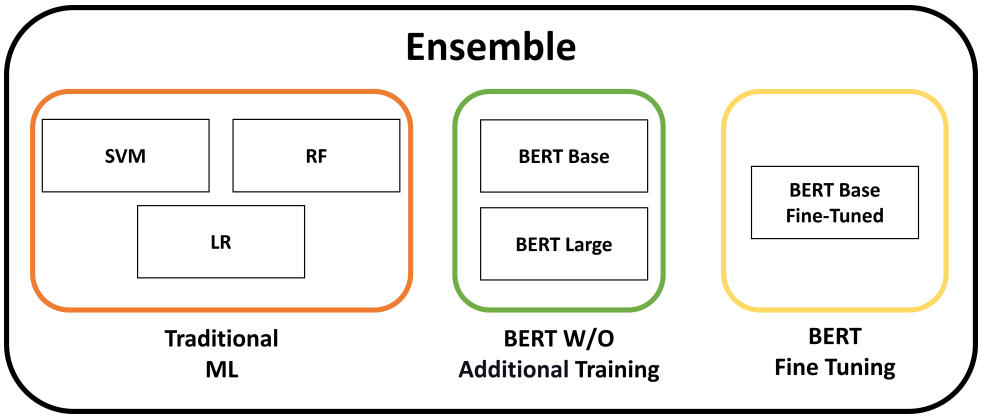

Figure 4.6: Algorithms that constitute the ensemble and their corresponding group.

Given the scope of the project and the chosen algorithms to build the ensemble, a Bagging approach is considered the most appropriate. Boosting would not be a viable solution because some of the ensemble's algorithms do not include a training process. As for Stacking, this could be implemented, but Bagging is a more scalable and faster solution. This approach is more adequate for near real-time situations in which time is crucial, and it is also easier to add, remove, or substitute new algorithms since it will not require a new training process like with Stacking.

The traditional algorithms were used during the dissertations from which the dataset chosen to support this project came, Pereira's work [92]. Although there multiple common algorithms are adequate for this problem, only the three referred in his work were used. Choosing these as the ensemble's common machine learning algorithms makes sense since it allows one to follow his line of work and compare efficiency and efficacy with other solutions.

As for Google Bert Models without training involved, usually, this is not a common approach for these types of problems. Most authors fine-tune the chosen Google Bert model to perform the intended task. However, obtaining the embeddings and working with dictionaries related to each class makes it possible to develop a classification model that does not require additional training of the already pre-trained Google Bert. This solution is not only faster but also requires fewer computational resources while also being simple to expand the scope by increasing the dictionaries and including new subjects related to transports.

Finally, in the fine-tuned Google Bert Models, the last layers of the models are retrained and responsible for classifying the chosen content according to the training done using a specific dataset for the problem in question. This was the last chosen group because it is the common way of using Google Bert and was yet to be explored for transportation text classification problems.

Although GPT-3.5 and GPT-4 are very popular due to the ChatGPT release, these models were not included in the ensemble due to their high cost. Appendix C outlines the calculations for using these models. As this section shows, the cost of using these models would be prohibitive, not only during the learning and training phase but also in the future, since, for each prediction, it would get more expensive. Appendix C also includes the results of manual tests conducted with ChatGPT to evaluate the potential usefulness of the models for addressing the specific problem at hand.

In the next chapter, Chapter 5, the details regarding the implementation for the ensemble are provided in detail, with an accurate description of how each model/algorithm was implemented and optimized.

4.2.3 Sentiment Analysis

As the literature review displayed (Section 3.2.4), authors have many different ways of performing sentiment analysis in the context of social media and transports.

Since not everyone uses the same approach, this can result in different interpretations of the same situation, influenced by the tendencies of each algorithm. Understanding how the most common approaches diverge is the first step to implementing something that can improve this heterogeneity while also giving more accurate sentiment predictions.

This work studied four algorithms for sentiment analysis: VADER, TextBlob, Afinn, and a BERT Base model fine-tuned. These models are popular and therefore used for many projects where sentiment analysis is necessary. All of them can provide a compound polarity, exactly what is desired to build an ensemble. Although some models can be fine-tuned, due to the lack of transportation-related content manually labeled, both for sentiment analysis and text content, for this project, the models used will be either pre-trained or based on lexicon rules without additions to the lexicon.

Having the numeric polarity is crucial to interpreting different situations since, for example, a negative sentiment with a polarity of -5 should probably be related to something much worse than what a negative sentiment with a polarity of -0.5 is related to.

The suggested approach to deal with this is also an ensemble. Work developed on the sentiment analysis topic [99] shows that the algorithms to do this can diverge from each other, particularly for neutral sentiments, in which some prefer negative values close to zero. In contrast, others prefer positive values close to zero. The proposed architecture can be seen in Figure 4.7.

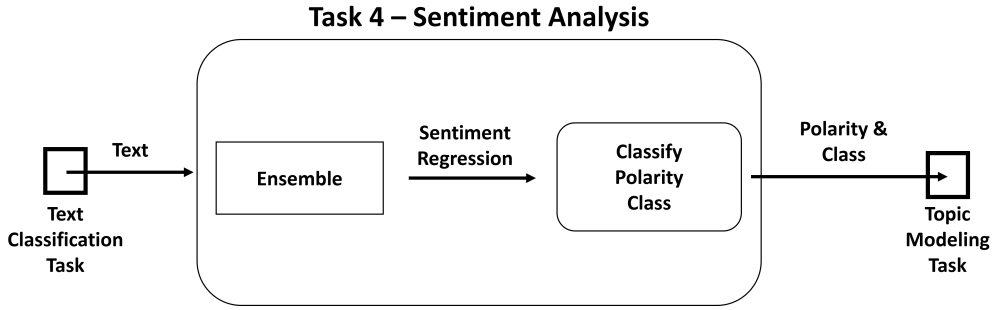

Figure 4.7: Sentiment analysis task architecture.

Considering this architecture, two possible distinct approaches can be implemented to deal with this divergence. One of them is an ensemble that predicts the sentiment for every text and has a final prediction result, which is the average of the polarities given by each model. The other possible approach, because the divergences are not so relevant on the extremes of the polarity, is to choose one algorithm as the main one, preferably the best according to performance metrics, and only run an ensemble for the predictions that are contained in the most common divergence interval. These two approaches' differences are also detailed in Figure 4.8.

Just running the ensemble all the time is the most usual approach, but the time saved by not doing this might be worth more than the increase in performance, especially for near real-time tasks.

The following chapter, Chapter 5, details the implementation of all the individual algorithms and sentiment analysis ensemble solutions described here.

4.2.4 Topic Modeling

The last task used to complement and help improve the evaluation result is Topic Modeling combined with Topic Labeling, done using text generation to summarize and explain the resulting

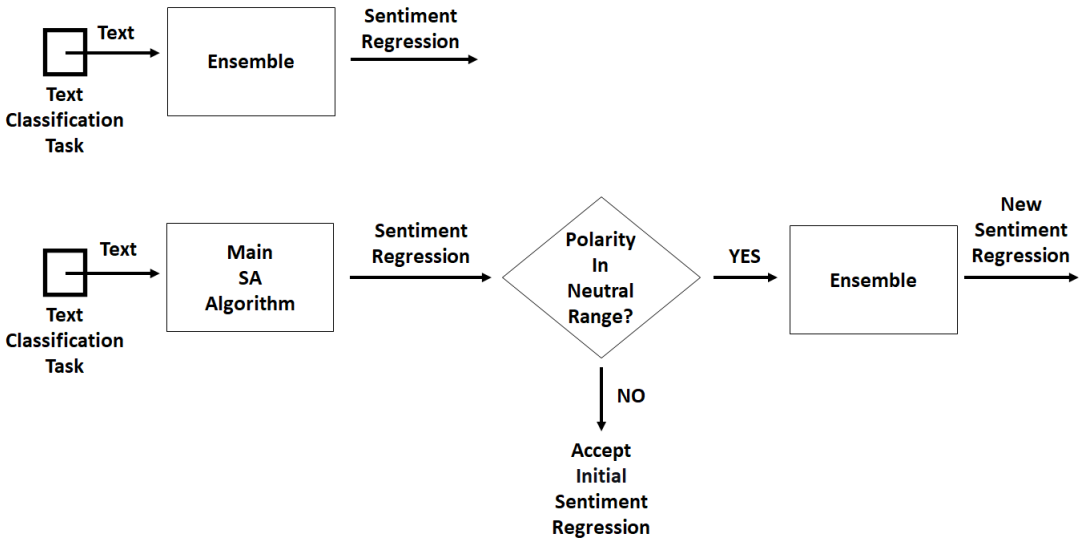

Figure 4.8: Sentiment analysis possible architectures to use an ensemble.

topics using brief sentences instead of just lists of words.

Topic modeling is already part of some of the reviewed works, but most authors consider that just a list of words is enough to understand a specific topic. Although this might be true sometimes, the contrary can also happen. In that case, the list of words becomes irrelevant for the analysis since it will not give any additional information.

In prior research unrelated to transportation [68, 12, 7], authors used different techniques to generate a text that could resume a topic, an approach referred to by most authors as Automatic Topic Labeling. These models used a list of words as input, which is the output provided by most topic modeling available algorithms. With the increased popularity of LLMs in the last year, it is believed that it is worth investigating this subject again and possibly trying to use the multiple LLMs available to improve the results obtained for the previous solutions.

Using prompt engineering, it is possible not only to provide as input the list of words related to the topic but also to give context about this list, e.g., the year or even the location from which the texts are being written. This was something that could not be considered in previous works.

To implement this approach, the chosen technique for Topic Modelling is Latent Dirichlet Allocation (LDA). Because the focus of this study is the labels generated for the topics and not the topics themselves (already explored in the literature), LDA seems a good choice since it is the most common way to do it. Besides the fact that it works fast, the fact it is a proven solution was part of the decision to use it for this final task. As for Text Generation, multiple available LLMs already pre-trained can be used for the intended task. However, most of them are beyond a paywall or require heavy computational resources and extensive time to produce results. Hence, the chosen approach was to try out the most popular models that are made available online for free as a chatbot, which are Open AI ChatGPT [7] and Google Bard [8].

[7] https://chat.openai.com/

[8] https://bard.google.com/?hl=en - Accessed using a VPN due to region limitations

The imagined architecture would be the one displayed in Figure 4.9. The only difference from the actual architecture is that the LLM part was substituted for the chatbots and therefore does not follow an automatized flow.

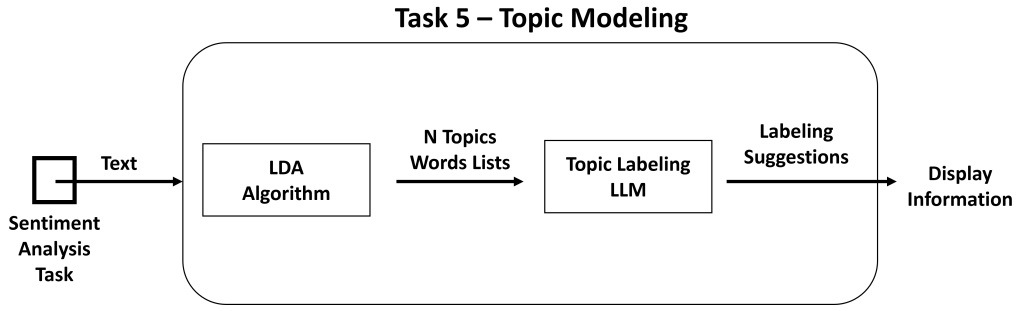

Figure 4.9: Topic modeling task architecture.

Since LLMs are still a relatively new subject, there is a new development almost every week, and big differences exist between the available models. Trying out more than one is important to understand if at least one is viable or if this is not a good application of these models for now.

4.2.5 Performance

Performance evaluation is important to every developed work, so it must be structured and presented clearly.

To compare the performance of the individual models and ensembles implemented both for the transportation-related classifier and sentiment analysis, four different evaluation metrics are considered: accuracy, precision, recall, and the f1-score.

These metrics are the most common ones for text classification tasks [46], and therefore it is believed that they are the most adequate to be used for comparing the performance of these algorithms.

In the following equations, TP means True Positive, TN means True Negative, FP means False Positive, and FT means true negative.

The first metric, Accuracy, gives a comparison between the actual and the predicted class of each data point, and it can be problematic when evaluating an unbalanced dataset as shown in Equation 4.1:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.1}$$

Precision compares true positives and false positives, therefore providing a different look than the one Accuracy gives. Using this metric, it is possible to understand if the model only predicts a class well because it also predicts it in other false cases. It allows an understanding of the proportion of positive cases identified that there were actually correct, as demonstrated by Equation 4.2:

$$Precision = \frac{TP}{TP+FP} \tag{4.2}$$

The Recall uses the same thought process as Precision, but now the focus is on false negatives instead of false positives. It calculates the proportion of the actual positives that were identified correctly, and it can be calculated using the Equation 4.3:

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

The last performance metric used for evaluation is the F1-Score, which combines Precision with Recall. It can be calculated using equation 4.4:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

The ideal values for the four measures are as close to 1 or 100%, in case they are presented using percentages, as possible, which with a properly diverse test dataset, would indicate that the used algorithms generate very few false positives or false negatives.

For the transportation-related classification, the evaluation was conducted with two different experiences, one using a dataset of 1000 classified tweets only from New York City and another using 3000 tweets from three different cities, New York, London, and Melbourne. The second dataset served as a broader evaluation set, simultaneously allowing performance assessment across different geographies.

To ensure reliable results, k-fold cross-validation, with k=5, was used for the machine learning models that required additional training. Cross-validation involves splitting the dataset into multiple subsets, training the model on some subsets, and testing its performance on the remaining subset. It helps estimate the model's accuracy on new, unseen data and is useful for avoiding overfitting, a common problem in machine learning. The most common values for k are 5 and 10, which have been shown to provide a good balance between bias and variance in many cases. Due to the dataset dimension, using $k = 5$ is more than enough to ensure it works as intended. Since the dataset to evaluate this task (text classification), performance is relatively small, cross-validation was considered a good technique to make the results more trustworthy.

Since the duration of each execution was also timed to understand which algorithms were faster or slower, every model was tested using five repetitions in an attempt to remove possible disturbances.

For sentiment analysis, to understand if the ensembles provided results as good as the individual models, both groups were tested using a dataset available online, specifically for tweets SA, which is Sentiment140. Doing this was an important part of evaluating this task because it allows to understand if the ensembles developed were capable of eliminating the dispersion between solutions while still maintaining the level of accuracy that is expected for a good evaluation.

It is important to highlight that using performance metrics was the approach taken to evaluate the classification tasks, transportation-related and sentiment analysis, but not for the topic modeling since it is possible to have multiple correct suggested sentences. For this last task, there can only be a subjective discussion about whether the suggestions are reasonable and relevant or not. The evaluation will be conducted by having multiple persons manually classify the generated

Table 4.4: Computer specifications.

Operating System	Windows 10 Home 64 bits
Processor	11^{th} Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz
Graphics Card	NVIDIA GeForce GTX 1650 with Max-Q design
RAM	16GB
Type of Storage	SSD

texts from 1 to 5 using a Likert Scale and then study which model produces the best labels/topic explanation according to the classifiers' opinions.

After all the tasks are implemented and tested out individually, the final assessment is to apply the evaluation pipeline to real-life cases found in the data presented in Section 4.2.1 and understand if it produces relevant results and for which cases it works best or worse.

This work was conducted and evaluated using a commodity laptop with the characteristics presented in Table 4.4.

4.3 Summary

This section provided an overview of the methodological approach chosen for the project. It began by presenting the problem and the related questions, through which it was possible to understand what needed to be improved using the literature review as a baseline and what could be newly developed. This was followed by explaining the data relevant to the project, which was divided into Twitter content and the New York City events/accidents archive. After the data explanation, the architecture was presented together with each of the individual tasks. This started with an explanation of the extraction and preprocessing phase, followed by the transportation-related text classification, in which the ensemble approaches and each group of algorithms, traditional machine learning methods, Google BERT models without additional training, and fine-tuned BERT models were presented. As for sentiment analysis, each algorithm was explained together with the ensemble approach used to minimize the differences between different approaches. Topic Modeling was the last task detailed, and the focus was both on the LDA algorithm used to generate the topics and the LLMs used to generate the labels/explanations to detail these same topics. Performance evaluation was the final focus of this chapter, and it discussed the different metrics and approaches used to evaluate the developed work correctly. This evaluation consisted of both an individual assessment of each algorithm, using metrics like accuracy or F1-score, and a generalized evaluation of the developed pipeline by comparing the results to real-life cases.

Chapter 5

Implementation

The following chapter provides an overview of how the suggested methodological approach was implemented for each one of the tasks explained in the previous chapter (Chapter 4). It is organized into three sections, each addressing a different task. First (Section 5.1), the implementation of the ensembles for transportation-related content classification is explained, with details about each one of the individual algorithms. This is followed by the sentiment analysis task (Section 5.2), in which there is a comparison between multiple methods and the suggested implementation to try to deal with the discrepancies between them. For the last task, Topic Modeling and Labeling (Section 5.3), there is a brief description of the implementation to identify the topics and details about how LLMs can be used to summarize these topics by generating a proper label when given a list of words and the necessary context about it.

5.1 Ensemble Transports Related Text Classification

The chosen type of ensemble for this problem was a Bagging one. It was implemented using a regular/majority and a weighted voting system.

In the regular voting system, also known as majority voting, the final classification is determined by the class that receives the most votes from the individual classifiers. When the number of classifiers is odd, a tie is impossible. However, if the number is even, a tie can occur, and one possible solution is to choose one of the tied classes randomly. Alternatively, the tied samples can be discarded from the analysis to avoid compromising the reliability of the ensemble. For this project, to avoid the risk of losing important content, it was decided that in case of a tie, the text would be classified as related.

In the weighted voting approach, models have different voting weights that reflect the performance of each model, with usually higher weights being reserved for the models that perform better individually [2]. The sum of the weights should equal one, and the value for each weight should recall the relations between the chosen performance metrics [46]. The final prediction is not necessarily the one with the most votes since the outcome is influenced by the number of votes and their strength or magnitude.

Since the two approaches were considered valid for the problem being studied, implementing both made it possible to understand if one was better than the other and, if the difference existed, how relevant it was.

The following subsections provide details about the implementation of each algorithmic group that is part of the ensemble, followed by a subsection that describes the approach and techniques used to group the models and save time during executions.

5.1.1 Traditional Machine Learning Algorithms

The three chosen algorithms from this group are the Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF). These algorithms do not get as input the text in its natural form but as a combination of embeddings and bag-of-words.

The embeddings were generated using the Gensim library [1], an NLP Python library. The bag-of-words and the classification algorithms were implemented using the scikit-learn [2] library, which is a software machine learning library, also available for free in Python.

Starting with the features, these are generated with models that are made available pre-trained with large amounts of data. For this implementation, the final vector combines the vectors (matrices) generated by a bag-of-words and a bag of embeddings. This mix of features was studied in Pereira's work [92], and he concluded that it gives the best results when compared with both individual approaches. The two algorithms used are countVectorizer and paragraph2vec. Converting the original text to this type of numerical representation is necessary both for training and testing.

Regarding the classification algorithms, SVM finds an optimal hyperplane that separates data points of different classes in the feature space [45]. LR is a statistical algorithm used for binary or multi-class classification [29]. LR can also be used for unsupervised learning tasks but primarily for supervised learning. It models the relationship between the independent variables and the probability of a certain class using the logistic function. RF is an ensemble learning algorithm that combines multiple decision trees to make predictions [17]. Each tree is trained on a random subset of the training data, and the final prediction is made by aggregating the predictions of individual trees. All three models output the value 0 if the content is unrelated to transports and value 1 in case it is.

Each one of these classifiers has hyperparameters that can be changed according to the task. The library used provides a solution to understand the best parameters: cross-validation with different combinations of hyperparameters. After multiple runs with different combinations, in the end, the best combination of parameters is highlighted and used during the training and testing. After making this study, it was concluded that the standard parameters (the ones chosen by omission) were the best for all the models, except for the SVM kernel, for which the *linear* kernel was the best choice.

[1] https://radimrehurek.com/gensim/
[2] https://scikit-learn.org/stable/

5.1.2 Google BERT

The second and third groups of classifiers used in the ensemble both relate to one model, Google BERT (Bidirectional Encoder Representations from Transformers).

BERT was released by Google in 2018, and it uses transformer architecture and bidirectional training to capture the contextual meaning of words. Other models that are context-free models, such as word2vec or GloVe, generate a single-word embedding representation for each word in the vocabulary. Words with multiple meanings tend to lose context when represented like this. The word mouse is a good example of this in the sentences "the mouse is trapped in that cage" and "my computer mouse is not working".

BERT is built after the transformer architecture introduced by Vaswani et al. [112]. Transformers are deep learning models that utilize self-attention mechanisms to capture relationships between words in a sentence. The self-attention mechanism allows each word to attend to all other words in the sentence, enabling the model to understand the context and dependencies between words.

Tokenization is also a big part of the process. Before being used, the input text is broken into smaller chunks or tokens, such as words or subwords. Each token is assigned a unique numerical representation. WordPiece, also developed by Google, is used for tokenization, breaking words into subword units. Taking as an example the word "cooking", it might be tokenized into "cook" and "##ing" (## indicates a subword). This approach allows BERT to handle out-of-vocabulary words by representing them as subwords or combinations of subwords. BERT also processes input sequences by adding special tokens to mark the beginning and end of a sentence and separate pairs of sentences in tasks that involve sentence pairs (e.g., question answering). It includes a [CLS] token at the beginning of the input, which serves as a classifier token, a [SEP] token to separate sentences, and [PAD] to reach the max length expected, as Figure 5.1 shows.

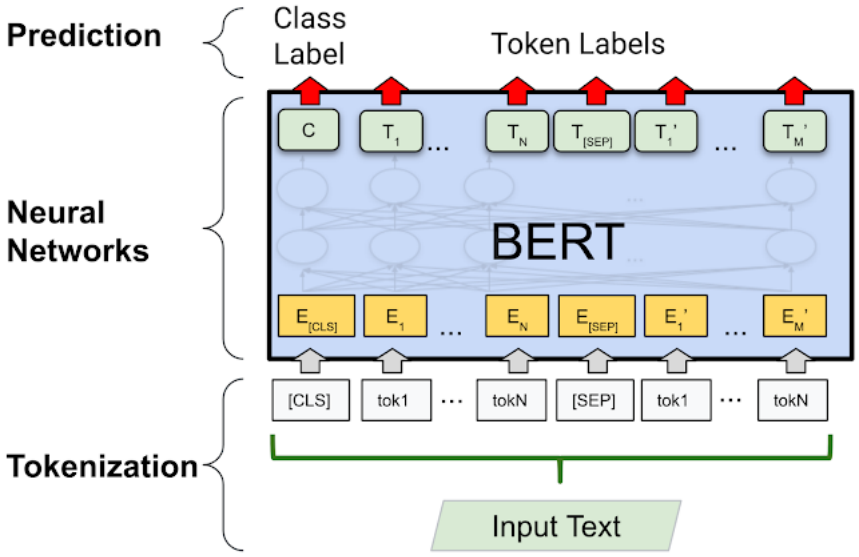

Figure 5.1: Google BERT Tokenization (Adapted from [104]).

Table 5.1: Google BERT early models comparison.

	BERT Models	
	BERT-Base	**BERT-Large**
Transform Block Layers	12	24
Hidden Units	768	1024
Attention Heads	12	16
Parameters	110M	340M
Training Corpus Words Size	800M	3300M

Multiple Google BERT models are currently available, but only two versions were initially available, BERT-Base and BERT-Large. Table 5.1 provides a comparison between both of them. As the name suggests, the Large model is much bigger, with a training corpus size almost four times larger than the Base model.

In 2019, Google released twenty-four smaller models [108] that consume fewer resources and take less time to execute without compromising the performance too much. The smallest model is BERT-Tiny, which has 2 transformer layers and 128 hidden units per layer. Some other examples of these smaller models are BERT-Mini, BERT-Small, and BERT-Medium. There are also some models [61, 65] developed by different companies, such as Facebook and Google AI Language, to reduce the memory footprint of BERT and make it more efficient for deployment on devices with limited resources. Each of these models has unique advantages and may be more suitable for certain use cases depending on the specific requirements of a given task.

BERT models can be trained on any large data set, with some of the most common choices being Wikipedia and BooksCorpus due to their dimensions and content diversity.

Models can also be classified as uncased or cased, depending on how they process the input received. Uncased models will ignore the text case and treat it all as if it was lowercase. Using cased models can be helpful for cases in which the accent and capital letters play an important role in understating the text content.

When it comes to applying BERT to new tasks, the model takes advantage of transfer learning. By leveraging the knowledge learned during pre-training, BERT can quickly adapt to specific tasks with minimal training on task-specific data. This transfer learning approach makes BERT highly efficient and effective in various NLP applications.

One of the typical tasks that Google BERT can be adapted to perform is text classification [41], regularly done by fine-tuning one of the available pre-trained models on a labeled dataset specific to the task.

Although the fine-tuning approach is the most common way of dealing with a task like the one described, it is also possible to follow a different process, using dictionaries of words for each class and making use of the Cosine Similarity, as suggested by Di Pietro [94].

The Cosine Similarity, represented in equation 5.1, can be used to calculate how close two vectors are to each other, with each vector representing a vector of embeddings.

$$\text{Cosine Similarity}(A,B) = cos(\Theta) = \frac{A \cdot B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.1}$$

This approach was already partially explored in the context of transportation [79]. However, in the implementation description, it is possible to understand that no threshold was defined for the average cosine similarity value, which was considered not the best approach after discussion with the author (Pietro [94]) of the article that served as a reference for this solution.

In this new implementation, for each text, the value used for the classification is an average sum of the cosine similarities between the text (a tweet) and each word of the dictionary. This value is normalized to fit in the range $0-1$. Since the point is to identify which tweets are related to transportation and which are not, the implementation consisted in adapting Pietro's multi-class classifier suggestion [94] to work with a binary classification problem. As for dictionaries used, because one of the classes is just contrary to the other (transportation-related/unrelated), it is only necessary to have one single dictionary, in this case, one related to transports. The chosen dictionary is the one considered the best in Murçós [79] work. This subject was already explored and extensively evaluated by comparing three different dictionaries, concluding that the medium with thirty-five words, demonstrated in Table 5.2, is superior.

Table 5.2: Dictionary of transportation-related words.

accident	bus	highway	street	truck
avenue	buses	metro	streets	trucks
bicycle	cab	moto	subway	van
bicycles	car	motorcycle	taxi	vans
bike	cars	motorcycles	taxis	walk
bikes	driver	road	traffic	uber
boulevard	drivers	station	train	underground

After calculating the cosine similarity for a tweet and each dictionary word, if the average Cosine Similarity [60] value of the tweet is higher than 0.5, the value chosen as the threshold, then the tweet is classified as being related to transportation.

The dictionary contains some words in plural and singular forms because they might have different meanings. One of the differences that can exist is that one word can be a verb and the other a noun. As for different concepts, taking the word truck as an example, the word truck means a type of vehicle, but trucks can mean multiple vehicles or a part of a skateboard used to connect the board with the wheels.

Table 5.3 compares the average cosine similarity results for three different tweets using a dictionary with the plural words and one without them. It was still decided to include both forms, but conducting a larger study and deeply understanding this subject is important.

As explained in this section, multiple pre-trained models are available for free. Each model has strong and weak points that must be considered for each problem.

Table 5.3: Comparison for three different sentences between the similarity result obtained for a dictionary that includes plural words and one that does not.

	Similarity Results	
	Dictionary With Plural	**Dictionary Without Plural**
"Another accident involving bikes on fith avenue"	0.57	0.58
"Company trucks are delayed again due to the snow"	0.55	0.54
"My skateboard needs new trucks ASAP or I will crash"	0.51	0.51

The first chosen model was the Google BERT Base, which was already tested on this context [79], producing satisfactory results. One additional model was considered for individual evaluation and possible integration into the final ensemble, Google BERT Large, which is slower than the Base version but bigger and usually more accurate.

Considering the problem, the sentence case was considered irrelevant, so all the tweets were converted into lowercase during the pre-processing. This makes using BERT case models irrelevant, and because of that, only uncased models were tested. Capital letters can transmit emotions, making analyzing them usually more relevant for domains like sentiment analysis, as shown in Chan and Fyshe's work [21].

Hugging Face has both models, BERT-Base [3] and BERT-Large [4], already pre-trained and available for free for any person to use [34], so this was the chosen platform to import these models.

Although only these two were used, different models could also have been considered during implementation by changing the desired import. This would only require a new download because the rest of the process is the same, including tokenization. The text used for the import reflects the intended model name and architecture. It is important to highlight that this ease of adaptation is only true if the model only varies in the number of parameters. Otherwise, a change like going from an uncased to a cased model must be considered in prior phases, such as the data preprocessing.

The second way of using BERT for text classification is to fine-tune the pre-trained model on a labeled dataset specific to the text classification task. The fine-tuning process involves retraining the last few layers of the model on the labeled data. This data consists of text documents and their corresponding category. The categories include topics, emotions, sentiments, or predefined labels.

During the fine-tuning process, the BERT model learns to recognize the patterns and relationships between the words in the text documents and their respective categories. It can then classify new text documents into the categories learned during training.

Usually, using BERT for text classification can be divided into four tasks:

[3] https://huggingface.co/bert-base-uncased
[4] https://huggingface.co/bert-large-uncased

- Pre-processing: It consists of cleaning (deleting unwanted or defective data), normalizing the text, tokenizing it into words or subwords, and converting the text into numerical features that can be fed into the BERT model.
- Fine-tuning: One of the available pre-trained BERT models can be fine-tuned specifically to the intended text classification task by retraining the last few layers of the model on the labeled dataset. This involves feeding the preprocessed text data into the model and updating the model's weights based on the error between the predicted and actual categories.
- Evaluation: The fine-tuned BERT model is then evaluated on a separate validation set to measure its performance. This can be done using various metrics, but accuracy, precision, recall, and F1 score are typically the most common.
- Inference: After training and evaluating the model, if the results are satisfying, it can be used to classify new text documents into predefined classes.

For this project, the chosen model for fine-tuning was BERT Base. The Base model was selected based on several factors, including computational resources, training time constraints, dataset size, potential overfitting, and complexity requirements. BERT Large has a significantly larger architecture that demands more computational resources for training and inference. Its fine-tuning process requires extensive computational power, including GPU memory and processing capabilities. Given the limited computational resource availability, choosing BERT Base for fine-tuning is more practical. Additionally, considering the project timeline, training the model within a reasonable time frame is crucial to stay on track. Therefore, selecting BERT Base is a pragmatic decision. Moreover, the performance of larger models tends to improve when there is a substantial amount of training data. However, since the manually classified dataset is relatively limited in this case, fine-tuning BERT Large may not yield significant performance improvements over BERT Base. Despite its higher capacity, the larger model can also be more susceptible to overfitting, especially when working with limited data. By using BERT Base, which is smaller, the risk of overfitting can be mitigated, and generalization can be promoted, especially if the task at hand does not necessitate the additional complexity offered by BERT Large.

As for parameters, the batch size is 16, the learning rate is $5 * e^{-5}$, the number of epochs is 2, and the validation set is 20% of the training dataset. These are the standard recommendations presented in the original BERT paper [34]. Since they produced satisfactory results, there was no need to increase the values because it would take much longer to compute without necessarily many chances of improvement.

5.1.3 Ensemble Approaches

After explaining the individual algorithms, this subsection gives an overview of the different ensemble approaches implemented and how their execution is organized. Detailing this part is important because the implementation chosen allows for saving time during the executions of the ensemble, making it more viable for possible shorter-term tasks.

Two different ensemble approaches were implemented, majority voting and weighted voting. Since only the result calculation is different, the rest of the process is detailed as if it was one single approach.

The purpose solutions are parallel ensembles which means that models can run separately. When they all finish running, the results are grouped to choose the final prediction. This parallelism makes using threads for different algorithms possible, which could not be done for a sequential solution.

Starting with the traditional machine learning group, it has three different models, but all three use the same features. To save time, the features are only generated once and then used to feed the three models, each one running independently. It would also be possible to use the embeddings from the BERT models here, but since it was also important to compare the different approaches, it was decided not to do this.

For BERT models without additional training, there is no additional strategy besides running the Base and the Large model in different threads.

For the BERT Base model fine-tuned after doing the additional training, which takes a relatively long time to achieve, this model is saved so it can be then loaded for more executions. This is the norm since always needing to train the model would be an unnecessary use of time and resources while also making it impossible to use it for short-term tasks.

As for the weights, as the results chapter will show, after calculating the performance metrics for the individual solutions, it was possible to understand one algorithm was far better, so this one got 49% voting powers, and the rest of the percentage was divided across the models. This gives the rest of the algorithms a chance to outvote the best models if all the remaining agree.

5.2 Sentiment Analysis

For sentiment analysis, an ensemble that integrates multiple algorithms tested by other authors was the choice for implementation. Bagging was also the type of ensemble that seemed more adequate.

Unlike the previous task, since the objective for this one was to reduce the polarity difference between models, this is a regression problem. However, after calculating the polarity, a sentiment label is also attributed, which is secondary.

Considering this is a regression problem, the final prediction value from the ensemble is the result of the average of the individual predictions. A weighted average approach was also implemented based on the algorithm's performance. Because models use different scales to measure the sentiment, it is necessary to normalize each prediction to fit the desired interval, the scale previously explained.

The ensemble polarity limits chosen are between -1 and 1, with values between -0.1 and 0.1 being considered neutral. This scale represents the smallest scale found in one of the algorithms and is relatively common among some sentiment analysis algorithms. It is large enough that the polarity value becomes relevant for looking at and studying possible polarity variations. With an

even smaller scale, it would be harder to distinguish between close values. Working with a bigger scale would be a problem for normalization since it could generate false predictions because it would require conversions from smaller to larger gaps.

As mentioned in Section 4.2.3, the algorithms used for this implementation are VADER, TextBlob, Afinn, and BERT. Each one of them is briefly explained in the following subsections, except for BERT, which was already explained in the previous task. The chosen models do not require additional training or parameter changes/decisions.

After each prediction, since having a label might be useful for certain analyses, besides generating the numerical polarity, there is also a label assigned, which can be Neutral if the polarity value is between -0.1 and 0.1, Negative if it is below -0.1 and Positive if it higher than 0.1.

For the ensemble approach that only runs for close to neutral cases, if the prediction of what is considered the best model gives a value with polarity inside the range 0.2 to -0.2, then the remaining algorithms are executed. The new average value is considered the final prediction.

5.2.1 VADER

Starting with the Valence Aware Dictionary and sEntiment Reasoner algorithm [49], commonly known as VADER, was developed by researchers at the Georgia Institute of Technology, and it offers a fast and accurate approach to sentiment analysis that considers the nuances and complexities of human language.

VADER is specifically designed to handle the challenges of sentiment analysis in social media texts, where expressions are often informal, ungrammatical, and heavily influenced by context. Unlike traditional sentiment analysis techniques that rely on pre-constructed lexicons, it utilizes a combination of linguistic rules and a pre-trained model to provide sentiment scores at the sentence and document levels [49].

At the core of VADER is a sentiment lexicon, which contains a vast collection of lexical features such as words, phrases, and their associated sentiment intensity scores. Each entry in the lexicon is labeled with positive or negative sentiment and a scalar intensity score ranging from -4 (extremely negative) to +4 (extremely positive). The lexicon also accounts for lexical modifiers, booster words, and negations that can influence sentiment polarity.

VADER tokenizes the input into individual words and punctuation marks to determine the sentiment of a given text. It then checks for features in the sentiment lexicon and applies grammatical and syntactical rules to handle phrases and negations appropriately. VADER uses a combination of heuristics, such as capitalization, punctuation, and degree modifiers, to enhance the accuracy of sentiment scoring.

It is open-source, which makes it ideal for this project and part of the NLTK library. Its ability to handle noisy and informal text and its nuanced sentiment scoring makes it a valuable tool for businesses, researchers, and analysts seeking to understand and analyze sentiment in large volumes of text data.

5.2.2 TextBlob

TextBlob [5] is a popular Python library for NLP tasks, including text processing, part-of-speech tagging, noun phrase extraction, and sentiment analysis. It offers a simple and intuitive interface, making it widely used for various NLP applications, such as text classification, sentiment analysis, and language translation.

It is built upon the Natural Language Toolkit (NLTK) library, which provides a robust linguistic data and algorithms set. TextBlob combines the power of NLTK with a simplified API, making it accessible even to users with limited NLP expertise.

One of the key features of TextBlob is its sentiment analysis capability. A pre-trained machine learning model can analyze a given text's sentiment polarity (positive, negative, or neutral). This allows users to gain insights into the emotional tone of textual data, making it useful for tasks such as opinion mining, customer feedback analysis, and social media monitoring.

TextBlob's sentiment analysis leverages a trained Naive Bayes classifier that assigns sentiment labels based on a labeled dataset. The algorithm uses a combination of textual features, including words, word frequencies, and word positions, to make predictions about the sentiment of a given text. The classifier is trained on large corpora of annotated data, enabling it to generalize well to unseen texts, making it also a good choice for the project.

To perform sentiment analysis with TextBlob, the input text is first tokenized into individual words or sentences. Each word or sentence is then assigned a sentiment polarity score ranging from -1 (highly negative) to +1 (highly positive). Additionally, TextBlob provides a subjectivity score ranging from 0 (objective) to 1 (subjective), which indicates the degree of opinion or factually expressed in the text, that was disregarded considering the objectives.

5.2.3 Afinn

The Afinn [81] algorithm is a popular sentiment analysis tool used for determining the sentiment polarity of textual data. It is designed to provide a simple and efficient approach to sentiment analysis, making it widely used in various applications, including social media monitoring, customer feedback analysis, and market research.

It relies on a sentiment lexicon consisting of a collection of words, phrases, or terms and their associated sentiment scores. Each entry in the lexicon is assigned a polarity label indicating whether it conveys positive, negative, or neutral sentiment. The sentiment scores typically range from -5 to +5, with negative values representing negative sentiment, positive values representing positive sentiment, and zero indicating a neutral sentiment.

Unlike more complex machine learning-based approaches, Afinn uses a straightforward lookup mechanism. Given a text, the algorithm matches the words or phrases in the lexicon to the words present in the text. It then aggregates the sentiment scores of the matched words to calculate an overall sentiment polarity score for the text.

[5] https://textblob.readthedocs.io/en/dev/

The lexicon used by the Afinn algorithm is typically created through a combination of manual annotation and automated techniques. Domain experts assign sentiment labels and scores to words based on their semantic meanings and associations. This process allows the algorithm to capture the sentiment conveyed by individual words or phrases.

In addition to individual word matching, the algorithm can consider context and language patterns to enhance sentiment analysis. It considers the surrounding words, grammatical structure, and negations within the text to refine the sentiment scoring. This contextual analysis helps to handle cases where the sentiment of a phrase or sentence cannot be determined solely by the sentiment of its words.

The simplicity and speed of the Afinn algorithm make it an attractive choice for sentiment analysis tasks, especially in scenarios where real-time processing and resource efficiency are crucial. While it may lack the sophistication of more advanced machine learning models, it provides a practical and effective solution for basic sentiment analysis requirements. It is another good choice for near real-time tasks, just like the one studied during this work.

5.3 Topic Modeling and Labeling

This last task is divided into two parts, first the identification of topics and then the labeling of each one of the topics.

Starting with topic modeling this was implemented using one of the most common approaches, which is LDA. It is a probabilistic generative model that assumes each document is a mixture of various topics, and each topic is a distribution of words. It aims to uncover these latent topics by analyzing text word patterns. It works by assigning a probability distribution to each word in a document, representing the likelihood of that word belonging to each topic. Through an iterative process, it then determines the topic composition for each document and the word distribution for each topic.

To train an LDA model, it is necessary to start by deciding the number of topics in advance, the algorithm randomly assigns a topic to each word in each text, it then iterates over each word and its assigned topic, updating the topic assignments based on the current topic assignments of other words in the text and the word distributions of topics. It repeats the previous step until the model converges.

Once the LDA model has been trained, it can be used to infer the topic distribution of new, unseen documents. This is done by calculating the probability of each topic given the words in the document.

The Scikit library mentioned in other tasks was also the one chosen to implement this, as for the number of topics, it is difficult to predict how many they can be since the final application for this is social media content, not a closed source of data. Because the objective is to mainly deal with near real-time tasks and content already filtered to be related to transportation, it was assumed that the number of topics talked about would be relatively low, so five topics was the final number decided. The hyperparameters were not changed from the standard solution available.

After generating the topics, each one of the lists is passed on to the text generation models chosen, in this case, the chatbots ChatGPT and Google Bard. Each list is given with proper contexts, like the location of the texts and the date they are from. This is done using prompt engineering, which consists in asking the bot to generate a label that can be a text or a single phrase and that describes what each list (topic) means. Besides the information about the data, the prompt given to the chatbot also contains an example of what is pretended, in this case, the example is a list of words and a label considered adequate.

It would have been more interesting to use LLMs automatically so that the pipeline produced could be fully autonomous. Unfortunately, there are two different problems with this, the cost and the necessary resources. Regarding the cost, some LLMs currently have APIs available that could be used to generate these labels, but unfortunately, they have a cost for each request done, which would scale fast during the test phase. As for the resources, although every day new models are published and explained in detail, adapting them to become text generators plus training them requires powerful resources and a lot of time to achieve good results. Trying to perform this process would mean much more development time and access to different hardware.

By using the available and very popular chatbots, it is possible to do a proof of concept, and if it shows potential, then try to improve this final task with better and more autonomous tools.

5.4 Summary

This section outlined how the suggested methodological approach was implemented to address the identified problem. It began by detailing the implementation of the transports text-related classifier in which it was possible to understand how each algorithm was implemented and what was considered to save time while running the ensemble, like the use of threads, for example. The sentiment analysis implementation followed this, also suggested as an ensemble, and here, each of the three models' details was explained together with the solution to combine the different predictions even though they come in different scales. The last section was about topic modeling and labeling, and here the approach chosen to generate the topics was presented together with a possible solution to generate labels which is a proof of concept supported by the current most popular chatbots.

Chapter 6

Results and Discussion

This chapter provides the results obtained for the implementation described in the Implementation Chapter 5. It is organized into six sections.

The first three sections provide evaluation details, mainly related to performance metrics, for the TC (Section 6.1), SA (Section 6.2), and TM (Section 6.3) tasks. This is followed by a section that provides a reflection on the results obtained and what they mean for the identified problem while also talking about real-life cases and why they were not part of the final assessment (Section 6.4). The chapter concludes with a concise summary of each section, providing the main results and thoughts.

6.1 Transport Related Classification

The first task accessed was the transport-related text classification. First, the results for the individual solutions are presented using the performance metrics chosen and detailed in the Methodological Approach section. After talking about the individual models, the results for the ensembles developed are then described, followed by a comparison between both ensembles and the best individual solutions.

6.1.1 Individual Results

Starting with the individual solutions, Table 6.1 summarizes all the performance metrics assessed for each group of algorithms used in the ensemble implementations. The first three columns represent the traditional machine learning algorithms, followed by the two BERT solutions without additional training, and lastly, the BERT-Base fined tuned.

Experience 1 represents the tests run using a dataset with only data from New York City, and *Experience 2* represents the tests that used a dataset with content from the three available cities, New York, Melbourne, and London. Besides the performance metrics, the execution times both for training and testing are also displayed so that it is possible to understand the time frame that each algorithm is more adequate to work with.

Table 6.1: Performance metrics comparison between the different individual approaches used for transports-related text classification.

	SVM	LR	RF	BERT-base	BERT-large	BERT fine-tuned
Experience 1						
Execution Time (*s*):						
- Training time	0.390±0.008	**0.390±0.005**	0.628±0.030	-	-	*6184.476±54.345*
- Prediction time	0.021±0.001	**0.019±0.003**	0.027±0.001	40.641±4.019	91.585±0.614	*761.912 ±5.637*
Performance (%):						
- Accuracy	67.0±4	67.1±3	66.8±5	67.0	*54.0*	**94.5±3**
- Precision	66.5±3	66.9±2	67.4±6	70.0	*66.0*	**93.4±2**
- Recall	68.4±6	67.4±6	64.8±9	67.0	*54.0*	**95.6±4**
- F1-Score	67.4±4	67.1±3	66.0±7	60.6	*44.0*	**95.0±1**
Experience 2						
Execution Time (*s*):						
- Training time	1.039±0.018	**0.968±0.017**	1.846±0.025	-	-	18377.758±35.345
- Prediction time	0.096±0.005	**0.063±0.006**	0.071±0.007	108.365±0.457	266.906±2.665	2293.379±4.857
Performance (%):						
- Accuracy	66.3±3	65.4±2	67.9±2	67.0	*55.0*	**96.2 ±1**
- Precision	*64.2±3*	64.7±2	67.9±2	69.0	67.0	**94.5 ±2**
- Recall	73.7±2	68.2±4	68.1±3	67.0	*55.0*	**95.6 ±1**
- F1-Score	68.6±2	66.3±2	67.9±2	65.0	*44.0*	**94.7 ±2**

Symbol ±: Represents the standard deviation considering the k-fold cross validation with k=5
Bold values: Represents the entry with the best result for each row, which is the highest for performance metrics and the lowest for executions times
Italic values: Represents the entry with the worst result for each row, which is the lowest for performance metrics and the highest for executions times

Starting by analyzing the execution times, the difference from Experience 1 to Experience 2 is also close to being directly proportional to the sample size. Inside each group, for the common methods, there is no significant difference between the three, but for the training time, the RF is two times slower than the other two. For the BERT models without training, the Base variant takes less than half the time to predict than the Large one, which is expected since the second has a much bigger architecture.

Comparing the groups, there is a clear difference between the groups both for training and testing. For training, there are only two groups that can be compared. The traditional machine learning algorithms and the BERT-Base fine-tuned. Here for both experiences, the traditional methods are clearly faster, with the slowest traditional method, RF, still being almost 1000x faster, however, with better GPUs, it is believed that this execution time could be heavily reduced to a value that would not make the difference so significant. The same thing can be said for the training process, now also considering the BERT methods without additional training, which are faster than the fine-tuned model but still much slower than the common algorithms.

As for performance, although four metrics are presented, the metric chosen to compare algorithms is the F1-Score. Starting by comparing the three choices for the common methods, between SVM and LR there is no difference, and between them and the RF there is not a difference significant enough to draw a conclusion from and decide if it is possible to conclude which one is better mathematically. As for the models without training, the BERT-Large is by a big margin worse than the Base version, which could be related to the dimension of the model. Lastly, now comparing all the models as a whole, the fine-tuned BERT is substantially better than any of the remaining solutions and therefore is the best individual solution, considering that the prediction time is not

a problem. Between experiences, there seems not to exist a significant difference between one experience and the other, which can be a product of the relatively small increase from one dataset to another, which only has 2000 more tweets.

6.1.2 Ensembles Results

Regarding the ensemble solution, Table 6.2 compares both ensemble implementations for the experience with all the cities since the objective is to have the best ensemble possible free of context. Since the models were previously trained and there is no additional training, it only shows the average prediction time.

Table 6.2: Performance metrics comparison between the two ensemble approaches defined, majority and weighted voting used for transports-related text classification.

	Majority Ensemble	**Weighted Ensemble**
Execution Time (*s*):		
- Prediction time	**760.25**	765.12
Performance (%):		
- Accuracy	94.3	**96.4**
- Precision	93.7	**95.3**
- Recall	92.2	**95.7**
- F1-Score	93.1	**95.5**

Bold values: Represents the entry with the best result for each row, which is the highest for performance metrics and the lowest for executions times

As it is possible to see, the prediction time is very close to time values previously measured for the fine-tuned BERT. Since this is the slowest individual model, it becomes the bottleneck for the ensemble.

As for the performance metrics, the values are not much different both when compared to the best individual results and each other. Regarding the comparison with the best individual results, the fine-tuned version already had really good results, so it would not be easy to improve this significantly. As for the comparison between both ensembles, as expected, the result for the weighted was slightly better, and this can be from the fact that the best model only needs another model to agree with his prediction to win, hence it is harder to be overthrown by slightly worse models.

Lastly, Table 6.3 shows a confusion matrix with different tweets. For each tweet, it is possible to see the actual class it belongs to, which can be positive if it is related to transports and negative if it is not, and the class predicted according to the ensemble. The four tweets used here got classified the same way for both ensembles.

It is possible to understand that false predictions are usually associated with tweets containing words that can have different meanings depending on the context used.

Table 6.3: Examples of text classification of transport and non-transport messages using the confusion matrix classification.

		Predict class	
		Positive	Negative
Actual class	Positive	*True-Positive* "Stuck in traffic again on my way to the airport. Will probably miss my flight. So frustrating!"	*False-Negative* "Train at 5 pm. Ready to embrace challenges and push from your body's limits?"
	Negative	*False-Positive* "When roads intersect with fate, resilience emerges to pave the way forward."	*True-Negative* "Finally got a promotion at work! Celebrating tonight with my friends. Life is good!"

6.2 Sentiment Analysis

Now for sentiment analysis, first, it is important to see the difference in polarities when using different algorithms, so Figure 6.1 demonstrates the polarities predicted by VADER, TextBlob, Afinn, and a BERT model fine-tuned for sentiment analysis, using a dataset of 1500 transports-related tweets.

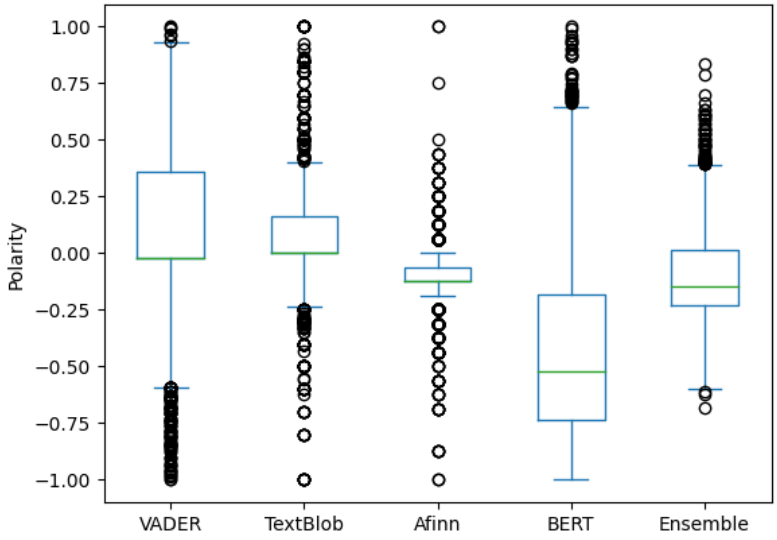

Figure 6.1: Sentiment analysis polarity distribution for VADER, TextBlob, Afinn, BERT and the Ensemble.

As it is possible to see, the different algorithms predict very different polarities. This can completely change the perspective of the person that is interpreting these results. For example, when looking at this dataset through VADER, it is valid to assume that there are no traffic problems since the polarity is majorly neutral. However, by looking at the BERT results, it is much more

plausible to assume there is something atypical and probably bad happening since the sentiment is majorly negative.

The same Figure 6.1 also demonstrates the polarities generated using the ensemble, giving the average value of the predictions generated by the four algorithms.

Lastly, it is important to understand if the results from the ensemble can be considered satisfactory, and this was done using a known dataset, Sentiment140 [40], that is available online [1] and contains tweets and their sentiment class labeled. Unfortunately, the dataset used does not contain the neutral class, so the ensemble had to be evaluated as a binary problem that outputs a prediction that is either positive ($>= 0$) or negative (< 0).

Although the tweets used for this validation are not directly related to transports, they are still tweets, and how sentiments are expressed should not vary with topics. Besides this, when testing a classification methodology, using credible and available sources is important so that other persons can also recreate and validate the work. Sentiment classification can also be something very subjective, and reputable data sets, besides having a dimension bigger than any data set that could be created during this dissertation, also have the advantage of being validated by multiple users.

Table 6.4 demonstrates the performance results achieved with the ensemble for a dataset sample chosen for evaluation. The sample is stratified to include the same percentage of each of the two possible classes, Positive, and Negative, with a total of three thousand tweets. The table also compares the ensemble's results and each algorithm used to build it (VADER, TextBlob, Afinn, BERT).

Table 6.4: Sentiment analysis performance metrics for a sample of the dataset Sentiment140.

	Accuracy	**Positive**			**Negative**			**F1-Macro**
		P	**R**	**F1**	**P**	**R**	**F1**	
VADER	65.3	66.5	61.7	64.0	64.3	69.0	66.6	65.3
TextBlob	61.0	57.0	89.1	69.5	75.1	32.8	45.7	57.6
Afinn	62.9	73.9	38.9	50.9	58.5	86.3	69.7	60.3
BERT	52.0	55.1	21.5	31.0	51.2	82.5	63.5	47.1
Ensemble	65.2	71.6	50.4	59.2	61.7	80.0	69.7	64.4

As it is possible to see, the performance results obtained for the ensemble are satisfactory when compared with the individual values since the results were not compromised by doing this implementation. However, 65% is still relatively low, so it would be important to obtain better results to estimate the users' feelings better. Unfortunately, the analysis shows the problem is not with the ensemble but with the currently available models that usually work best for long texts.

Now that the results of the normal ensemble were compared with the results of the individual models, it is also important to analyze the differences between a standard average ensemble and a weighted average one. Since negative tweets are expected to have more relevancy than positive ones in the context of traffic, the weights were distributed considering the performance of the

[1] http://help.sentiment140.com/for-students

algorithms for the negative class. TextBlob got 40%, VADER got 30%, Afinn 20%, and BERT, which is the closest to 50% precision (the lowest), got only 10%. The results are displayed in Table 6.5.

Table 6.5: Sentiment analysis performance metrics for a normal average ensemble (the base solution) and a weighted average ensemble.

	Accuracy	Positive			Negative			F1-Macro
		P	R	F1	P	R	F1	
Base Ensemble	65.3	66.5	61.7	64.0	64.3	69.0	66.6	65.3
Weighted Ensemble	66.1	68.1	60.5	64.1	64.5	71.7	67.9	66.0

Although the results for the weighted ensemble are slightly better than those for the base ensemble, the difference is extremely small. However, making this analysis was meaningful because the difference still exists, and it may become more significant with more weight optimization or different models.

The final comparison for this sentiment analysis task is between the base ensemble and the approach that only uses the ensemble for close-to-neutral predictions. In this comparison, besides the performance metrics is also important to know the average time to analyze a tweet so that it is possible to understand if the trade-off between accuracy and time saved is worth it or not. The algorithm chosen to be the main one is the one that got the biggest weight for the weighted average, TextBlob, for the same reason. Table 6.6 exhibits the differences between the two approaches.

Table 6.6: Sentiment analysis performance metrics for the ensemble approach that is always executed and the ensemble approach that only runs for close to neutral cases.

	Accuracy	Positive			Negative			F1-Macro	Average Time
		P	R	F1	P	R	F1		
Always Ensemble	65.3	66.5	61.7	64.0	64.3	69.0	66.6	65.3	63.3
Ensemble Neutral Cases	61.9	57.8	88.6	69.9	75.5	35.2	48.0	59.0	9.1

Although the ensemble that runs for every tweet has better accuracy, the prediction time is also much higher when compared with the ensemble that only runs for close to neutral cases. Depending on the task term the ensemble is being used for, it might be worth trading the 5% accuracy for a solution almost six times faster.

Lastly, Table 6.7 shows four examples of tweets classified according to their sentiment, also using a confusion matrix.

Table 6.7: Examples of sentiment classification of transport-related tweets using the confusion matrix classification.

		Predict class	
		Positive	Negative
Actual class	Positive	*True-Positive* "Loving the efficient public transportation options. Just hop on the subway or catch a bus!"	*False-Negative* "Where is the traffic congestion today?"
	Negative	*False-Positive* "I've always wanted to experience a slow-motion race. NY is perfect for that!"	*True-Negative* "Stuck in endless traffic jams."

6.3 Topic Modeling & Labeling

Lastly, for Topic Modeling, it was also necessary to use a set of data to extract topics and generate the labels for them. Since topic modeling is an unsupervised task, there are almost no viable and diverse datasets, especially related to transports. Once again, creating a dataset big and diverse enough to evaluate this task in a relevant manner was not a viable option since it would consume a lot of time. This time would be spent first on the research of tweets and then grouping them to cover multiple and various topics. So what was done was five different word lists were created, the first two directly related to transports and the other three that could be related depending on the context but could also be misleading topics.

By doing this, the first part, the topic modeling, ended up not being tested but since this is such an old approach and already explored in the context of both Twitter and transports, there would not be an additional contribution to the research area, so it can be dismissed. This algorithm also has been tested and produced good results for multiple well-renowned data sets. Therefore there is no point in remaking the same evaluations already done using important benchmarks.

Table 6.8 shows the list of words used for each topic. After creating the lists of words, which simulated an LDA output, each list was given as input to the LLMs together with an adequate prompt. The outputs generated by these models in response to the requested labels are also displayed in Table 6.8, which shows the suggestions of each model for each one of the lists.

The lists of words were relatively simple and served just as a starting point to study topic labeling, still, it is interesting to notice that although both models were given the freedom to choose a larger or smaller label, while ChatGPT preferred to suggest a sentence, Bard chose a two/three-word label for most topics. Another point in which they diverge is that ChatGPT only suggested one label for each topic, but Bard suggested multiple for some topics.

When both models were asked if they would choose different labels if they knew all the topics were directly related to traffic, ChatGPT only decided to add "traffic problems because" to the

beginning of the last three topic labels. Bard, in contrast, changed the last three labels to: "Traffic conditions during fashion show", "Traffic conditions during presidential inauguration" and "Traffic conditions during NFL game".

For each label generation, the two models always specified in the end that these were very broad suggestions that could be wrong or taken out of context.

Although the evaluation can not be as rigorous as the ones made for the previous tasks, it is still necessary to understand if the labels generated are good enough for a user to understand a topic. To do this, four different users classified each label from each algorithm using a Likert scale from 1 to 5, with 1 meaning "Label does not in any way explain the topic in a useful manner" and 5 meaning "Label allows to understand what this topic is about completely". The classifications attributed by each user are organized in Table 6.9.

Table 6.8: List of words that characterize each topic found in the dataset using the LDA topic modeling algorithm, together with a comparison of the labels generated by OpenAI ChatGPT and Google Bard.

	Words list	**OpenAI ChatGPT**	**Google Bard**
Topic 1	Snow, accident, dead, cars, highway	- Fatal snow-related car accidents on New York highways	- Snowstorm-related traffic fatality - Deadly car accident on snowy highway
Topic 2	transit, game, sunny, traffic, waste	- Delays in transit due to traffic while travelling to a sunny game day	- Gameday traffic - Transit delays - Sunny day commute
Topic 3	fashion, prada, tuesday, influencers, clothes	- Prada fashion show on tuesday	- Fashion trends for Tuesday
Topic 4	president, election, biden, trump, debate	- Presidential debate involving Joe Biden and Donald Trump	- President election discussion
Topic 5	giants, vikings, nfc, jones, kurk	- NFL game between the Giants and the Vikings	- NFL game preview

Table 6.9: Grades assigned to each label generated for the topics identified.

	User 1		**User 2**		**User 3**		**User 4**	
	GPT	Bard	GPT	Bard	GPT	Bard	GPT	Bard
Topic 1	4	3	5	3	4	4	3	5
Topic 2	5	3	4	3	4	3	5	4
Topic 3	2	1	1	1	3	1	2	2
Topic 4	3	3	5	4	4	3	3	4
Topic 5	2	2	3	2	2	1	1	3

In Figure 6.2, it is possible to see the difference between the average result obtained by ChatGPT and by Google Bard for each one of the five topics evaluated using the Likert Scale.

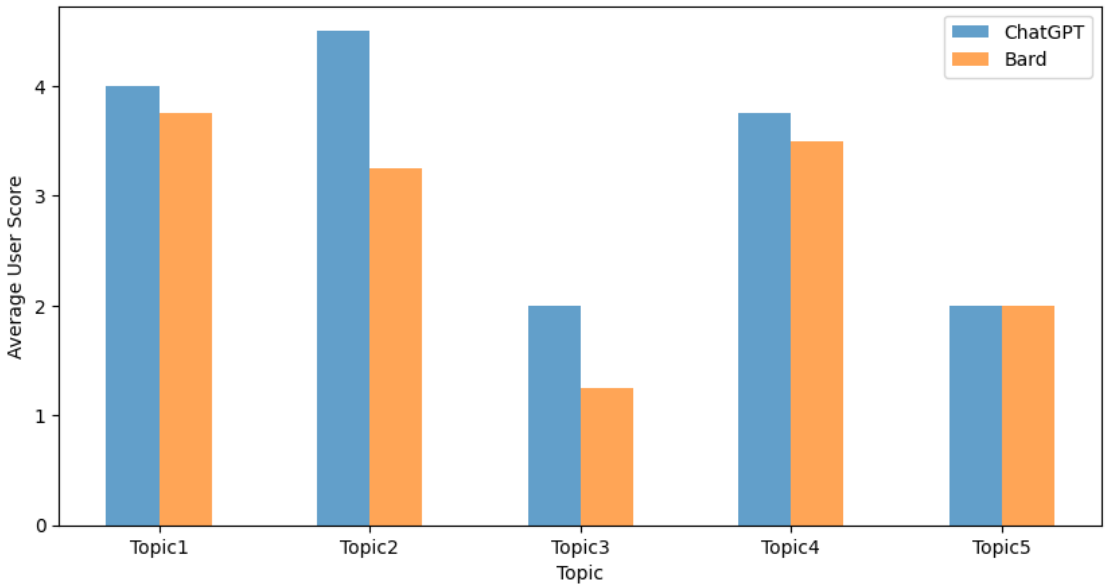

Figure 6.2: Average user score assigned to each topic label generated by ChatGPT and Bard.

According to the graph, users preferred the ChatGPT output for four of the topics, with only topic five being a tie. Only four users is a small sample, but it is still interesting to see that this evaluation group considered none of the Bard labels the best. Figure 6.3 gives a more generalized look at the difference in classifications for ChatGPT and Bard.

6.4 Discussion

By performing all these different tests, it was possible to understand that the work developed produced satisfactory for each of the tasks that the pipeline encapsulates. However, there are still some points that are worth highlighting and discussing.

For transport-related text classification in particular, it is possible to understand that the best results come from the slower models. For near real-time tasks, unless powerful hardware is ready to be used, it would probably be necessary to abdicate from the best solutions and choose the faster ones, trading quickness for accuracy.

For sentiment analysis, the ensemble results were not great. Still, they are aligned with the results from the individual models, leading to believe that the problem is not in the implementation developed. This new solution would probably improve if the individual algorithms were also improved.

As for the topic modeling, evaluating an automated solution directly integrated with the pipeline was not possible, but experimenting with the chatbots was still interesting. Although most topics were not very challenging, there were still differences between models and room for improvement with a different prompt, possibly containing more information, like the weight of each word for the topic, or by adding more words/context to the lists.

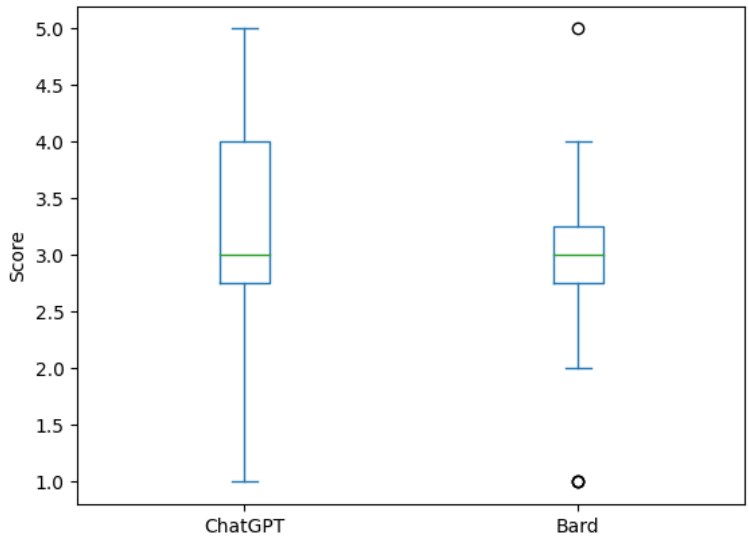

Figure 6.3: Distribution of scores attributed to each model.

After assessing the performance for each task, the next important step would be to evaluate how the implemented pipeline performs as a whole. Since the previous performance metrics only allow a perception of each component's individual performance, identifying real-life cases and trying the implement solution to identify them possibly is a good way to understand how well the project developed performs and if it can already be applied to everyday transport networks evaluation.

The data presented in Section 4.1 was carefully selected so that it was possible to identify cases useful for these tests, and different situations were considered adequate. When looking for cases, the main concern must be ensuring they are as diversified as possible so that this assessment can provide a good perspective on which situations are best evaluated and which are not.

Since there is a lack of open source data regarding traffic congestion for specific locations, some assumptions need to be made while choosing real-life cases, like the existence of more traffic during rush hours or congestion near sports facilities right before and after the games (people arriving and leaving).

The rigorous evaluation approach had several benefits that contributed to the pipeline's overall quality. Firstly, it ensured that each component performed optimally in isolation, addressing unique challenges posed by the transport data. By targeting the specific characteristics of each task, the tests delivered insights that a global test might have overlooked, revealing opportunities for fine-tuning and improvement.

Unfortunately, due to time and data limitations, a comprehensive, end-to-end pipeline test was not feasible within this project's scope. However, it's essential to note that, while valuable, such a test would not fundamentally alter the understanding of the system's capabilities. The robustness of the individual task testing and the usage of real-world data already offers a realistic representation of how the pipeline would function in practical situations, particularly considering that transports content was used during the evaluations for two of the three tasks.

Another issue is that due to the lack of exact locations on the social media content and the normal day-to-day, multiple things are happening in New York simultaneously, so it is almost impossible to study isolated situations.

In summary, while end-to-end testing is beneficial, the individual evaluation of each task with real-world data is a reliable approximation of the pipeline's performance. The thorough examination of each task assures that the pipeline is prepared to handle the complexities and variances it would encounter in practical transport situations, thereby enhancing its overall applicability and robustness. This makes this approach a viable strategy for pipeline testing, particularly in contexts with time or data constraints.

6.5 Summary

This chapter provided an overview of the results obtained from the work developed. In the development of the pipeline for near real-time evaluation of transport networks using social media content, the methodology was grounded on three crucial tasks: text evaluation to filter transport-related content, sentiment analysis to gauge public perception, and topic modeling combined with topic labeling to categorize the issues at hand. To fully understand the performance and capability of the pipeline, each task was scrutinized individually using relevant performance metrics. The chapter started by presenting the text classification task metrics results, where it was possible to see the performance achieved for each individual algorithm and for the two different ensemble approaches. After the text classification, the next task assessed was the sentiment analysis, and here, it demonstrated the dispersion between algorithms, followed by the polarity values predicted using the ensemble. There was also a comparison between the performance results for each algorithm and the base ensemble developed. There were also comparisons between this ensemble and a weighted average one and an approach for which the ensemble was only executed for cases with predictions close to a polarity of 0. Topic modeling and labeling was the last task evaluated, and it started with a display of the topics found in the dataset used, together with the labels generated by the two different LLMs used, ChatGPT and Bard. This section finished with an evaluation of the suggested labels using a Likert Scale to measure each suggestion's quality and usefulness. Although the real-life cases ended up not being a part of the tests, the content retrieved related to them supported some of the evaluations done for each task, so this chapter ended with a reflection on them and their necessity.

Chapter 7

Conclusions and Future Work

This work proposed a pipeline capable of automatically evaluating transportation networks using microblogs from social media, in this case, Twitter. The pipeline includes three different tasks: transports-related text classification (i), which is a bagging ensemble constituted by six different algorithms, SVM, RF, LR, a BERT-Base, and BERT Large without additional training and a BERT Base fine-tuned, implemented using two different ensemble approaches, majority, and weighted voting; sentiment analysis (ii), for which the study conducted showed that there was a lot of dispersion in the literature, so it was necessary to create something that could fix this while maintaining accurate predictions. A bagging regression ensemble composed of the algorithms and tools VADER, TextBlob, Afinn, and BERT, and implemented using a regular and a weighted average; topic modeling together with labeling (iii), implemented with an LDA algorithm used to identify topics in multiple tweets and then ChatGPT and Bard used to generate labels that could sum up these same topics. These three tasks collectively provide data that allows the identification of relevant transports content, displaying the different users' sentiments, and understanding of the topics being discussed that are expected to be relevant traffic events and their implications.

The development of this pipeline was supported by a literature review conducted during the dissertation that allowed to identify the most common approaches for this type of problem and also to understand what could still be improved or even implemented for the first time.

The work developed was then carefully evaluated, with this evaluation showing that each individual task achieved good results and improved what was already available. The text classification implementation achieved an accuracy of 96% making it almost as good as a manual extraction which takes much more time. The sentiment analysis ensemble maintained the results obtained by the individual and well-established algorithms while still dealing with the dispersion of polarities. The topic model with labeling produced good results, with users classifying most labels with a value of three or more on a five-level scale. As for the entire pipeline, even though not fully confronted with real-life cases, it was possible to extrapolate from the tasks evaluated, considering possible real cases that, achieving the best results possible takes more time than what is desired for near real-time tasks since the best solutions for each task are also the slowest.

After this summary of the work, this chapter is divided into four sections. It starts by explaining what are considered the main contributions resulting from the project (Section 7.1). This is followed by the limitations faced and possible improvements that can be done to deal with them (Section 7.2). The next section talks about possible future work that can be interesting to explore, considering what was already done and what can still be added (Section 7.3). This chapter ends with a section that details the publications developed during the dissertation (Section 7.4).

7.1 Main Contributions

The first contribution from this dissertation was a pseudo-systematic review provided in the state-of-the-art. The study done during the initial phase of the work made it possible to compile several works relevant to the topic of using social media to evaluate transports automatically. Subjects like traffic events and incidents detection, transportation user satisfaction, and congestion prediction were some of the ones studied and presented here. The resulting table is an easy way for other authors to understand what was already explored and to think of new ways to improve it.

Regarding the pipeline developed, it is believed that this is the first implementation that includes text classification, sentiment analysis, and topic modeling combined with topic labeling to evaluate transport networks. This makes it an important step to this topic, with the added plus that it can be recreated and improved with new tasks or changes to the ones already present.

For transportation-related text classification, besides providing a comparison between different individual solutions, there is also an ensemble that produces results at least 15% better than what was used in other works and can help authors to operate with more relevant information. Considering the research done, it is believed that this was also the first time Google BERT was fine-tuned to perform text classification regarding transports in general.

For sentiment analysis, the referenced implementation provides a way to deal with the clearly visible dispersion across the literature. Although sentiment analysis ensembles are not new, the one implemented is focused on the most common models used for the topic of transports, so it can be attractive for other authors to test their works and see if there are differences in the evaluation results.

As for topic modeling, since this part was mainly tested for a general context and not only for transports, this work provided a new way of advancing the generation of labels for topics. Despite the fact that this was a relatively small study, the potential shown makes it an important step and motivation to try more models and different prompts.

Overall, the work developed showed the capacity to respond to the problem and the potential to keep being improved so that the evaluation process can become even more complete. The pipeline itself is also a contribution for anyone interested in this type of automatic transport network evaluation.

7.2 Limitations

Although this work makes important contributions to the topic of using social media as a data source to evaluate transport networks, there are still some limitations and, therefore, things that can be improved.

First, the dataset of transports-related social media content. Currently, it only has 3000 entries, with 1500 being related and 1500 being unrelated to transports. It would be important to create a bigger dataset, more diversified regarding, e.g., types of traffic events, hours, and locations. This is a crucial step to improving the training and having performance results that reflect the truth.

Since the older extracting process is now discontinued, the pipeline still needs a way to extract new data. A possible improvement would be to study how the older extraction process can be changed and calculate the costs of using the new Twitter API.

There are also concerns regarding the execution speed of the pipeline, mainly related to the text classification task. A good improvement would be to find new ways to reduce this time, making the ensemble a more viable choice for near real-time uses.

The last limitation is related to the speed at which natural language processing content was being published during this work. 2022 was, and 2023 is, a very important year for this topic, with really significant advances, mainly in LLMs. It was not easy to keep track of everything that was happening, so of course, there are probably new models or algorithms that can be interesting additions to the tasks studied.

7.3 Future Work

The work done during this dissertation allowed to explore new ways to evaluate traffic networks using social media. It is important to keep working and researching new methods to complement it, possibly producing more accurate results, and giving more information to interested users.

Some suggested approaches to explore in the future are:

- Create a large dataset of manually labeled transportation content. The lack of relevant data makes training and evaluating different models very difficult. Since everyone works on different data, it is also complicated to compare solutions. Putting together a significant amount of data and labeling it according to well-detailed criteria would greatly contribute to the research community. With the advances in text generation, using a model like GPT4 to generate social media-like text can be interesting and save a lot of time.

- Use more than one text-based social media as a data source. Depending on the city that is being studied, there can be more than one social network that people use to communicate their thoughts, so by only using one, it is possible that information is being lost. It is even confirmed that Meta is working on a Twitter clone [16], and it can become another important data source.

- Explore image or video-based social networks. Depending on the permission to extract and use this type of content, it would be interesting to use this type of social network to look for images/videos that are related to what users are writing about. It would even be more interesting to train computer vision models so they could identify and label events like crashes or traffic jams.
- Work with more than one language. Big cities are multicultural, so only dealing with content written in English can mean that a big part of the information is being lost. Either translating content from other languages to English or having different models that can perform the described tasks using multiple languages could lead to more reliable results.
- Develop a mobile application with lighter models so that users can use this on the go for their desired locations.
- Study the relation between the average sentiment polarity value and the gravity of the situations detected. If a relation is found, it might be possible for the pipeline to make better suggestions about what is happening using this polarity as an indicator.
- Try to use data regarding future events like the archives retrieved for New York City and see if it is possible to use them to justify traffic problems. It could be very interesting to have a tool capable of giving as output a problem and its possible origin, e.g., "Congestion on 5^{th} avenue possible due to a basketball game nearby".

Finally, even though this work was focused on transportation, it could also be adapted for many subjects like politics or customer satisfaction. Changing the extraction process for the desired topic could, e.g., enable a company to identify specific stores from which users complain. Making this adaptation and exploring other problems could be a viable path for future work and provide significant advances for user opinion mining.

7.4 Publications

This dissertation resulted in content that was included in two different papers. The first paper was initially written during Murçós' work [79]. The reviewing work consisted of complementing the literature review with the new papers reviewed during this dissertation and explaining some of the topics in more detail, considering the newly acquired knowledge and the reviewers' suggestions.

- Tânia Fontes, Francisco Murços, Eduardo Carneiro, Joel Ribeiro, and Rosaldo Rossetti - "**Leveraging social media as a source of mobility intelligence: A NLP-based approach**" In IEEE Open Journal of Intelligent Transportation Systems. In review.

The second paper results from the study and development done for transportation-related content classification. It provides a review of how this is done across the available literature, a comparison between three different groups of models, and a suggestion about how someone can choose which model is the most adequate for a specific task.

- Eduardo Carneiro, Tânia Fontes, and Rosaldo Rossetti - "**Enhancing decision-making in transportation management: A comparative study of text classification models**" In 26th IEEE International Conference on Intelligent Transportation Systems ITSC 2023. Accepted.

After the dissertation is finished, there is also the ambition to publish two more papers, one that provides a review of what algorithms authors use to perform sentiment analysis on content related to transportation and the differences between them and a paper that includes the development work done during this project, with particular focus on the pipeline implemented.

References

[1] Twitter api documentation | docs | twitter developer platform. Available at https://developer.twitter.com/en/docs/twitter-api, Accessed: July 2022.

[2] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37(3):279–294, 2009.

[3] Stamatios-Aggelos Alexandropoulos, Christos Aridas, Sotiris Kotsiantis, and Michael Vrahatis. *Stacking Strong Ensembles of Classifiers*, pages 545–556. 05 2019.

[4] Farman Ali, Shaker El-Sappagh, and Daehan Kwak. Fuzzy ontology and lstm-based text mining: A transportation network monitoring system for assisting travel. *Sensors*, 19(2), 2019.

[5] Farman Ali, Daehan Kwak, Pervez Khan, Shaker El-Sappagh, Amjad Ali, Sana Ullah, Kye Hyun Kim, and Kyung-Sup Kwak. Transportation sentiment analysis using word embedding and ontology-based topic modeling. *Knowledge-Based Systems*, 174:27–42, 2019.

[6] Adel Almohammad and Panagiotis Georgakis. Public twitter data and transport network status. In *2020 10TH INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND TECHNOLOGY (ICIST)*, pages 169–174, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2020. IEEE.

[7] Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. Automatic generation of topic labels. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, jul 2020.

[8] Sonia Anastasia and Indra Budi. Twitter sentiment analysis of online transportation service providers. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 359–365, 2016.

[9] A. R. Atmadja, W. Uriawan, F. Pritisen, D. S. Maylawati, and A. Arbain. Comparison of naive bayes and k-nearest neighbours for online transportation using sentiment analysis in social media. In *4TH ANNUAL APPLIED SCIENCE AND ENGINEERING CONFERENCE, 2019*, volume 1402 of *Journal of Physics Conference Series*. IOP PUBLISHING LTD, 2019.

[10] Anique Azhar, Saddaf Rubab, Malik M. Khan, Yawar Abbas Bangash, Mohammad Dahman Alshehri, Fizza Illahi, and Ali Kashif Bashir. Detection and prediction of traffic accidents using deep learning techniques. *Cluster Computing*, 26(1):477–493, 2023.

[11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, mar 2003.

[12] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 953–963, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, mar 2003.

[14] Subhasree Bose, Urmi Saha, Debanjana Kar, Saptarsi Goswami, Amlan Nayak, and Satyajit Chakrabarti. *RSentiment: A Tool to Extract Meaningful Insights from Textual Reviews*, pages 259–268. 03 2017.

[15] Bret Boyd. Urbanization and the mass movement of people to cities, Dec 2019. Available at https://graylinegroup.com/urbanization-catalyst-overview/, Accessed: May 2022.

[16] John Brandon. Meta is working on a "sanely run" twitter clone, Jun 2023. Available at https://www.forbes.com/sites/johnbbrandon/2023/06/10/meta-is-ready-to-launch-a-sanely-run-twitter-clone/?sh=8c621004360f, Accessed: July 2023.

[17] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[18] Peter Bühlmann. Bagging, boosting and ensemble methods. *Handbook of Computational Statistics*, 01 2012.

[19] Emre Calisir and Marco Brambilla. The problem of data cleaning for knowledge extraction from social media. In Cesare Pautasso, Fernando Sánchez-Figueroa, Kari Systä, and Juan Manuel Murillo Rodríguez, editors, *Current Trends in Web Engineering*, pages 115–125, Cham, 2018. Springer International Publishing.

[20] Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco, editors. *A practical guide to sentiment analysis*. Socio-Affective Computing. Springer International Publishing, Basel, Switzerland, 1 edition, April 2017.

[21] Sophia Chan and Alona Fyshe. Social and emotional correlates of capitalization on Twitter. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 10–15, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics.

[22] Haoliang Chang, Jianxiang Huang, Weiran Yao, Weizun Zhao, and Lishuai Li. How do new transit stations affect people's sentiment and activity? a case study based on social media data in hong kong. *Transport Policy*, 120:139–155, 2022.

[23] Haoliang Chang, Lishuai Li, Jianxiang Huang, Qingpeng Zhang, and Kwai-Sang Chin. Tracking traffic congestion and accidents using social media data: A case study of shanghai. *Accident Analysis Prevention*, 169:106618, 2022.

[24] Xu Chen, Zihe Wang, and Xuan Di. Sentiment analysis on multimodal transportation during the covid-19 using social media data. *Information*, 14(2), 2023.

[25] Noam Chomsky. *Syntactic Structures*. Mouton Publishers, The Hague, Netherlands, 1957.

[26] Rob Churchill and Lisa Singh. The evolution of topic modeling. *ACM Comput. Surv.*, 54(10s), nov 2022.

[27] Weibo Corporation. Business overview. Available at http://ir.weibo.com/corporate-profile, Accessed: July 2022.

[28] Metropolitan Council. White paper 1: The negative effects of traffic congestion on the twin cities and the state of minnesota. Technical report, 2020.

[29] J.S. Cramer. The origins of logistic regression. Working Paper 2002-119/4, Tinbergen Institute, December 2002.

[30] Monir Dahbi, Rachid Saadane, and Samir Mbarki. Social media sentiment monitoring in smart cities an application to moroccan dialects. In *4TH INTERNATIONAL CONFERENCE ON SMART CITY APPLICATIONS (SCA' 19)*, 1515 BROADWAY, NEW YORK, NY 10036-9998 USA, 2019. ASSOC COMPUTING MACHINERY.

[31] Subasish Das, Xiaoduan Sun, and Anandi Dutta. Text mining and topic modeling of compendiums of papers from transportation research board annual meetings. *TRANSPORTATION RESEARCH RECORD*, (2552):48–56, 2018.

[32] Stephen Boyd Davis and Mike Saunders. How social media can help improve and redesign transport systems, Jun 2014. Available at https://www.theguardian.com/sustainable-business/social-media-redesign-transport-systems-cities, Accessed: June 2022.

[33] Essam Debie and Kamran Shafi. Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses. *Pattern Analysis and Applications*, 22(2):519–536, 2019.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[35] The Economist. The hidden cost of congestion, Feb 2018. Available at https://www.economist.com/graphic-detail/2018/02/28/the-hidden-cost-of-congestion, Accessed: July 2022.

[36] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation, 2020.

[37] International Transport Forum. *ITF Transport Outlook 2017*. 2017.

[38] Ayelet Gal-Tzur, Susan M. Grant-Muller, Tsvi Kuflik, Einat Minkov, Silvio Nocera, and Itay Shoor. The potential of social media in delivering transport policy goals. *Transport Policy*, 32:115–123, 2014.

[39] Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms: From text to predictions. *Information*, 13(2), 2022.

[40] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):2009, 2009.

[41] Santiago González-Carvajal and Eduardo C. Garrido-Merchán. Comparing BERT against traditional machine learning text classification. *CoRR*, abs/2005.13012, 2020.

[42] Tenba Group. What is sina weibo? know your chinese social media!, May 2022. Available at https://tenbagroup.com/what-is-sina-weibo-know-your-chinese-social-media/, Accessed: July 2022.

[43] N. Nima Haghighi, Xiaoyue Cathy Liu, Ran Wei, Wenwen Li, and Hu Shao. Using twitter data for transit performance assessment: a framework for evaluating transit riders' opinions about quality of service. *PUBLIC TRANSPORT*, 10(2):363–377, AUG 2018.

[44] N. Nima Haghighi, Xiaoyue Cathy Liu, Ran Wei, Wenwen Li, and Hu Shao. Using twitter data for transit performance assessment: a framework for evaluating transit riders' opinions about quality of service. *Public Transport*, 10(2):363–377, 2018.

[45] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[46] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5:01–11, 03 2015.

[47] W. John Hutchins. The georgetown-ibm experiment demonstrated in january 1954. In Robert E. Frederking and Kathryn B. Taylor, editors, *Machine Translation: From Real Users to Research*, pages 102–114, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[48] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.

[49] Clayton J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14).*, pages 216–225, 2014.

[50] Anne Immonen, Pekka Pääkkönen, and Eila Ovaska. Evaluating the quality of social media data in big data architecture. *IEEE Access*, 3:2028–2043, 2015.

[51] Tommi S. Jaakkola and Michael I. Jordan. *Improving the Mean Field Approximation Via the Use of Mixture Distributions*, pages 163–173. Springer Netherlands, Dordrecht, 1998.

[52] Dave Johnson. 'what is linkedin?': A beginner's guide to the popular professional networking and career development site, Sep 2019. Available at https://www.businessinsider.com/what-is-linkedin, Accessed: July 2022.

[53] Karen Sparck Jones. *Natural Language Processing: A Historical Review*, pages 3–16. Springer Netherlands, Dordrecht, 1994.

[54] Daniel Jurafsky and James H Martin. *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft Summary of Contents*. Third edition draft edition, 2023.

[55] Sarah M Kaufman. How social media moves new york: Twitter use by transportation providers in the new york region. Technical report, 2012.

[56] Simon Kemp. Digital 2022: Global overview report - datareportal – global digital insights, Jan 2022.

[57] Pooja Kherwa and Poonam Bansal. Topic modeling: A comprehensive review. *ICST Transactions on Scalable Information Systems*, 7:159623, 07 2018.

[58] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744, July 2022.

[59] Gargi Kulkarni, Lourdes Abellera, and Anand Panangadan. Unsupervised classification of online community input to advance transportation services. In *2018 IEEE 8TH ANNUAL COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE (CCWC)*, pages 261–267, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2018. IEEE.

[60] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6, 2016.

[61] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020.

[62] Dawei Li, Yujia Zhang, and Cheng Li. Mining public opinion on transportation systems based on social media data. *SUSTAINABILITY*, 11(15), AUG 2019.

[63] Elizabeth D. Liddy. Natural language processing. In *Encyclopedia of Library and Information Science*. Marcel Decker, Inc., New York, 2nd edition, 2001.

[64] Linkedin. What is linkedin and how can i use it? Available at https://www.linkedin.com/help/linkedin/answer/a548441/what-is-linkedin-and-how-can-i-use-it-?lang=en, Accessed: July 2022.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[66] Margarita Loktionova. Top trending topics: What people talk about online, May 2022. Available at https://www.semrush.com/blog/top-trending-topics/, Accessed: July 2022.

[67] Shuli Luo, Sylvia Y. He, Susan Grant-Muller, and Linqi Song. Influential factors in customer satisfaction of transit services: Using crowdsourced data to capture the heterogeneity across individuals, space and time. *Transport Policy*, 131:173–183, 2023.

[68] Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic labeling of topics. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pages 1227–1232, 2009.

[69] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[70] Dale Markowitz. Meet ai's multitool: Vector embeddings, Mar 2022. Available at https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings, Accessed: April 2023.

[71] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[72] Wang Mengzhen. China's national library to archive 200 billion Sina Weibo posts, 2019. Available at https://news.cgtn.com/news/3d3d674d79677a4e34457a6333566d54/index.html, Accessed June 2023.

[73] Meta. Facebook. Available at https://about.facebook.com/technologies/facebook-app/, Accessed: July 2022.

[74] Meta. Instagram. Available at https://about.facebook.com/technologies/instagram/, Accessed: July 2022.

[75] Meta. What is instagram? Available at https://help.instagram.com/424737657584573, Accessed: July 2022.

[76] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. 2013.

[77] Dibya Nandan Mishra and Rajeev Kumar Panda. Decoding customer experiences in rail transport service: application of hybrid sentiment analysis. *Public Transport*, 15(1):31–60, 2023.

[78] Joseph Muguro, Waweru Njeri, Kojiro Matsushita, and Minoru Sasaki. Road traffic conditions in kenya: Exploring the policies and traffic cultures from unstructured user-generated data using nlp. *IATSS Research*, 46(3):329–344, 2022.

[79] Francisco André Barreiros Murçós. Urban Transport Evaluation Using Knowledge Extracted from Social Media. October 2021.

[80] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. 2022.

[81] F. Å. Nielsen. Afinn, Mar 2011.

[82] Zayyana Nurthohari, Dana Indra Sensuse, and Sofian Lusa. Sentiment analysis of jakarta bus rapid transportation services using support vector machine. In *2022 International Conference on Data Science and Its Applications (ICoDSA)*, pages 171–176, 2022.

[83] NY Open Data. 511 NY Sporting, Concert, and Special Events: Beginning 2010, 2010. Available at https://data.ny.gov/Transportation/511-NY-Sporting-Concert-and-Special-Events-Beginni/3ha4-4nfg/data, Accessed: February 2023.

[84] NY Open Data. NYC Permitted Event Information - Historical, 2017. Available at https://data.cityofnewyork.us/City-Government/NYC-Permitted-Event-Information-Historical/bkfu-528j, Accessed: February 2023.

[85] European Court of Auditors. Urban mobility in the eu audit preview. Technical report, 2019.

[86] OpenAI. Gpt-4 technical report. Technical report, 2023.

[87] Joaquin Osorio-Arjona, Jiri Horak, Radek Svoboda, and Yolanda Garcia-Ruiz. Social media semantic perceptions on madrid metro system: Using twitter data to link complaints to space. *SUSTAINABLE CITIES AND SOCIETY*, 64, JAN 2021.

[88] Oxford. Microblogging noun - definition, pictures, pronunciation and usage notes. Available at https://www.oxfordlearnersdictionaries.com/definition/english/microblogging, Accessed: July 2022.

[89] Cristian Padurariu and Mihaela Elena Breaban. Dealing with data imbalance in text classification. *Procedia Computer Science*, 159:736–745, 2019. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019.

[90] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[91] Sarah Penny. Six traffic counts and classification study methods, Jul 2021. Available at https://www.smatstraffic.com/2021/07/21/counts-and-classification-study-methods/, Accessed: July 2022.

[92] João Filipe Figueiredo Pereira. Social media text processing and semantic analysis for smart cities. *CoRR*, abs/1709.03406, 2017.

[93] Martin Perez. What is web scraping and what is it used for?, Dec 2021. Available at https://www.parsehub.com/blog/what-is-web-scraping/, Accessed: July 2022.

[94] Mauro Di Pietro. BERT for Text Classification with NO model training — towardsdatascience.com. Available at https://towardsdatascience.com/text-classification-with-no-model-training-935fe0e42180, Accessed: March 2023.

[95] Pinterest. About pinterest. Available at https://help.pinterest.com/en/guide/all-about-pinterest, Accessed: July 2022.

[96] Bing Qi, Aaron Costin, and Mengda Jia. A framework with efficient extraction and analysis of twitter data for evaluating public opinions on transportation services. *TRAVEL BEHAVIOUR AND SOCIETY*, 21:10–23, OCT 2020.

[97] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[98] Aldy Rialdy, Wisnu Uriawan, F Pritisen, Dian Maylawati, and A Arbain. Comparison of naive bayes and k-nearest neighbours for online transportation using sentiment analysis in social media. *Journal of Physics: Conference Series*, 1402:077029, 12 2019.

[99] Filipe N. Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):23, 2016.

[100] Bruno P. Santos, Paulo H. L. Rettore, Heitor S. Ramos, Luiz F. M. Vieira, and Antonio A. E. Loureiro. Enriching traffic information with a spatiotemporal model based on social media. In *2018 IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS (ISCC)*, IEEE Symposium on Computers and Communications ISCC, pages 469–474, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2018. IEEE, IEEE.

[101] Lisa Schweitzer. Planning and social media: A case study of public transit and stigma on twitter. *Journal of the American Planning Association*, 80(3):218–238, 2014.

[102] SentiStrength. Sentistrength. Available at http://sentistrength.wlv.ac.uk/, Accessed: July 2022.

[103] Foram P. Shah and Vibha Patel. A review on feature selection and feature extraction for text classification. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2264–2268, 2016.

[104] Xinying Song. A fast wordpiece tokenization system, Dec 2021. Available at https://ai.googleblog.com/2021/12/a-fast-wordpiece-tokenization-system.html, Accessed: June 2022.

[105] Internet Live Stats. Twitter usage statistics. Available at https://www.internetlivestats.com/twitter-statistics/, Accessed: May 2022.

[106] HERE Technologies. Here technologies. Available at https://www.here.com/", organization=HERE Technologies, Accessed: July 2022.

[107] Jose Tomas Mendez, Hans Lobel, Denis Parra, and Juan Carlos Herrera. Using twitter to infer user satisfaction with public transport: The case of santiago, chile. *IEEE ACCESS*, 7:60255–60263, 2019.

[108] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.

[109] Twitter. About twitter | our company and priorities. Available at https://about.twitter.com/en, Accessed: July 2022.

[110] Daniela Ulloa, Pedro Saleiro, Rosaldo J. F. Rossetti, and Elis Regina Silva. Mining social media for open innovation in transportation systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 169–174, 2016.

[111] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[112] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[113] Joseph Nathanael Witanto, Hyotaek Lim, and Mohammed Atiquzzaman. Smart government framework with geo-crowdsourcing and social media analysis. *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE*, 89:1–9, DEC 2018.

[114] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer Learning*. Cambridge University Press, 2020.

[115] Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 233–242, New York, NY, USA, 2014. Association for Computing Machinery.

[116] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

[117] Weizhong Zhao, James J. Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC Bioinformatics*, 16:S8 – S8, 2015.

[118] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman Hall/CRC, 1st edition, 2012.

[119] Halil İbrahim Cebeci, Samet Güner, Yusuf Arslan, and Emrah Aydemir. Barriers and drivers for biking: What can policymakers learn from social media analytics? *Journal of Transport Health*, 28:101542, 2023.

Appendix A

New York City Data - Additional Table

This appendix contains information about the New York City files retrieved for the project. The table represents a file and details what each column means.

Table A.1: 511 NY Sporting, Concert, and Special Events: Beginning 2010 fields.

	Description
Event Type	Textual description of what type of event it is
Organization Name	The name of the organization responsible for the event
Facility Name	The facility/location in which the event will take place
Direction	Direction of travel where the event exists
City	The city in which the event will take place
County	The county in which the event will take place
State	The state in which the event will take place
Create Time	Time the event was created in the 511 system
Close Time	Time the event was closed in the 511 system
Event Description	Textual description of what happens during the event
Responding Organization Id	The Id from the organization responsible for the event
Latitude	The latitude of the event
Longitude	The longitude of the event

Appendix B

Data Extraction & Preprocessing Considerations

This appendix serves as a possible guideline for things to consider in future projects that involve social media. It contains considerations regarding both the extraction and the preprocessing processes.

Starting with the extraction, it is important to consider multiple factors when deciding how and from where to do it.

The first thing to consider is the type of analysis that will be conducted, i.e., long, mid, or short-term. For short-term tasks, choosing a social network that allows data extraction in near real-time is necessary. The duration of this process should also not take a lot of time because having the information late makes it less relevant and most likely outdated. For mid and long-term analysis, the flux of information can be slower since the final results are expected with extended deadlines.

Another thing that needs to be considered is how well the extraction can be restricted to a certain area and how good is the location information that comes with each result. Regarding the filter quality, getting information from outside of the restricted area is a problem because it will influence the final results with comments from other places that are not being analyzed. As for the location information, it is much harder to conduct microanalysis without detailed info, so lack of information or poor quality only allows the chance to conduct macro studies.

Lastly, the languages that will be extracted are also something to think about. The methods chosen for the evaluation might not be prepared to deal with certain languages, so making sure only text with the one(s) chosen is selected is very important otherwise, the final results might also be affected.

When trying to improve the quality of the information, especially content from social media, it is crucial to deal with possible spam done by one or more users. Depending on the type of analysis, long or short term, this needs to be taken into account so that the information provided by one single person or by bots does not single handily influence the final result of the study. A suggestion to deal with this during the preprocessing is to check the similarity between different

texts and the time gap between messages from the same user. Using these strategies might allow avoiding situations similar to the one caused by the Berlin artist Simon Weckert, that created the illusion of a traffic jam in Google Maps by putting a red cart with ninety-nine phones on an empty street [1].

As for the preprocessing, it is also important to remember that different tasks will require different types of preprocessing, so guaranteeing the original text form is always preserved, if possible, with an ID will facilitate improving the results for each task compared to an execution done using a single preprocessing format. One important difference to highlight is the casing of the text. While using uncased text is the norm for most category classifiers, for sentiment analysis, it might be useful to know if a person is writing in all caps (possibly indicating a sentiment like anger).

In the past, emojis were discarded by most authors, but nowadays, there are libraries and other tools available to convert them into text. By doing this, the texts that the algorithms receive will have more context and possibly more information which can help both for text classification (e.g., car emoji) and for sentiment analysis (e.g., crying emoji).

The last thing that can also be done is to look for possible language combinations in the text (content with two or more languages) and translate it to the primary one in analysis. It is relatively common among certain age groups to use expressions from other languages. Although this is something easy to understand for humans (if they are familiar with the language), most algorithms will expect just one singular language.

[1] https://www.simonweckert.com/googlemapshacks.html

Appendix C

OpenAI GPT Costs

It was necessary to calculate the possible costs of using OpenAI GPT models to decide if they would be viable for this work.

The first cost estimation was the following:

- 18 dollars of initial credit to try the solution.
- 0.02 for every 1000 tokens.
- 1000 tokens ≈ 750 words.
- (18/0.02) * 1000 = 900k tokens ≈ 675K words.
- This is just for base question-answering tasks, using fine-tuned models is even more expensive, and it is necessary to consider the multiple epochs used to train models.

After this initial calculation, with the release of GPT3.5 and GPT4 and the increasing popularity of ChatGPT, there were cost changes. Still, it was already decided that only free models would be used.

Nevertheless, Figure C.1 shows ChatGPT prediction for a tweet that is related to transports, and Figure C.2 shows the prediction for a tweet that clearly is unrelated.

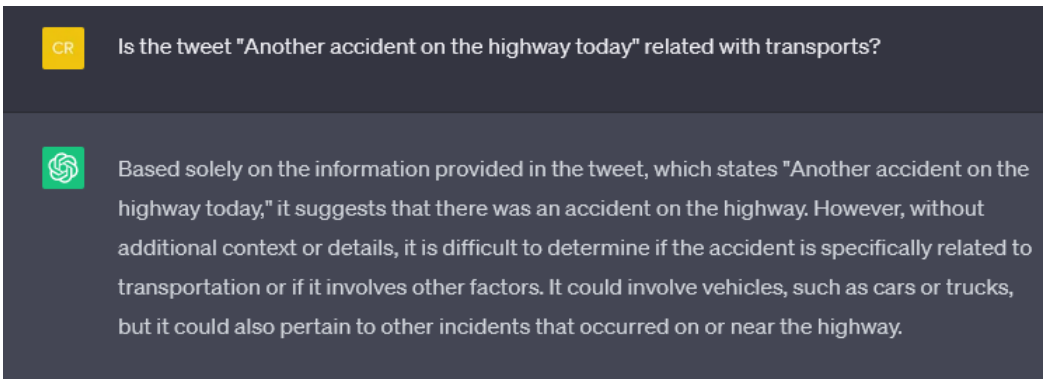

Figure C.1: ChatGPT prediction for a tweet related to transports.

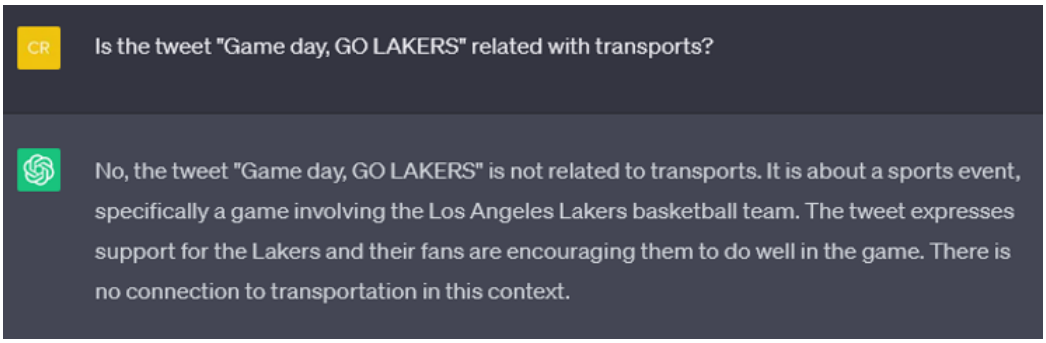

Figure C.2: ChatGPT prediction for a tweet unrelated to transports.