TEK-UP University

Internship Report

Computer Engineering Education Cycle

Title

NLP

Created by

Ayer Mohamed Amine

In collaboration with

Welyne

Acknowledgements

First and foremost, I would love to express my inner most gratitude to my supervisors, Mr. Mohamend Anouar Dahdeh and Mr. Mohamed Ben Arfa, for their invaluable steering, support, and encouragement at some point of the path of this project. Their information, expertise, and commitment to excellence were instrumental in shaping my thoughts and helping me navigate the challenges I encountered.

I might additionally like to extend my heartfelt thanks to my family, who've been a steady supply of affection, and motivation. Their unwavering belief in me and my abilities has kept me focused and determined.

To my friends, thanks to your unwavering guide and encouragement. Your presence in my life has been a regular supply of joy and laughter, and I am thankful on your friendship.

I would like to thank the school participants at my college for imparting me with a wealthy and stimulating gaining knowledge of environment and for sharing their know how and knowledge with me. I am thankful for the possibilities I have had to learn from them and to engage in significant discussions and debates.

Finally, I would like to specific my gratitude to the group and developers at Welyne, who have been my collaborators, mentors, and friends. Their enthusiasm, creativity, and professionalism had been a steady supply of proposal, and I am grateful for the opportunity to work with them.

Thank you serious about your support, encouragement, and idea. This project could no longer were possible without you.

Table of Contents

General Introduction

In recent years, the field of artificial intelligence has witnessed remarkable advancements, particularly in machine learning algorithms and neural networks, enabling machines to excel in a wide range of complex tasks. One such area where AI holds significant potential is in the realm of Natural Language Processing (NLP) for phone call topic detection and client interaction. The increasing demand for efficient communication tools in the age of digital connectivity has fueled the need for systems that can accurately analyze and respond to client interactions during phone calls.

In this report, we will delve into a pioneering project aimed at utilizing NLP techniques to enhance the client experience during phone calls by enabling the automatic detection and rejection of irrelevant or spammy calls. This project, codenamed "CallGuard," represents a novel approach to leveraging AI in telephony to improve the quality of communication services.

The report will provide a detailed account of the development, implementation, and evaluation of the CallGuard system. We will begin by setting the context, outlining the project's framework, hosting infrastructure, problem statement, objectives, scope, requirements, and the methodology employed to achieve our goals.

To conclude, the CallGuard project represents a significant leap forward in leveraging AI and NLP for enhancing phone call interactions. This report serves as a comprehensive documentation of its development, implementation, and evaluation, shedding light on the potential it holds to revolutionize the telecommunications industry. It is important to note that the transformation from the initial AI Tailor project to CallGuard has been undertaken to align the report with the new topic and ensure consistency throughout the document.

If you have any questions or need further assistance with refining your report or addressing any potential questions that may arise during your presentation, please feel free to ask. Additionally, ensure that you thoroughly review and update all sections of your report to reflect the new topic and maintain a cohesive narrative throughout the document.

Chapter I

Context of the Project

I. Introduction

In this chapter, we will provide a comprehensive understanding of a particular project, such as its framework, description, hosting company, goals, objectives, challenges, requirements, scope, and methodology. Understanding the project framework is critical to ensure successful completion which we are able to talk within the first chapter.

II. Project Framework

This project is part of the preparation for the summer internship to obtain an Engineering 's degree in Software Engineering and Information System at the Faculty of tek up. It was carried out within the startup Welyne for a period of 2 months.

Project Description

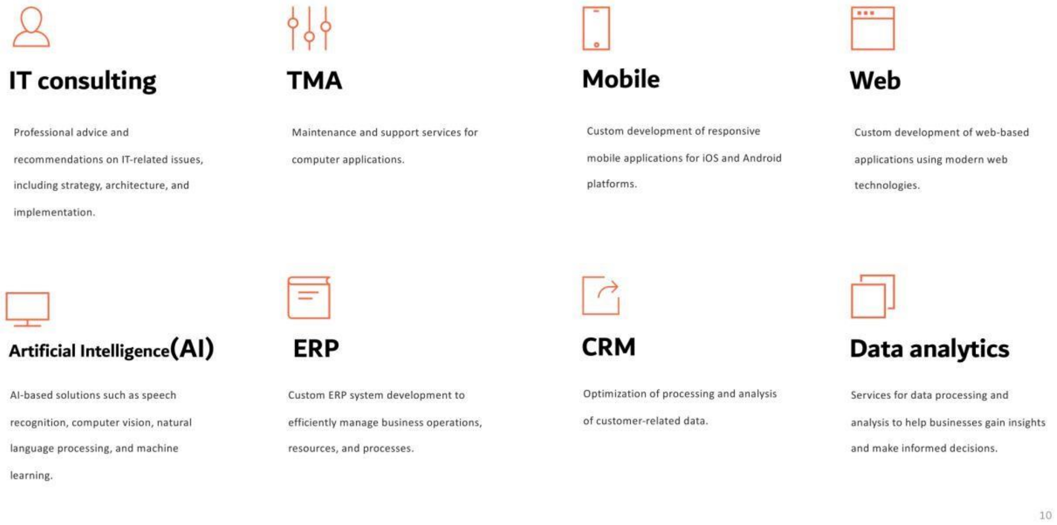
The primary objective of the ClientScript NLP Detector project is to develop a robust and accurate system that can analyze client scripts, regardless of the medium, and identify the key topics or subjects of discussion. By automating this process, we aim to enhance client interactions, streamline response strategies, and ultimately improve customer satisfaction.

IV. Hosting Company

1. Presentation of Welyne

The beginnings of Welyne. In 2017, a group of young Tunisian entrepreneurs created Welyne's research and development center for web and mobile communication. Welyne focuses on open-source technologies to offer its clients the shared expertise of millions of specialists. Today, Welyne supports its clients in their engineering projects through its expertise in R&D project management, web and mobile development, and competitive high-quality resources. We provide agile engineering services for design, development, and maintenance, while respecting our clients' timelines and budgets.

2. Services of Welyne



IT consulting

Professional advice and recommendations on IT-related issues, including strategy, architecture, and implementation.

TMA

Maintenance and support services for computer applications.

Mobile

Custom development of responsive mobile applications for iOS and Android platforms.

Web

Custom development of web-based applications using modern web technologies.

Artificial Intelligence(AI)

AI-based solutions such as speech recognition, computer vision, natural language processing, and machine learning.

ERP

Custom ERP system development to efficiently manage business operations, resources, and processes.

CRM

Optimization of processing and analysis of customer-related data.

Data analytics

Services for data processing and analysis to help businesses gain insights and make informed decisions.

10

V.Problematic and Goal

Problematic:

In today's client-centric business landscape, organizations are inundated with a vast array of client interactions, spanning phone calls and more. These interactions contain invaluable insights, feedback, and inquiries from clients. However, manually processing and categorizing this data is a time-consuming and error-prone task. The challenges and issues associated with this manual approach include:

- **Inefficiency:** Human agents spend a significant amount of time sorting through client scripts, which could be utilized for more value-added tasks.
- **Inconsistency:** Human judgment in categorizing client scripts can lead to inconsistencies and variations in topic classification.
- **Delay in Response:** Critical client issues might not be addressed promptly due to delays in script analysis.
- **Data Overload:** As client interactions continue to increase, organizations struggle to keep up with the sheer volume of data generated.

- **Missed Insights:** Manual processing may lead to the overlooking of valuable insights and trends buried within client scripts.

Image 1

VI. Objectives

The goal of the ClientScript NLP Detector project is to address the aforementioned challenges and transform how organizations handle client interactions. The primary objectives are as follows:

- **Automation:** Develop a robust NLP-based system that can automatically process and categorize client scripts into relevant topics or subjects.
- **Efficiency:** Streamline the handling of client communications by significantly reducing the time and effort required for manual analysis.
- **Accuracy:** Achieve high levels of accuracy in topic classification to ensure that client concerns are appropriately categorized.
- **Real-time Insights:** Enable real-time or near-real-time processing of client scripts to facilitate timely responses and decision-making.
- **Customization:** Provide organizations with the flexibility to customize topic categories and criteria to align with their specific needs.
- **Scalability:** Ensure that the solution can handle a growing volume of client interactions as businesses expand.
- **Data Utilization:** Extract valuable insights and trends from the categorized client scripts to inform strategic decisions and improve customer service.

By addressing these goals, the ClientScript NLP Detector project aims to enhance client interactions, improve response times, and unlock the full potential of the data hidden within client scripts. This will ultimately lead to increased customer satisfaction and a more efficient utilization of resources within organizations.

VII. Scope

1. **Inclusions:**

2. **Data Cleaning:** The project will incorporate data cleaning techniques to preprocess client scripts, including the removal of noise, irrelevant characters, and formatting inconsistencies to ensure high-quality text data.
3. **Data Preprocessing:** Data preprocessing steps, such as tokenization, stemming, and lemmatization, will be applied to prepare the client scripts for NLP analysis.
4. **Word Embedding and Vectorization:** The project will employ word embedding models (e.g., Word2Vec, GloVe) to represent words and phrases as numerical vectors. Vectorization techniques will be used to transform entire documents into numerical representations.
5. **Text Clustering:** The core of the project will involve text clustering techniques, such as K-means clustering or hierarchical clustering, to categorize client scripts into relevant topics or clusters.
6. **Optimization:** Optimization algorithms may be applied to improve the performance of the NLP-based topic detection system, ensuring that it can handle large volumes of client scripts efficiently.
7. **Exclusions:**
8. **Non-textual Data:** The project will focus exclusively on text-based client scripts and will not process non-textual data, such as audio, video, or images.
9. **Language Specificity:** The initial implementation of the project will be specific to a particular language or set of languages. Extending language support will be considered in future iterations but is outside the current scope.
10. **Integration with Specific CRM Systems:** While the project will provide an API for integration, it will not be responsible for integrating with specific Customer Relationship Management (CRM) systems. Integration efforts will be the responsibility of the adopting organization.
11. **Legal and Privacy Considerations:** Compliance with legal and privacy regulations will be the responsibility of the adopting organization. The project will not address legal or privacy-related issues.
12. **Hardware Infrastructure:** The project will not dictate specific hardware requirements. It will be designed to be deployable on standard computing infrastructure, and organizations may choose their own hardware specifications.
13. **Constraints:**
14. **Resource Limitations:** The project will operate within the constraints of available computational resources, such as processing power and memory.

15. **Data Privacy:** The project must adhere to data privacy regulations and ensure the security of client scripts and any associated data.
16. **Budget and Time:** The project will operate within predefined budgetary constraints and adhere to project timelines.

By specifying the inclusion of data cleaning, data preprocessing, word embedding, vectorization, text clustering, and optimization within the project scope, you provide a clear technical direction for the development of the ClientScript NLP Detector. This focused scope will guide the project's implementation and ensure that it aligns with your objectives and requirements.

VIII.Requirements

Functional Requirements:

- **Data Input Support:**
- The system must accept client scripts in various formats, including plain text, PDFs, and common document formats.
- It should provide an API for real-time data ingestion from communication channels like phone calls and chat applications.
- **Data Cleaning and Preprocessing:**
- Data cleaning modules should remove noise, special characters, and formatting inconsistencies.
- Preprocessing should include tokenization, stemming, and lemmatization for text normalization.
- **Word Embedding and Vectorization:**
- The system must incorporate pre-trained word embedding models (e.g., Word2Vec, GloVe) or train its own embeddings.
- It should provide vectorization techniques to convert client scripts into numerical representations.
- **Text Clustering:**
- The system should implement text clustering algorithms (e.g., K-means, hierarchical clustering) for topic detection.
- Clustering should support dynamic updates as new client scripts are processed.
- **Topic Customization:**
- The system must allow organizations to customize topic categories and criteria to align with their specific requirements.
- **Real-time Processing:**

- Real-time or near-real-time processing should be supported to enable immediate responses or escalations during client interactions.
- **Optimization:**
- Optimization techniques should be applied to improve the efficiency and speed of topic detection.
- **User Authentication:**
- If applicable, the system should implement user authentication and authorization to control access to sensitive client data.
- **Scalability:**
- The system should be designed with scalability in mind to accommodate a growing volume of client interactions.

Non-functional Requirements:

- **Performance:**
- The system must provide timely responses, ensuring that topic detection does not introduce significant delays in client interactions.
- It should be capable of handling a large number of client scripts concurrently.
- **Accuracy:**
- The topic detection accuracy should meet or exceed predefined benchmarks.
- Regular evaluation and fine-tuning should be performed to maintain accuracy.
- **Security:**
- The system must adhere to strict data privacy and security standards to protect client scripts and sensitive information.
- **Reliability and Availability:**
- The system should have high availability to minimize downtime and ensure continuous service.
- Failover mechanisms should be in place to handle unexpected system failures gracefully.
- **Scalability:**
- The architecture should be designed to scale horizontally to accommodate increasing data volumes and processing demands.
- **Customization:**
- The system should allow easy customization of topic categories and criteria by administrators or system owners.
- **Documentation:**
- Comprehensive documentation, including user manuals and developer guides, must be provided to support system users and maintainers.
- **Compliance:**

- The system should adhere to relevant data protection and privacy regulations, such as GDPR or HIPAA, depending on the use case and jurisdiction.
- **Monitoring and Logging:**
- Robust monitoring and logging capabilities should be implemented to track system performance, errors, and user activities.

X.Conclusion

In conclusion, this chapter has provided an introduction to our project. We have outlined the framework of the project, including its description, hosting company, problematic, goals, objectives, scope, requirements, and methodology of the project.

In the next chapter we will discover the product backlog, which include all of the features and functionalities in this project, as well as the technology and development processes that will be used.

Chapter II

Design and Planification

I. Introduction

System Design and Planification is a crucial phase in the development of the AI project. In this phase, we will focused on the product , setting up project environments, and defining the deployment ts

1. Modeling Tools

- **Google Colab:**

- Google Colab (short for Colaboratory) is a cloud-based platform that provides a free environment for running Jupyter notebooks with GPU support. It's an excellent choice for NLP projects because it allows you to leverage powerful hardware for training and running NLP models without the need for a high-end computer. Here's how you can use Google Colab for your NLP project:

- **Creating Notebooks:** You can create Jupyter notebooks on Google Colab. These notebooks can contain your NLP code, documentation, and visualizations.
- **GPU Support:** Google Colab provides free GPU access, which is crucial for training deep learning models used in NLP tasks.
- **Installing Libraries:** You can install NLP libraries such as spaCy, NLTK, TensorFlow, PyTorch, and Hugging Face Transformers to work on your NLP tasks.
- **Data Integration:** You can upload your datasets to Google Colab or access data from Google Drive, making it convenient for data preprocessing and analysis.
- **Collaboration:** Google Colab allows you to collaborate with others in real-time, which can be helpful if you're working on a team project.
- **OpenAI API:**

- The OpenAI API provides access to advanced natural language models, including GPT-3. It can be a powerful tool for various NLP tasks, such as text generation, language translation, and text summarization. Here's how you can use the OpenAI API for your NLP project:

- **API Access:** You need to sign up for access to the OpenAI API and obtain an API key.
- **Integration:** You can integrate the OpenAI API into your NLP code to leverage its capabilities. OpenAI provides API documentation and example code to help you get started.
- **Text Generation:** You can use the API to generate human-like text, answer questions, or even create chatbots.
- **Custom Models:** OpenAI allows you to fine-tune their models on your specific tasks or datasets, which can be particularly useful for domain-specific NLP tasks.
- **Cost Consideration:** Be mindful of the cost associated with using the OpenAI API, as it typically involves a usage-based pricing model.

2. Library

When working on NLP (Natural Language Processing) projects, you'll need various libraries and frameworks to perform tasks like text processing, machine learning, and model development. Here are some essential libraries and frameworks commonly used in NLP projects:

NLTK (Natural Language Toolkit):

NLTK is a popular Python library for NLP. It provides tools and resources for tasks like tokenization, stemming, lemmatization, part-of-speech tagging, and more.

scikit-learn:
scikit-learn is a machine learning library that provides a wide range of tools for classification, regression, clustering, and model evaluation. It's often used in NLP for text classification and sentiment analysis.
Website: https://scikit-learn.org/stable/

Transformers (Hugging Face):

- Transformers is a library by Hugging Face that offers pre-trained state-of-the-art models for various NLP tasks. It includes BERT, GPT-2, and other models.
- Website: https://huggingface.co/transformers/

PyTorch and TensorFlow:

- These deep learning frameworks are essential for building and training neural networks for NLP tasks. PyTorch and TensorFlow both have NLP-specific libraries and pre-trained models.

OpenAI API:

You can use the OpenAI API, specifically the GPT-3 model, for word embeddings by providing a prompt that instructs the model to embed a specific word or phrase in context. The API response will contain the contextual embedding of that word or phrase.

3. Backend:

MongoDB is a NoSQL database management system known for its flexibility and scalability. It stores data in a flexible, schema-less format called BSON (Binary JSON), making it suitable for handling unstructured or semi-structured data. MongoDB is designed for horizontal scalability, which means you can easily expand your database by adding more servers. It is commonly used in web and mobile applications, IoT, and big data projects due to its ability to handle large volumes of data and support for geospatial data.

V. Deployment Environment

Clever Cloud: is a European Platform as a Service company. Clever Cloud provides PaaS solutions (runtimes, managed databases, object storage, etc.) to help developers deploy, run their apps and achieve software delivery faster in the cloud.

VI. Conclusion

In this chapter, we delved into the crucial phase of system design and planning for our NLP project. This phase lays the foundation for the successful development and implementation of our NLP solution. We initiated this process by outlining the key functionalities and features that our NLP system should encompass, ensuring that it aligns seamlessly with the project's objectives.

Furthermore, we took significant steps to identify the necessary project environments, which are vital for the efficient execution of our NLP tasks. These environments include the development and production environments, which will serve as the backdrops for our experiments and the eventual deployment of our NLP solution.

With these preliminary preparations meticulously set in motion, we are now well-prepared to progress into the next phase of our project, where we will begin building the core components of our NLP system. We will commence this journey by addressing fundamental elements such as user authentication and management, laying the groundwork for the comprehensive development and deployment of our NLP capabilities.

.

Chapter III

Data cleaning

I. Introduction

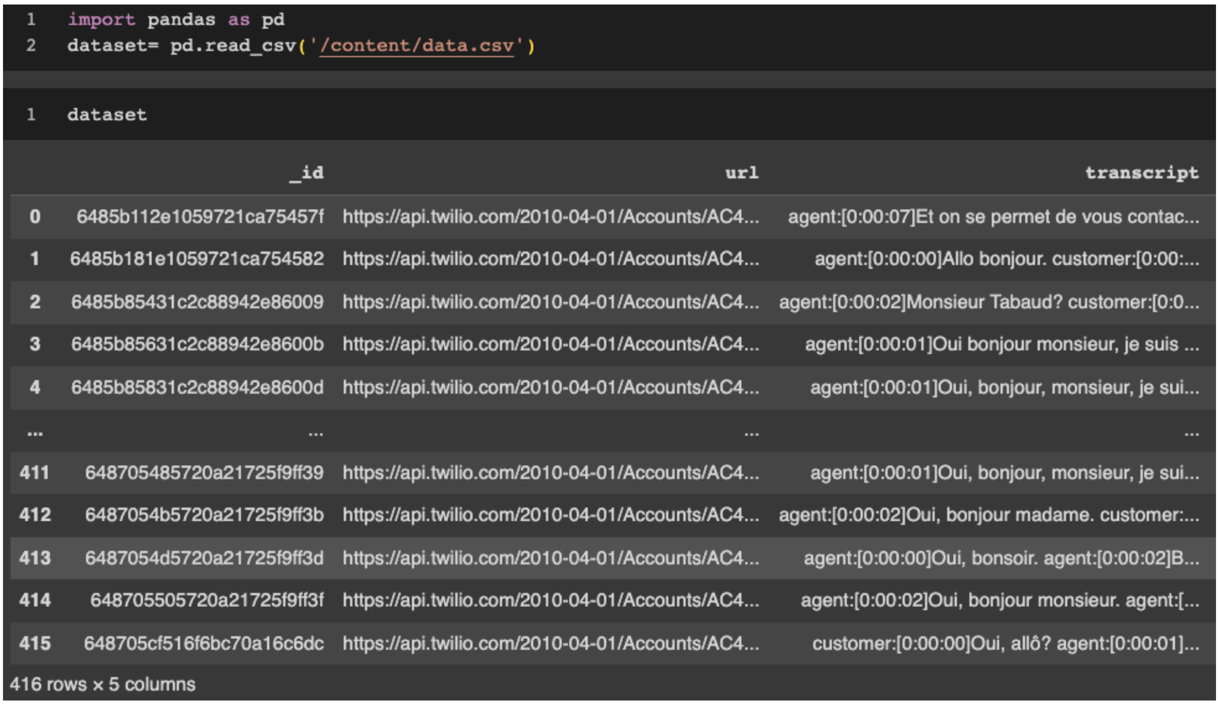
In this chapter, we delve into the crucial phase of data cleaning, a foundational step in the analysis of phone call data. The process involves ensuring data integrity, removing inconsistencies, and preparing the dataset for meaningful analysis.

II. Objective

The primary objective during the data cleaning phase was to prepare the phone call data for in-depth analysis, specifically targeting short customer calls lasting less than 2 minutes. The goal was to identify rejection reasons such as customer unavailability, prior product ownership, or disinterest.

III. Data Acquisition and Safeguarding

The data cleaning process commenced with establishing a connection to the MongoDB server, granting access to the valuable phone call data. To ensure data preservation and integrity, a precautionary measure was implemented: saving the original dataset locally as a CSV file. This safeguarded the raw data while providing a separate copy for analysis and manipulation.



```python
import pandas as pd
dataset= pd.read_csv('/content/data.csv')
```

```python
dataset
```

	_id	url	transcript
0	6485b112e1059721ca75457f	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:07]Et on se permet de vous contac...
1	6485b181e1059721ca754582	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:00]Allo bonjour. customer:[0:00:...
2	6485b85431c2c88942e86009	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:02]Monsieur Tabaud? customer:[0:0...
3	6485b85631c2c88942e8600b	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:01]Oui bonjour monsieur, je suis ...
4	6485b85831c2c88942e8600d	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:01]Oui, bonjour, monsieur, je sui...
...	...	...	...
411	648705485720a21725f9ff39	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:01]Oui, bonjour, monsieur, je sui...
412	6487054b5720a21725f9ff3b	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:02]Oui, bonjour madame. customer:...
413	6487054d5720a21725f9ff3d	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:00]Oui, bonsoir. agent:[0:00:02]B...
414	648705505720a21725f9ff3f	https://api.twilio.com/2010-04-01/Accounts/AC4...	agent:[0:00:02]Oui, bonjour monsieur. agent:[...
415	648705cf516f6bc70a16c6dc	https://api.twilio.com/2010-04-01/Accounts/AC4...	customer:[0:00:00]Oui, allô? agent:[0:00:01]...

416 rows × 5 columns

IV. Filtering Short Calls

To narrow our focus, calls exceeding the 2-minute mark were filtered out from the dataset. This step allowed us to specifically target short customer calls, which often contain critical information regarding rejection reasons.

V. Partitioning Data

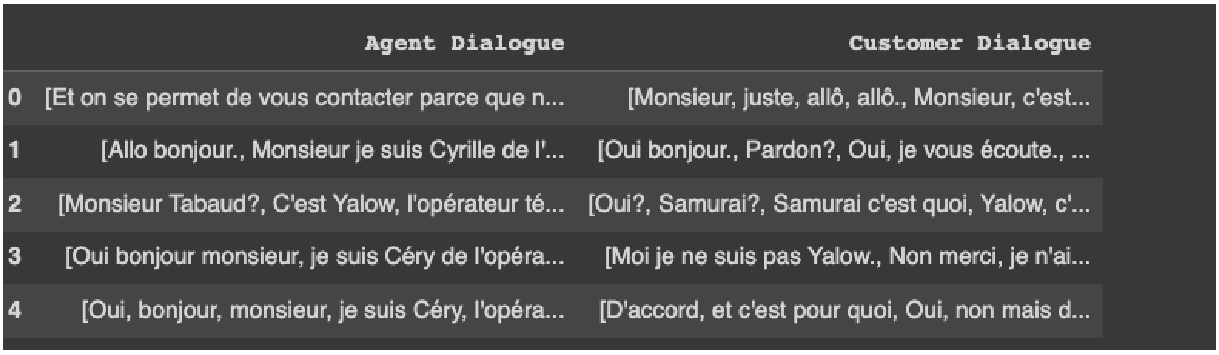
The next pivotal step involved partitioning the data into two distinct categories: client scripts and agent scripts. This partitioning was essential for distinguishing between the information provided by the customer and the responses from the client side.

VI. Removing Call Duration Information

In pursuit of a more concise analysis, the call duration information was removed from both client and agent scripts. This step streamlined the dataset, making it more conducive to identifying rejection reasons without the influence of call duration.

VII. Data Organization

The processed client and agent scripts were meticulously organized and structured within a new dataframe. This organization enhanced the dataset's readability and facilitated focused analysis of the rejection reasons.



	Agent Dialogue	Customer Dialogue
0	[Et on se permet de vous contacter parce que n...	[Monsieur, juste, allô, allô., Monsieur, c'est...
1	[Allo bonjour., Monsieur je suis Cyrille de l'...	[Oui bonjour., Pardon?, Oui, je vous écoute., ...
2	[Monsieur Tabaud?, C'est Yalow, l'opérateur té...	[Oui?, Samurai?, Samurai c'est quoi, Yalow, c'...
3	[Oui bonjour monsieur, je suis Céry de l'opéra...	[Moi je ne suis pas Yalow., Non merci, je n'ai...
4	[Oui, bonjour, monsieur, je suis Céry, l'opéra...	[D'accord, et c'est pour quoi, Oui, non mais d...

VIII. Conclusion

The data cleaning phase represents a pivotal milestone in our analysis of short customer calls. By safeguarding the original data, filtering short calls, partitioning data into client and agent scripts, and removing call duration information, we have prepared a clean and structured dataset ready for detailed analysis of rejection reasons.

In the subsequent chapter, our focus will shift towards analyzing and extracting valuable insights from this refined dataset, shedding light on the primary reasons fo

Chapter IV

Data preprocessing

I. Introduction

In the realm of data science, the journey from raw data to actionable insights begins with a critical phase known as data preprocessing. This chapter delves into the intricate art of data preparation, an indispensable step that ensures our data is primed for rigorous analysis and modeling. The process of data preprocessing is akin to refining raw materials before crafting a masterpiece.

II. The Crucial Role of Data Preprocessing

Data preprocessing is the unsung hero of data science, silently working behind the scenes to cleanse and structure data for further exploration. Whether dealing with numerical data, images, or in our case, text data, the significance of this phase cannot be overstated. In the context of Natural Language Processing (NLP), where we grapple with the nuances of human language, data preprocessing takes on a distinct importance.

III. Navigating the Landscape of Text Data Preprocessing

Our journey into data preprocessing focuses particularly on text data, where unstructured text is transformed into a structured format amenable to analysis. Below, we explore the key steps in text data preprocessing:

1. Tokenization: At the heart of text analysis lies tokenization. This process disassembles text into its constituent units, often words or subwords, known as tokens. Tokenization is akin to breaking a long sentence into individual words, making text more manageable for analysis. Popular libraries such as NLTK, spaCy, and NLTK in Python provide robust tokenization functions.

2. Lowercasing: In the quest for uniformity and simplicity, all text is converted to lowercase. This treatment ensures that words with different cases are treated as identical, reducing the complexity of the vocabulary.

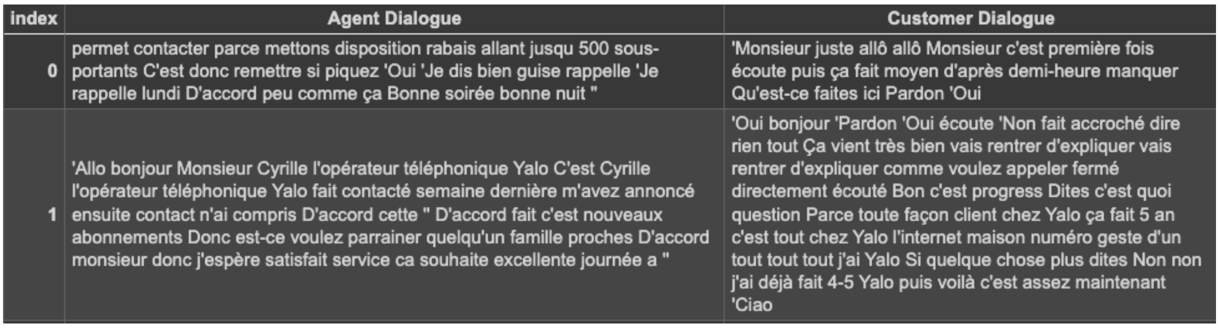
3. Removing Punctuation: Punctuation marks, those silent characters that punctuate sentences, are oftentimes removed from the text. While essential for grammar, in isolation, they contribute little to the meaning of the text.

4. Stopword Removal: The realm of text is often cluttered with "stopwords" like "and," "the," and "is" – words that occur frequently but hold minimal significance in isolation. The removal of stopwords helps declutter the data, preserving only the most informative content. Libraries like NLTK and spaCy provide curated lists of stopwords.

5. Stemming and Lemmatization: Language is rich with variations of words. For instance, "running" might become "run" in its simplest form. Stemming and lemmatization are techniques employed to reduce words to their root or base forms. While stemming offers a simpler approach, lemmatization leverages linguistic knowledge for precise transformations.

6. Special Character Removal: Special characters, symbols, and non-alphanumeric characters are often extraneous and can be safely removed from text data, contributing little to the analysis or modeling process.

7. Removing Extra Spaces: In the quest for consistency, extra spaces are normalized to a single space, ensuring uniform text processing.



index	Agent Dialogue	Customer Dialogue
0	permet contacter parce mettons disposition rabais allant jusqu 500 sous-portants C'est donc remettre si piquez 'Oui 'Je dis bien guise rappelle 'Je rappelle lundi D'accord peu comme ça Bonne soirée bonne nuit "	'Monsieur juste allô allô Monsieur c'est première fois écoute puis ça fait moyen d'après demi-heure manquer Qu'est-ce faites ici Pardon 'Oui
1	'Allo bonjour Monsieur Cyrille l'opérateur téléphonique Yalo C'est Cyrille l'opérateur téléphonique Yalo fait contacté semaine dernière m'avez annoncé ensuite contact n'ai compris D'accord cette " D'accord fait c'est nouveaux abonnements Donc est-ce voulez parrainer quelqu'un famille proches D'accord monsieur donc j'espère satisfait service ca souhaite excellente journée a "	'Oui bonjour 'Pardon 'Oui écoute 'Non fait accroché dire rien tout Ça vient très bien vais rentrer d'expliquer vais rentrer d'expliquer comme voulez appeler fermé directement écouté Bon c'est progress Dites c'est quoi question Parce toute façon client chez Yalo ça fait 5 an c'est tout chez Yalo l'internet maison numéro geste d'un tout tout tout j'ai Yalo Si quelque chose plus dites Non non j'ai déjà fait 4-5 Yalo puis voilà c'est assez maintenant 'Ciao

IV. Conclusion

Data preprocessing stands as a testament to the meticulousness required in the data science journey. Through the lens of text data preprocessing, we've explored the intricate steps necessary to prepare unstructured text for analysis and modeling. These steps serve as the foundation upon which we build our insights and drive informed decision-making.

As we progress through the subsequent chapters of our data science report, we will harness the power of well-preprocessed data to extract valuable insights, uncover hidden patterns, and ultimately empower data-driven decision-making.

Chapter V

Word Vectorization, Word Embedding, Unsupervised Machine Learning

I. Introduction

In this chapter, we delve into the transformative power of word vectorization and word embedding in the field of Natural Language Processing (NLP). We explore how these techniques enable machines to understand and work with textual data, paving the way for unsupervised learning algorithms and crucial metrics to evaluate their performance.

II. Word Vectorization

Word vectorization, also known as word representation, is the process of translating words into numerical vectors, making them suitable as inputs for machine learning algorithms. This process is essential in NLP tasks, enabling computers to understand and process inherently non-numeric textual data.

A. TF-IDF: Term Frequency-Inverse Document Frequency

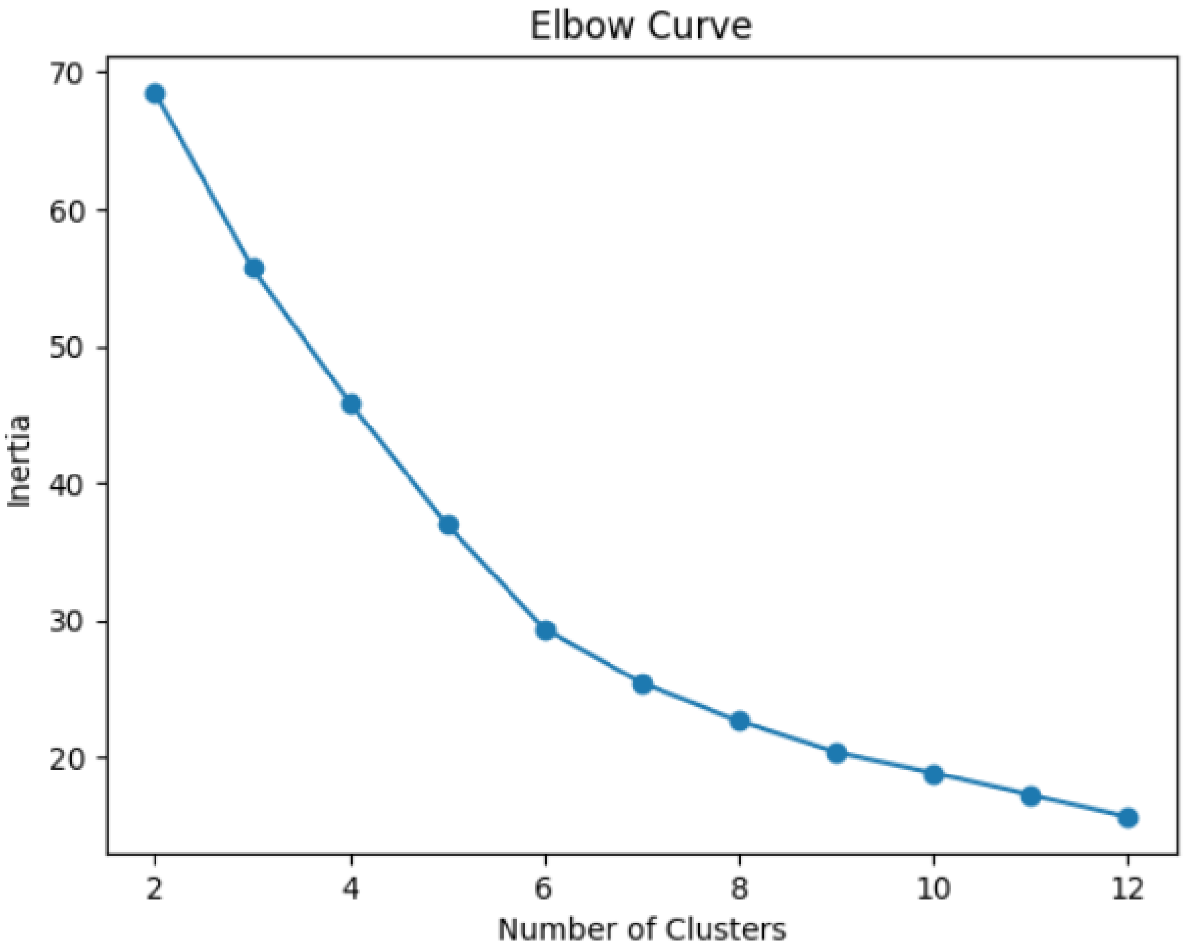
In our case, we adopted the TF-IDF (*Term Frequency-Inverse Document Frequency*) word vectorization technique. TF-IDF is a numerical representation of a word's importance in a specific document and across an entire corpus. Mathematically expressed as TF-IDF(w, d) = TF(w, d) * IDF(w), this technique measures both the word's frequency in a specific document and its rarity across the entire corpus. This approach allows us to precisely weigh the importance of words.

III. Unsupervised Machine Learning: K-means Clustering

With textual data now transformed into numerical vectors, we venture into the realm of unsupervised machine learning. Our task falls under the domain of clustering, where we seek to group similar data points into clusters. For this purpose, we chose the well-known *K-means* algorithm.

A. The Elbow Technique for Determining the Optimal Number of Clusters

Determining the optimal number of clusters is a crucial task in K-means clustering. We applied the *elbow technique*, which identifies the "elbow point" where the rate of explained variation significantly changes. This inflection point indicates the optimal number of clusters that effectively capture the data patterns.



IV. Model Performance Evaluation: Silhouette Score

To assess the effectiveness of our clustering models, we turned to the *silhouette score* as a key metric. The silhouette score quantifies the quality of clusters in unsupervised learning, particularly in K-means. It measures how well data points belong to their own cluster compared to neighboring clusters. A higher silhouette score signifies well-separated and distinct clusters.

Here's how the silhouette score is calculated:

For each data point, calculate:

- **a**: The average distance from the data point to the other points in the same cluster (cohesion).
- **b**: The smallest average distance from the data point to points in other clusters (separation).

The silhouette score for a single data point is given by:

$\text{silhouette score}=\max(a,b)b-a$

The silhouette score for the entire dataset is the average of the silhouette scores of individual data points.

Interpretation of silhouette scores:

- A score near +1 indicates that the data point is far away from neighboring clusters and well-matched to its own cluster.
- A score near 0 indicates that the data point is on or very close to the decision boundary between two neighboring clusters.
- A score near -1 indicates that the data point is assigned to the wrong cluster.

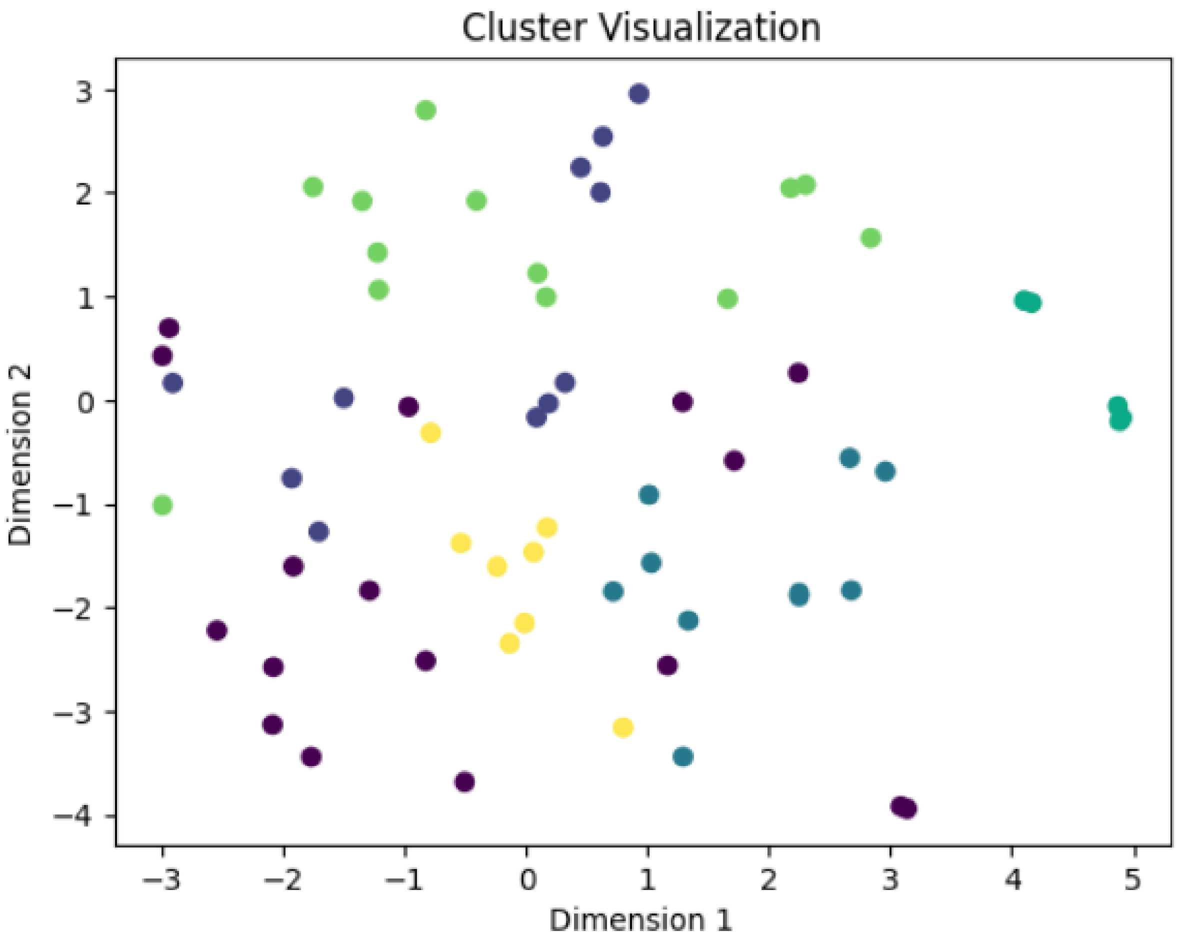
The inclusion of the silhouette score adds a crucial dimension to our evaluation of clustering performance, allowing us to quantitatively measure the quality of the clusters generated by our models.

V. Word Embedding

Word embedding is a fundamental technique in NLP that aims to represent words as continuous vectors in a high-dimensional space. These vectors capture semantic relationships between words, enhancing machine learning models' understanding of word meanings and contexts.

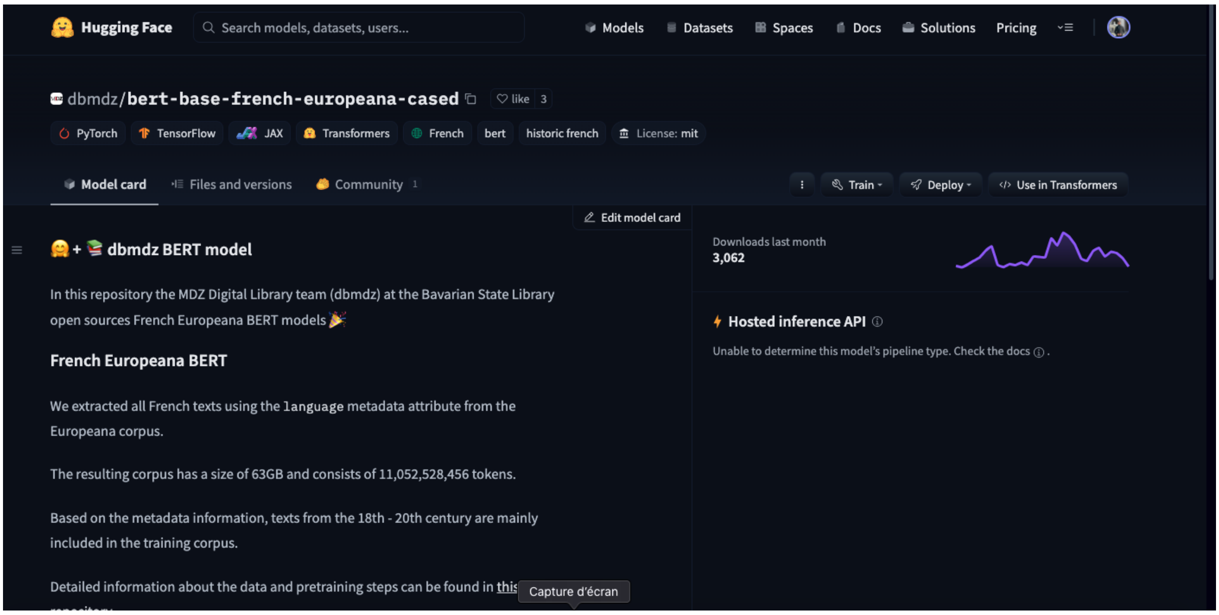
A. Constructing Word Embeddings with LSTM

We leveraged LSTM (*Long Short-Term Memory*) networks, a type of recurrent neural network (RNN) architecture, to generate word embeddings. These embeddings transformed words into numerical vectors encapsulating their meanings. We then applied an agglomerative clustering method to group similar embeddings into clusters.



B. Pretrained Models and Their Challenges

The dataset's size posed challenges for effective embedding of tokens using our LSTM model. We turned to a pretrained model (BERT) but encountered difficulties due to varying token counts, resulting in matrices of different sizes. Attempts to standardize matrix sizes led to a decrease in embedding quality, making it challenging for the machine learning model to perform effective clustering, with a silhouette score of less than 1. This underscores the importance of a balanced dataset and appropriate embedding techniques for meaningful clustering outcomes.



🤗 + 📚 dbmdz BERT model

In this repository the MDZ Digital Library team (dbmdz) at the Bavarian State Library open sources French Europeana BERT models 🎉

French Europeana BERT

We extracted all French texts using the `language` metadata attribute from the Europeana corpus.

The resulting corpus has a size of 63GB and consists of 11,052,528,456 tokens.

Based on the metadata information, texts from the 18th - 20th century are mainly included in the training corpus.

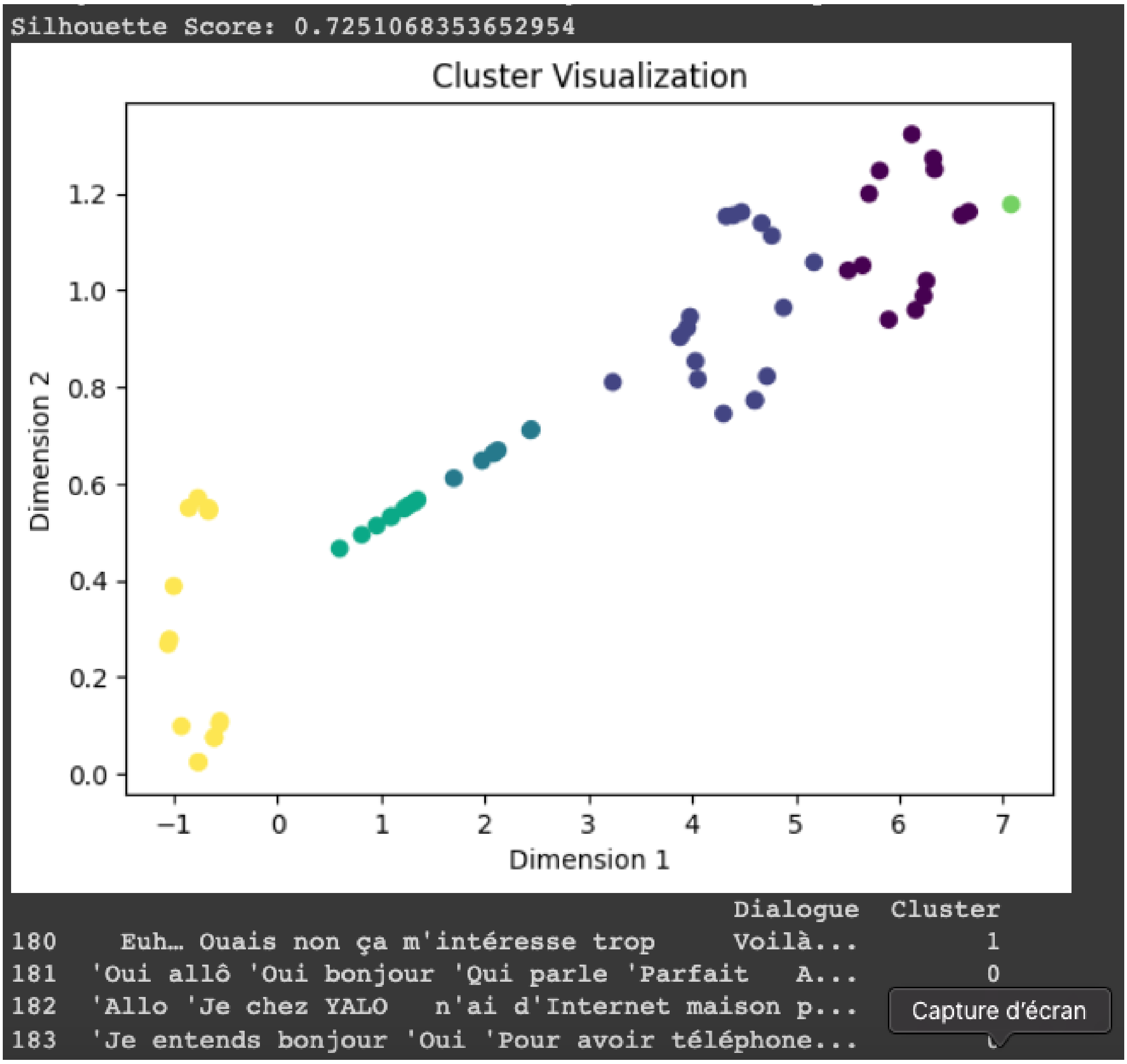
Detailed information about the data and pretraining steps can be found in this

Unable to determine this model's pipeline type. Check the docs ⓘ .

VI. OpenAI Embedding API

Incorporating the OpenAI embedding API into our workflow proved to be a game-changer. This API harnesses the power of a robust pretrained model trained on a vast dataset, providing highly reliable word embeddings. When seamlessly integrated with our K-means clustering model, these embeddings yielded remarkable results.

The pinnacle of this achievement was a remarkable silhouette score of 0.7, emphasizing the significant separation achieved between clusters. This outcome serves as a testament to the effectiveness of our embedding strategy and subsequent clustering. It underscores the model's exceptional ability to discern and categorize distinct clusters effectively, further enhancing our understanding of the underlying data structures and relationships.



```
Silhouette Score: 0.7251068353652954
```

```
                                               Dialogue  Cluster
180    Euh… Ouais non ça m'intéresse trop      Voilà...        1
181  'Oui allô 'Oui bonjour 'Qui parle 'Parfait   A...        0
182  'Allo 'Je chez YALO   n'ai d'Internet maison p...
183  'Je entends bonjour 'Oui 'Pour avoir téléphone...
```

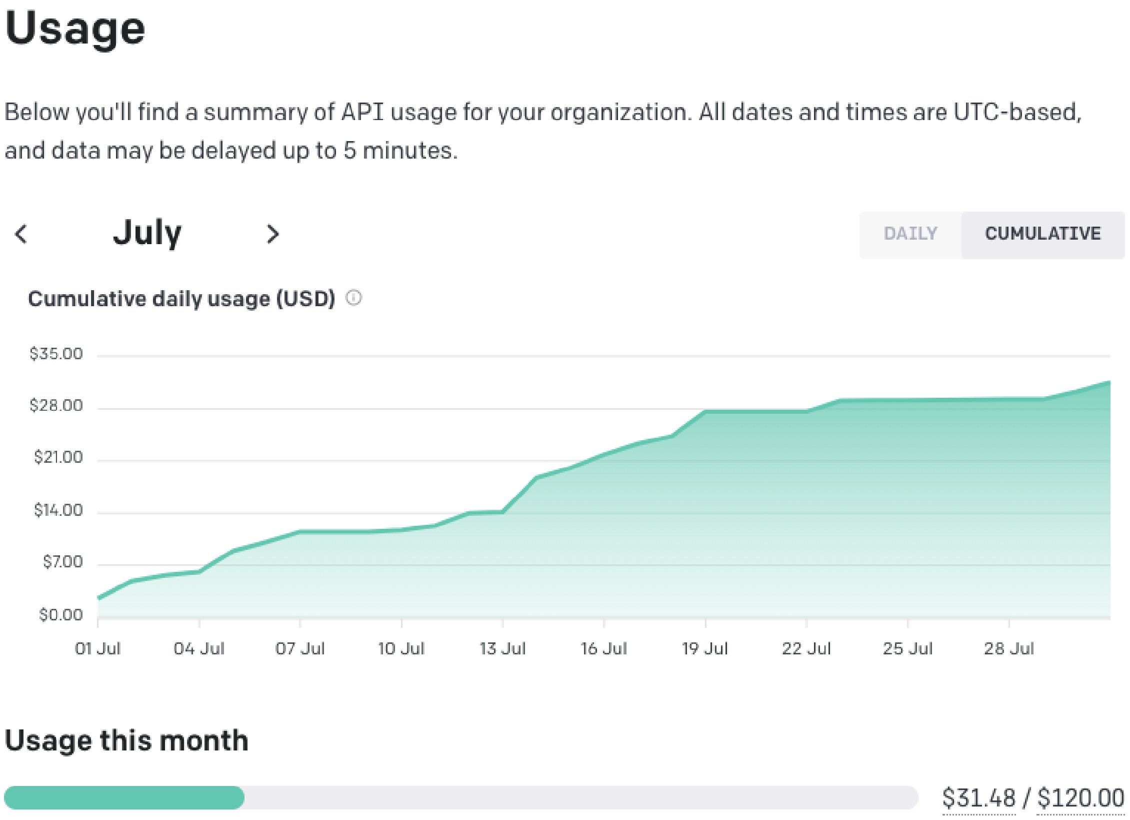
In conclusion, this chapter has unraveled the transformative potential of word embedding techniques and their critical role in advancing NLP. We have explored both LSTM-based and pretrained model approaches, gaining insights into their strengths and limitations. The integration of the OpenAI embedding API has solidified our ability to generate high-quality word embeddings and unlock deeper layers of meaning within textual data.

Chapter VI

Optimization and Insights

I. Introduction

In this chapter, we delve into the optimization strategies employed in our analysis, focusing on keyword-based algorithms and language detection techniques. These approaches are not only efficient in pinpointing specific issues but also offer valuable insights into enhancing client satisfaction and communication quality.



Usage

Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.

Usage this month

$31.48 / $120.00

II. Keyword-Based Algorithm

The core of our optimization strategy lies in the application of keyword-based algorithms to identify factors contributing to client dissatisfaction. This approach proves invaluable in swiftly identifying key reasons behind client concerns. By analyzing keywords within communications, we gain insights into a range of potential issues, including failed call attempts (voicemail), costly services, and trust-related concerns when clients engage with other companies. This algorithm enables the rapid identification of critical areas for improvement and timely intervention strategies.

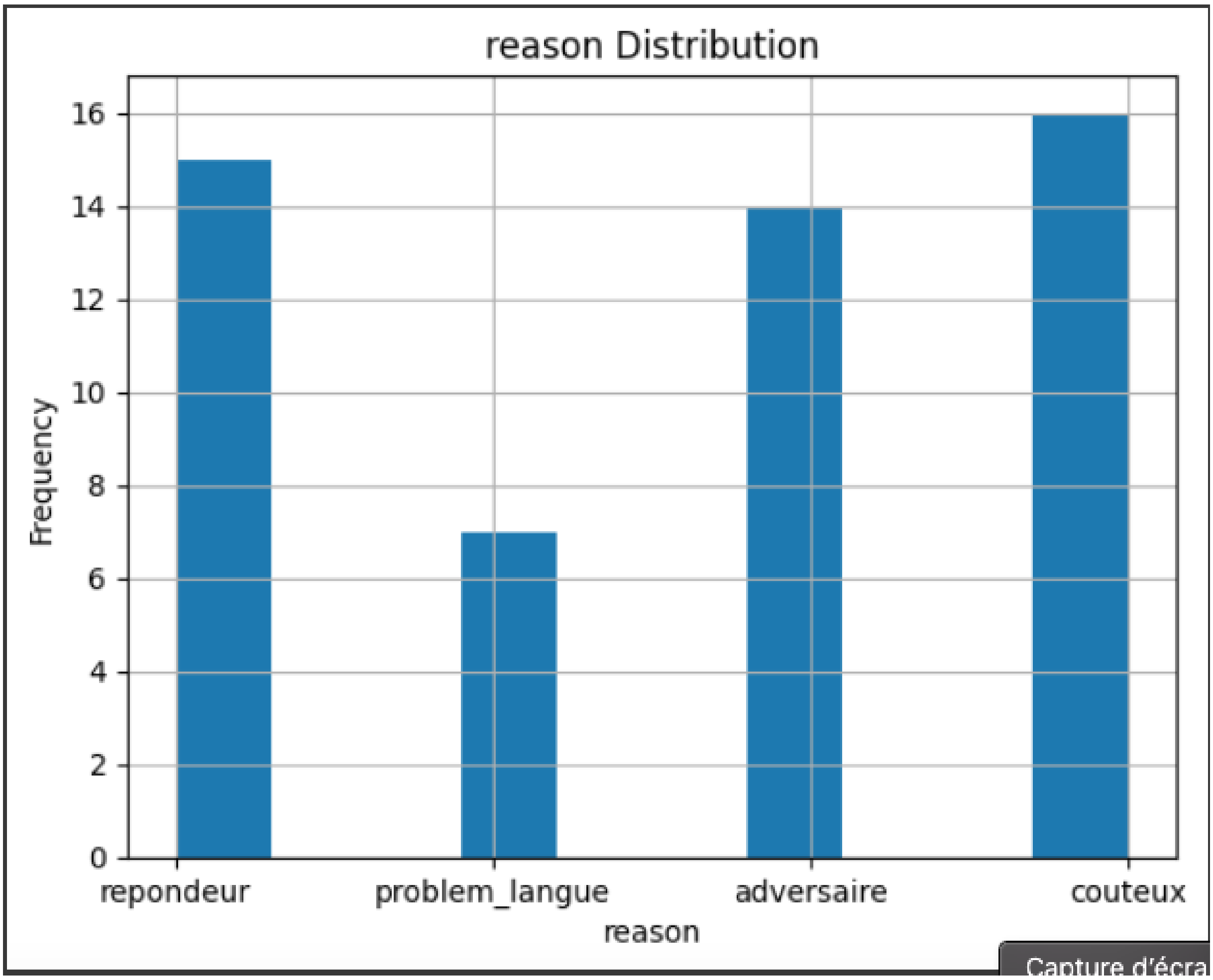
III. Language Detection

Effective communication is vital, and language mismatches can lead to misunderstandings and dissatisfaction. To address this, we employed the "detect" function from the "langdetect" package. This technique allowed us to distinguish discrepancies in language usage between clients and agents. As the primary language used by agents was French, instances where clients used a different language became potential indicators of language-related issues. This approach streamlined communication and enhanced the quality of interactions between clients and agents.

IV. Insights and Results

Through our optimization efforts, we achieved significant results. In a sample set of 200 cases, we accurately identified 50 instances of client dissatisfaction, as confirmed by our histogram analysis. Remarkably, this accounts for 25 percent of the dataset, signifying the substantial impact of our strategies. These insights not only provide valuable information for improving client satisfaction but also optimize resource utilization.

Additionally, we maintained cost-effectiveness by utilizing less than 20 percent of tokens for the OpenAI word embedding API. This demonstrates our commitment to preserving financial resources without compromising the accuracy of our analysis. The outcomes of our optimization strategies underscore their efficiency and effectiveness in uncovering key insights and areas for improvement within the dataset.



V. Conclusion

In this chapter, we explored the optimization techniques applied to our analysis, focusing on keyword-based algorithms and language detection. These strategies not only identified client dissatisfaction but also offered insights for improving communication and overall satisfaction. Our commitment to cost-effective analysis ensures that resources are used efficiently while delivering meaningful results.

General Conclusion

In this comprehensive report, we embarked on a journey through the realms of Natural Language Processing (NLP) and data analysis. Our mission was to analyze client communication data to uncover patterns, improve customer satisfaction, and optimize processes. Throughout this endeavor, we navigated through various stages, from data collection and preprocessing to advanced techniques like word vectorization, word embedding, and unsupervised machine learning. Our analysis culminated in actionable insights and optimization strategies. Let's recap the key findings and the impact of our work.

Data Collection and Preprocessing: We initiated our journey by connecting to a MongoDB server to access client call data. Safeguarding the original data and partitioning it into client and agent scripts enabled focused analysis.

Keyword-Based Algorithm and Language Detection: We harnessed the power of keyword-based algorithms to swiftly identify key factors contributing to client dissatisfaction. Additionally, our language detection techniques allowed us to address language-related issues, enhancing communication quality.

Word Vectorization and Word Embedding: Word vectorization, particularly using the TF-IDF technique, transformed textual data into numerical vectors for analysis. Word embedding with LSTM and pretrained models enriched our understanding of semantic relationships within the data.

Unsupervised Machine Learning: Leveraging K-means clustering, we grouped similar data points into clusters, providing insights into underlying patterns and structures.

Metrics and Optimization: We employed the silhouette score to quantitatively evaluate the quality of clusters, ensuring our models effectively segmented the data. This metric added a crucial dimension to our analysis.

OpenAI Embedding API: Integrating the OpenAI embedding API yielded remarkable results, with a silhouette score of 0.7, indicating significant cluster separation.

Optimization and Insights: We optimized our analysis by efficiently identifying instances of client dissatisfaction while economizing on resource utilization. Our efforts showcased the potential to preserve financial resources without compromising analysis accuracy.

General Conclusion: Our journey through NLP and data analysis has illuminated essential insights for improving client satisfaction, communication quality, and overall processes. By employing cutting-edge techniques and optimization strategies, we have laid the foundation for data-driven decision-making and continuous improvement.

As we conclude this report, we emphasize the importance of ongoing analysis and adaptation to the evolving needs of the organization. The journey through data analysis is a continuous one, and the insights gained here serve as a springboard for future endeavors. We look forward to the ongoing pursuit of excellence in client communication and satisfaction.

Thank you for joining us on this journey, and we remain committed to data-driven excellence in the future.

References

◆for machine learning : https://www.youtube.com/@statquest

◆pretrained model : https://huggingface.co/models

◆ Natural Language Toolkit :https: //www.nltk.org/

◆ OpenAI api : https://platform.openai.com/docs/models/gpt-3-5

◆ https://towardsdatascience.com/topics-per-class-using-bertopic-252314f2640